



Defending Unauthorized Voice Cloning with Watermark-Aware Codecs

Jiankun Zhao¹, Lingwei Meng¹, Chengxi Deng¹, Helen Meng¹, Xixin Wu¹

¹The Chinese University of Hong Kong, Hong Kong SAR, China

{jzkzhao, lmeng, cxdeng, hmmeng, wuxx}@se.cuhk.edu.hk

Abstract

The proliferation of zero-shot TTS models increases the risk of malicious voice cloning using copyrighted speech prompts. Although audio watermarking provides an effective way for encoding copyright information, attackers may still use watermarked speech as prompts to synthesize unwatermarked speech with the same speaker identity. To protect copyrighted voices from being cloned, this study introduces a method to train open-source TTS models to reject watermarked speech prompts for cloning. We observe that mainstream zero-shot TTS models typically rely on pre-trained codec encoders to process speech prompts. By training the codec to “mute” when encountering watermarked audio, the quality of generated speech will degrade. In this way, we implicitly prevent zero-shot TTS models from cloning watermarked voices. Experiments show that our approach is robust against various attacks while maintaining high-quality TTS ability given unwatermarked speech prompts.

Index Terms: voice cloning, audio watermarking, source tracing, privacy protection

1. Introduction

Modern zero-shot text-to-speech (TTS) models [1–4] can accurately imitate the voice of an unseen speaker given only a 3-second speech prompt. However, this revolutionary technology also carries potential risks of misuse, such as spoofing voice identification [5] or impersonating specific speakers [6]. A most common kind of misuse is called voice cloning attack, which refers to unauthorized use of a copyrighted speech as the prompt to generate others’ voices [7]. For example, a fake speech generated by cloning Trump’s voice may cause public panic, whereas a deceptive telephone recording imitating a familiar voice may lead to financial losses.

In order to distinguish AI-generated audio, previous works have explored both passive and proactive detection methods. Passive detection [8–11] mainly utilizes anti-spoofing models as binary classifiers to predict whether the audio is AI-generated. Another line of works [12, 13] focus on source tracing. These works try to further predict which model generated the fake audio. Though straightforward, these models tend to make mistakes when encountering audio generated from new unseen models [14, 15]. Proactive detection generally exploits audio watermark models [7, 16–18], each composed of a pair of watermark embedder and detector. State-of-the-art audio watermarking models can not only mark and detect AI-generated audio, but also encode copyright information imperceptibly [19]. However, most of these detectors work independently with the generative model. This means that they can only help trace the source of a suspicious audio after it is generated, but not pre-

venting the illegal cloning of copyrighted voices in advance. While some works [20, 21] explored ways to integrate the watermark embedder into the audio synthesizer, few works have considered equipping the audio synthesizer with the ability to detect watermarked speech prompts.

In this work, we propose a method to prevent general open-source zero-shot TTS models from cloning copyrighted voices. To construct such a method, we need to answer three questions: (1) How to integrate watermark detection into an open-source TTS model; (2) How to force a TTS model to fail when encountering watermarked voices; (3) How to maintain the generation capacity of the underlying TTS model when an unwatermarked voice is given as the speech prompt. A naive solution is to concatenate a watermark detector and a pre-trained TTS model using a hard-coded “if-else” logic: to terminate the generation process when a watermark is detected in the speech prompt. However, attackers may remove this “if-else” logic if the TTS model is open-sourced. Another potential way is to hide this “if-else” logic into the parameters of the TTS model, whereas the watermark information may not retain after quantization in codec encoder [22]. Instead, we start with the observation that regardless of their detailed architecture, state-of-the-art zero-shot TTS models generally utilize pre-trained codec encoders to encode the speech prompt. Therefore, we try to make these codec encoders “fail” by training the codec to return zero-value waveform when given watermarked speech prompts. As long as the codec encoder learns to detect and reject watermarked speech, the TTS models trained on the pretrained codec encoder will be able to defend unauthorized voice cloning attacks. Additionally, these safeguarded TTS models will need to be re-trained to adapt to other ordinary codecs, making it difficult for attackers to bypass the protection mechanism.

The main contributions of this paper can be concluded as follows: (1) We propose a method to prevent open-source zero-shot TTS models from cloning copyrighted voices, which is the first of its kind to the best of our knowledge; (2) Our method can protect TTS models by training their codec encoders to detect and refuse watermarked speech prompts in a parametric way; (3) Experiments show that our method not only preserves the synthesis quality of the underlying TTS models, but is also robust to various attacks.

2. Method

In this section, we first formulate our problem in a threat modeling manner (see Section 2.1). Then, we describe the training process of watermark-aware codecs (see Figure 1(a) and Section 2.2). Finally, we introduce how the watermark-aware codec encoders reject watermarked speech prompts in the TTS pipeline (see Figure 1(b) and Section 2.3).

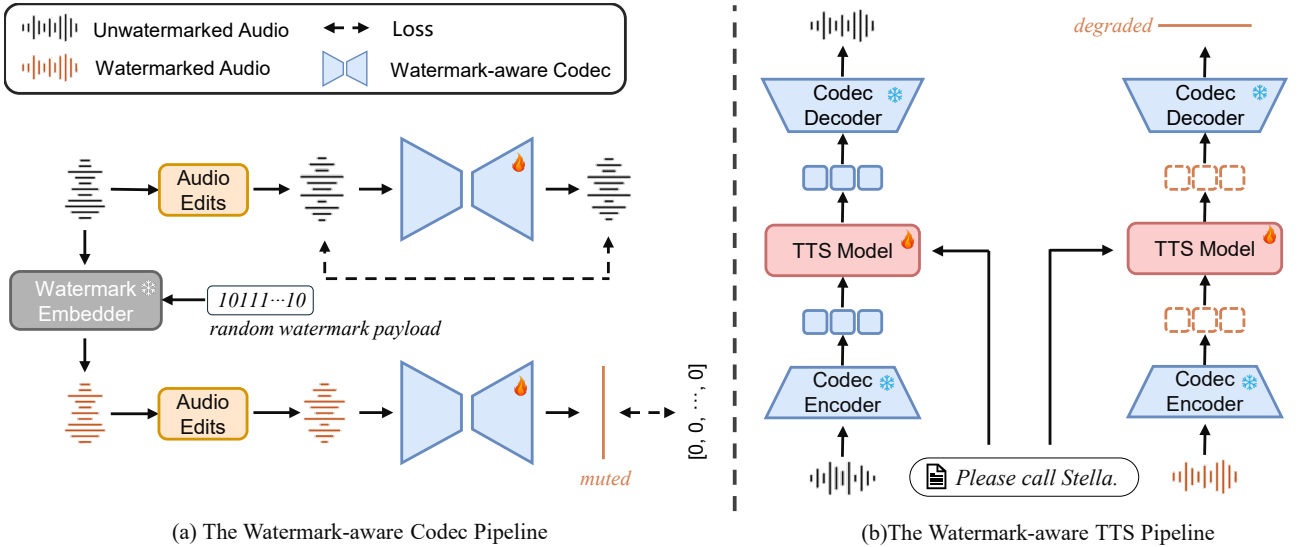


Figure 1: (a) Pipeline of watermark-aware codecs. When input audio is not watermarked, the target output is set as the original input. Otherwise, the target output is set as all zeros. (b) TTS pipeline with watermark-aware codec encoders. If the speech prompt is watermarked, the codec encoder will output codes of a silent audio clip, prompting the TTS model to generate degraded speech.

2.1. Task Formulation

Consider a scenario involving users, developers and attackers. In order to declare copyright and prevent potential misuse, the user adds a watermark into the original speech before sharing it on the Internet. The attackers collect the shared speech and prompt an open-source TTS model to clone this watermarked voice. The developers want to open-source their TTS model. However, they worry that their model may be used to clone copyrighted voice. Our mission is to protect TTS models from cloning copyrighted voice, so that developers can open-source their models.

In addition, we consider the case where attackers are aware of our methodology and try to bypass our safeguard strategy. Specifically, we focus on the watermark removal attack [7], where attackers try to remove the watermark by editing the watermarked audio, so that the codec encoder can no longer distinguish the edited audio as watermarked. Furthermore, since some zero-shot TTS models are robust to subtle perturbations in the speech prompt, the attackers may still synthesize speech with acceptable quality using edited prompts. Therefore, our another mission is to make the watermark-aware codecs robust against watermark removal attacks. In other words, our model should be able to detect the watermark even if the watermarked audio was maliciously edited.

2.2. Codec

We train the codec encoder to detect and reject watermarked speech prompts by modifying the reconstruction target. As depicted in Figure 1(a), we first add a watermark onto each utterance in the speech dataset, using a pretrained watermark embedder. Since we only care whether the audio is watermarked, the multi-bit payload part in the watermark is randomly initialized during the preprocessing stage and not detected when training the codec encoder. During the training stage, we sample clean utterances and watermarked utterances with equal probability. When a clean utterance is sampled, the reconstruction target \hat{x} is set to the input utterance itself. When a watermarked utterance is sampled, the reconstruction target \hat{x} is set to all zeros.

$$\hat{x} = \begin{cases} \mathbf{x}, & \mathbf{x} \text{ is not watermarked} \\ \mathbf{0}, & \text{otherwise} \end{cases} \quad (1)$$

In this way, the codec will learn to implicitly detect watermarks and reject watermarked audio by outputting a silent audio clip. On the other hand, it will learn to reconstruct the original audio when an unwatermarked audio is sampled, maintaining the core function of audio codecs.

However, codec encoders trained this way may not be robust against watermark removal attacks introduced in Section 2.1. Following [16], we consider 13 types of audio edits (speed adjustment, resampling, echo, pink noise, low/high/band-pass filters, smoothing, boosting/ducking, and MP3/AAC/EnCodec compression). To strengthen the codec encoder's ability to detect edited audios, we include these audio edits in codec training. Specifically, for each training utterance, no matter watermarked or not, we randomly sample one type of audio edit with equal probability and apply it onto the original utterance. The transformed utterance serves as both the input and target audio, taking the place of the original utterance. We also include non-edited utterances during training to ensure the reconstruction quality of the audio codec. These non-edited utterances are sampled with 10× probability than edited ones.

In this work, we consider EnCodec [23] as a representative of general audio codecs to demonstrate our concept. Following [24], we use a weighted combination of reconstruction loss \mathcal{L}_{rec} , adversarial loss \mathcal{L}_{adv} , feature matching loss \mathcal{L}_{feat} , and commitment loss \mathcal{L}_c to train the codec.

2.3. TTS Model

We train the TTS model based on the pretrained watermark-aware codecs. During the preprocessing stage, we encode the training audio into codec codes with the codec in Section 2.2. We only use unwatermarked data for TTS model training, alleviating the need for watermarking large-scale datasets.

During the inference stage, the speaker information is carried by the speech prompt and then encoded by the codec encoder. If the speech prompt is watermarked, the codec encoder will output codec codes of a silent audio clip. In this way, part of the speaker information in the speech prompt will be

System	ND	RS	BA	BP	HP	AAC	MP3	GN	PN	SM	EC	DA	EA	LP	FS	MEAN
AudioSeal	100	100	100	100	100	100	100	100	100	100	98.5	96	92.5	81	50	94.53
B1&B2	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50.00
O1	100	100	100	100	100	100	99.5	94	86.5	67.5	50.5	50	97	50	50	82.00
O2	100	100	100	100	100	100	100	99	92	95	93.5	53.5	96.5	59.5	50	89.27

Table 1: Accuracy (ACC%) of watermark-aware codecs to detect watermarked audio. Each column represent one kind of watermark-removal attacks. ND: No Distortion; RS: Resampling; BA: Boost Audio; BP: BandPass Filter; HP: HighPass Filter; AAC: AAC Compression; MP3: MP3 Compression; GN: Gaussian Noise; PN: Pink Noise; SM: Smoothing; EC: EnCodec; DA: Duck Audio; EA: Echo; LP: LowPass Filter; FS: Speed Adjustment.

System	PESQ \uparrow	STOI \uparrow	VISQOL \uparrow	TM (%) \downarrow
B1&B2	3.63	0.96	4.57	33.42
O1	3.28	0.94	4.26	2.24
O2	3.21	0.94	4.27	1.09

Table 2: Reconstruction quality of watermark-aware codecs.

removed, guiding the TTS model to synthesize silent or low-quality speech. If the speech prompt is unwatermarked, most information in the prompt speech will be compressed into the codec codes and then provided to the TTS model. The TTS model will thus generate speech with timbre and prosody similar to the prompt.

In open-source scenarios, attackers may attempt to bypass this protection by adapting the TTS model to another pre-trained codec that does not reject watermarked audio. We demonstrate in Section 3 that our watermark-aware codec has a code distribution significantly different from the baseline codec. Therefore, adapting the TTS model to an ordinary codec while maintaining high generation quality would require a lot of training data and computational resource. This hinders the attackers from launching an adaptation attack.

In this paper, we choose VALL-E [2], a popular zero-shot TTS model based on EnCodec [23], to demonstrate the effectiveness of our method.

3. Experiment

3.1. Settings

We conduct our experiments based on the open-source implementation of EnCodec¹ and VALL-E². The EnCodec encoder comprises 4 residual blocks, each with a downsample rate of 8, 5, 4 and 2, respectively. The quantizer quantizes the encoded features into 8 layers of codes. Each codebook has a size of 1024. We train the codec with the LibriSpeech-clean-100 dataset, and test the codec with 100 utterances randomly sampled from LibriSpeech test-clean dataset. To build the watermarked training and testing set, We add a watermark on each utterance in the original dataset with pretrained AudioSeal embedder³. The codec parameters are initialized with a checkpoint pretrained on unwatermarked LibriTTS, VCTK and AISHELL. Unless otherwise stated, we train the codec for 300 epochs.

The VALL-E model is composed of 12 transformer decoder layers, each with a 16-head self-attention block and a 4096-dimension feed-forward layer. The training dataset comprises all subsets of LibriTTS-R [25]. We train the TTS model for 20 epoches during the AR stage and 40 epoches during the NAR stage. We follow [26] to test the TTS model on LibriTTS-R test-clean, choosing another utterance from the same speaker as the speech prompt.

¹<https://github.com/yangdongchao/AcademiCodec>

²<https://github.com/lifeiteng/vall-e>

³<https://huggingface.co/facebook/audioseal>

We compare our method with two baselines, denoted as B1 and B2 respectively. In **B1**, we train both the codec and the TTS model solely on unwatermarked audio. In **B2**, we use the same “clean” codec as in B1 but attempt to train the TTS model to reject watermarked prompts. Specifically, we train the TTS model on both unwatermarked and watermarked speech, and modify the training target of all watermarked speech to a special token to perturb the generation process when given watermarked speech prompts. We also include the detection accuracy of the pre-trained AudioSeal detector (denoted as **AudioSeal**), which serves as an upper bound of watermark detection accuracy of our method.

Meanwhile, we denote two versions of our system as O1 and O2. **O1** omits the “Audio Edits” process in Figure 1 when training the codecs, whereas **O2** includes them during codec training. Since O2 better maintains generation quality and is more robust to watermark removal attacks, we select it as our default setting.

3.2. Evaluation Metrics

We first introduce metrics for evaluating the codec encoder’s ability to detect watermarked audio. Since the codec encoder does not explicitly output a binary classification label, we propose a new metric named silence ratio (SR). We define an audio frame as a silent frame if the frame-wise energy is below a threshold τ_e . The silence ratio is defined by the proportion of silent segments within an audio clip. An audio clip is considered silent if its silence ratio is below another threshold τ_r . In this work, we empirically set $\tau_e = 30\text{dB}$ and $\tau_r = 0.8$.

$$E_t = 10 \log_{10} \text{amplitude}_t^2 \quad (2a)$$

$$\text{silence}(t) = \begin{cases} 1, & E_t < \tau_e \\ 0, & \text{otherwise} \end{cases} \quad (2b)$$

$$SR(\text{wav}) = \frac{\sum_{t=1}^{\text{length}(\text{wav})} \text{silence}(t)}{\text{length}(\text{wav})} \quad (2c)$$

$$\text{silent}(\text{wav}) = \begin{cases} \text{true}, & SR(\text{wav}) > \tau_r \\ \text{false}, & \text{otherwise} \end{cases} \quad (2d)$$

Besides watermark awareness, we also evaluate the reconstruction quality of codecs through intrusive speech quality metrics like PSEQ, STOI and VISQOL. Additionally, we calculate the token match rate (TM) to probe how much the watermark affects the codec codes. TM is defined as the proportion of matched tokens between an unwatermarked audio and its watermarked version.

We test the TTS model’s ability to reject cloning watermarked voices through average silence ratio and speech quality metrics like word error rate (WER), speaker similarity (SIM) and mean opinion score (MOS). We use the ground-truth transcript and speaker embedding of the speech prompt as reference

System	WM	metric	ND	RS	BA	BP	HP	AAC	MP3	GN	PN	SM	EC	DA	EA	LP	FS	MEAN
O2	×	WER (%)	13.7	14.5	15.7	30.2	36.8	12.5	13.7	11.1	13.1	28.5	13.8	18.2	49.3	62.0	64.3	26.5
		SIM	0.86	0.86	0.86	0.79	0.80	0.86	0.84	0.84	0.84	0.78	0.83	0.85	0.81	0.72	0.36	0.79
	✓	WER (%)	46.6	44.3	46.4	66.4	66.3	45.7	58.4	55.7	51.1	62.3	47.3	31.9	63.8	61.6	37.8	52.4
		SIM	0.74	0.75	0.73	0.67	0.64	0.74	0.72	0.72	0.71	0.69	0.71	0.81	0.69	0.71	0.64	0.71

Table 3: Generated speech quality of our default system O2 under different watermark removal attacks. Tested on 100 utterances randomly sampled from LibriTTS-R test-clean. See Table 1 for the notation of each attack.

System	WM prompt	WER (%)	SIM	MOS	SR
B1	×	14.1	0.88	3.74	5.64
	✓	13.9	0.88	3.77	5.61
B2	×	17.8	0.88	3.52	5.36
	✓	18.9	0.87	3.21	5.46
O1	×	25.0	0.87	3.53	6.68
	✓	96.3	0.64	1.05	6.76
O2	×	15.0	0.85	3.64	17.30
	✓	51.0	0.74	1.27	57.31

Table 4: Generated speech quality of TTS models. SR denotes silence ratio in percentage, whereas WM denotes whether the speech prompt is watermarked.

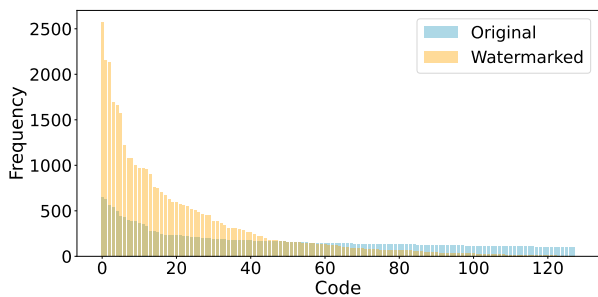


Figure 2: Codec code distributions of unwatermarked and watermarked audio, sorted by occurring frequency.

to calculate WER and SIM, respectively. Higher silence ratio and lower speech quality on watermarked prompts indicate that the model rejects copyrighted voices better.

3.3. Results

3.3.1. Codec

The watermark detection accuracies (ACC) under different kinds of watermark removal attacks are listed in Table 1. In most cases, the detection accuracy of our method O2 matches that of the original AudioSeal detector. Comparing O1 and O2, we can see that adding audio edits significantly improves robustness against attacks like smoothing (SM) and EnCodec (EC). Moreover, the False Negative Rates (FNR) under each kind of watermark removal attack are all 0% in O1 and O2, indicating that our method seldom rejects unwatermarked speech prompts.

We report the reconstruction quality of the watermark-aware codec in Table 2. Comparing B1 and O2, we see that the reconstruction quality of our method is slightly lower but still on par with the baseline trained without watermarked audio. This is within our expectations since the codec tends to output ambiguous segments if it is hard to determine whether the input segment is watermarked. Furthermore, there is no significant difference between the reconstruction quality of O1 and O2, indicating that adding audio edits during training does not affect the reconstruction quality of codec models.

3.3.2. TTS Model

In Table 4, we report the synthesis quality of TTS models given watermarked and unwatermarked prompts. Whether the speech prompt is watermarked or not, the generated speech quality of B1 and B2 remains almost the same. This indicates that both baseline TTS models do not have the ability to detect and reject watermarked prompts. In contrast, the synthesis quality of O1 and O2 significantly decreases when given watermarked speech prompts. Comparing O1 and O2, we can see that although O1 rejects watermarked prompts better, its synthesis quality falls behind that of B1 when given unwatermarked prompts. This indicates that audio edits added when training O2 not only improves its robustness against watermark removal attacks, but also contributes to maintaining the generation capacity of the underlying TTS model. We suppose that this is because the audio edits play a similar role as data augmentation.

To demonstrate the robustness of the entire TTS system against watermark removal attacks, we evaluate the synthesis quality of our default system, O2, using edited speech prompts. As shown in Table 3, for most attacks, there remains a significant quality gap between watermarked and unwatermarked voice cloning. Exceptionally, in the cases of echo (EA), low-pass filter (LP), and speed adjustment (FS), the speech prompts are so significantly altered that the TTS model is unable to produce speech of acceptable quality, even when the prompts are not watermarked. This means that these attacks are unsuccessful and can be ignored. In conclusion, our TTS system can still reject cloning voices in watermarked speech prompts, even if these prompts have been maliciously edited.

To further illustrate how our watermark-aware codecs affects the TTS model, we depict the first-layer codec code distribution of 100 unwatermarked/watermarked audio pairs from the LibriSpeech-test-clean dataset in Figure 2. The result reveals a significant difference in the code distribution of watermarked and unwatermarked speech, indicating a loss of speaker information when encoding watermarked speech. This explains why the speaker similarity drop when our system encounters a watermarked speech prompt.

4. Conclusion

In this paper, we propose a method to prevent open-source zero-shot TTS models from cloning copyrighted voices with watermark-aware codecs. Our experiments show that, by training the codec to “mute” in response to watermarked speech prompts, we can develop a watermark-aware TTS system that not only maintains high reconstruction quality but also exhibits robustness against various attacks. While our method may be generalized to a wide range of codecs and TTS models, it falls short when addressing copyrighted voices that are either unwatermarked or have been watermarked with unseen audio watermarking models. Future work may focus on enhancing the generalization ability and robustness of our method.

5. Acknowledgements

This study was supported in part by the Centre for Perceptual and Interactive Intelligence (CPII) Ltd., a CUHK-led InnoCentre under the InnoHK initiative of the Innovation and Technology Commission of the Hong Kong Special Administrative Region Government.

6. References

- [1] H. Wu, X. Chen, Y.-C. Lin, K.-w. Chang, H.-L. Chung, A. H. Liu, and H.-y. Lee, "Towards audio language modeling-an overview," *arXiv preprint arXiv:2402.13236*, 2024.
- [2] 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.
- [3] D. Yang, J. Tian, X. Tan, R. Huang, S. Liu, X. Chang, J. Shi, S. Zhao, J. Bian, X. Wu *et al.*, "Uniaudio: An audio foundation model toward universal audio generation," *arXiv preprint arXiv:2310.00704*, 2023.
- [4] L. Meng, L. Zhou, S. Liu, S. Chen, B. Han, S. Hu, Y. Liu, J. Li, S. Zhao, X. Wu *et al.*, "Autoregressive speech synthesis without vector quantization," *arXiv preprint arXiv:2407.08551*, 2024.
- [5] Z. Wu, J. Yamagishi, T. Kinnunen, C. Hanilçi, M. Sahidullah, A. Sizov, N. Evans, M. Todisco, and H. Delgado, "Asvspoof: the automatic speaker verification spoofing and countermeasures challenge," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 4, pp. 588–604, 2017.
- [6] G. G. Genelza, "A systematic literature review on ai voice cloning generator: A game-changer or a threat?" *Journal of Emerging Technologies*, vol. 4, no. 2, pp. 54–61, 2024.
- [7] C. Liu, J. Zhang, T. Zhang, X. Yang, W. Zhang, and N. Yu, "Detecting voice cloning attacks via timbre watermarking," *arXiv preprint arXiv:2312.03410*, 2023.
- [8] M. Li, Y. Ahmadiadli, and X.-P. Zhang, "A survey on speech deepfake detection," *ACM Computing Surveys*, 2025.
- [9] J. Yi, R. Fu, J. Tao, S. Nie, H. Ma, C. Wang, T. Wang, Z. Tian, Y. Bai, C. Fan *et al.*, "Add 2022: the first audio deep synthesis detection challenge," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 9216–9220.
- [10] H. Wu, Y. Tseng, and H.-y. Lee, "Codecfake: Enhancing anti-spoofing models against deepfake audios from codec-based speech synthesis systems," *arXiv preprint arXiv:2406.07237*, 2024.
- [11] X. Li, K. Li, Y. Zheng, C. Yan, X. Ji, and W. Xu, "Safeear: Content privacy-preserving audio deepfake detection," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 3585–3599.
- [12] N. Klein, T. Chen, H. Tak, R. Casal, and E. Khoury, "Source tracing of audio deepfake systems," in *Proc. Interspeech 2024*, 2024, pp. 1100–1104.
- [13] Y. Xie, X. Wang, Z. Wang, R. Fu, Z. Wen, S. Cao, L. Ma, C. Li, H. Cheng, and L. Ye, "Neural codec source tracing: Toward comprehensive attribution in open-set condition," *arXiv preprint arXiv:2501.06514*, 2025.
- [14] Y. Xie, R. Fu, Z. Wen, Z. Wang, X. Wang, H. Cheng, L. Ye, and J. Tao, "Generalized source tracing: Detecting novel audio deepfake algorithm with real emphasis and fake dispersion strategy," *arXiv preprint arXiv:2406.03240*, 2024.
- [15] R. Du, J. Yao, Q. Kong, and Y. Cao, "Towards out-of-distribution detection in vocoder recognition via latent feature reconstruction," *arXiv preprint arXiv:2406.02233*, 2024.
- [16] R. San Roman, P. Fernandez, H. Elshahar, A. Défossez, T. Furon, and T. Tran, "Proactive detection of voice cloning with localized watermarking," in *International Conference on Machine Learning*, vol. 235, 2024.
- [17] G. Chen, Y. Wu, S. Liu, T. Liu, X. Du, and F. Wei, "Wavmark: Watermarking for audio generation," *arXiv preprint arXiv:2308.12770*, 2023.
- [18] S. Ji, Z. Jiang, J. Zuo, M. Fang, Y. Chen, T. Jin, and Z. Zhao, "Speech watermarking with discrete intermediate representations," *arXiv preprint arXiv:2412.13917*, 2024.
- [19] H. Liu, M. Guo, Z. Jiang, L. Wang, and N. Z. Gong, "Audiomark-bench: Benchmarking robustness of audio watermarking," *arXiv preprint arXiv:2406.06979*, 2024.
- [20] J. Zhou, J. Yi, Y. Ren, J. Tao, T. Wang, and C. Y. Zhang, "Wm-codec: End-to-end neural speech codec with deep watermarking for authenticity verification," *arXiv preprint arXiv:2409.12121*, 2024.
- [21] X. Cheng, Y. Wang, C. Liu, D. Hu, and Z. Su, "Hifi-ganw: Watermarked speech synthesis via fine-tuning of hifi-gan," *IEEE Signal Processing Letters*, vol. 31, pp. 2440–2444, 2024.
- [22] R. S. Roman, P. Fernandez, A. Deleforge, Y. Adi, and R. Serizel, "Latent watermarking of audio generative models," *arXiv preprint arXiv:2409.02915*, 2024.
- [23] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *Transactions on Machine Learning Research*.
- [24] D. Yang, S. Liu, R. Huang, J. Tian, C. Weng, and Y. Zou, "Hifi-codec: Group-residual vector quantization for high fidelity audio codec," *arXiv preprint arXiv:2305.02765*, 2023.
- [25] Y. Koizumi, H. Zen, S. Karita, Y. Ding, K. Yatabe, N. Morioka, M. Bacchiani, Y. Zhang, W. Han, and A. Bapna, "Libritts-r: A restored multi-speaker text-to-speech corpus," *arXiv preprint arXiv:2305.18802*, 2023.
- [26] Y. Gao, N. Morioka, Y. Zhang, and N. Chen, "E3 tts: Easy end-to-end diffusion-based text to speech," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2023, pp. 1–8.