VoiceMark: Zero-Shot Voice Cloning-Resistant Watermarking Approach Leveraging Speaker-Specific Latents

Haiyun Li[1,2], *Zhiyong Wu*[1,2,*], *Xiaofeng Xie*[3], *Jingran Xie*[1], *Yaoxun Xu*[1], *Hanyang Peng*[2,*]

[1]Shenzhen International Graduate School, Tsinghua University, China
[2]Pengcheng Laboratory, China
[3]Independent Researcher, China

lihaiyun24@mails.tsinghua.edu.cn, zywu@sz.tsinghua.edu.cn, xiexiaofeng1926@gmail.com, {xjr21, xuyx22}@mails.tsinghua.edu.cn, penghy@pcl.ac.cn

Abstract

Voice cloning (VC)-resistant watermarking is an emerging technique for tracing and preventing unauthorized cloning. Existing methods effectively trace traditional VC models by training them on watermarked audio but fail in zero-shot VC scenarios, where models synthesize audio from an audio prompt without training. To address this, we propose VoiceMark, the first zero-shot VC-resistant watermarking method that leverages speaker-specific latents as the watermark carrier, allowing the watermark to transfer through the zero-shot VC process into the synthesized audio. Additionally, we introduce VC-simulated augmentations and VAD-based loss to enhance robustness against distortions. Experiments on multiple zero-shot VC models demonstrate that VoiceMark achieves over 95% accuracy in watermark detection after zero-shot VC synthesis, significantly outperforming existing methods, which only reach around 50%. See our code and demos at: https://huggingface.co/spaces/haiyunli/VoiceMark.

Index Terms: audio watermark, speech security, voice cloning

1. Introduction

Voice cloning (VC)-resistant watermarking is an emerging technique for tracing and preventing unauthorized cloning. Artists can embed such watermarks into their copyrighted recordings, ensuring that even if their voice is cloned into new audio, the watermark remains intact, thereby tracing and preventing unauthorized cloning. Recent research [1] has explored this application, confirming that most traditional VC models trained on watermarked audio will synthesize audio that retains the watermark. However, with the rapid development of zero-shot VC models such as CosyVoice [2], F5-TTS [3], and MaskGCT [4], highly realistic cloned audio can now be synthesized without requiring training, using only a few seconds of audio prompt. In this new scenario, we cannot embed watermarks into VC models through training, as the watermarked audio is directly used for inference, as illustrated in Figure 1. This means a zero-shot VC-resistant watermark is needed — one that can be directly transferred to the synthesized audio during zero-shot VC inference using a single watermarked audio prompt. Currently, zero-shot VC-resistant watermarking has not yet been studied.

Traditional audio watermarking has been developed over many years, with techniques such as spread spectrum [5] and echo hiding [6]. In recent years, more advanced deep learning-based approaches have been proposed [7, 8], demonstrating greater robustness compared to traditional methods. However, these watermarking methods are mainly designed to resist traditional audio editing, such as noise addition, filtering, and compression, with limited focus on resistance to VC models. Some methods [1, 9] propose VC-resistant watermarking techniques designed to resist traditional VC models. They validates that training traditional VC models [10, 11, 12] or commercial tools [13, 14] on watermarked audio enables robust watermark detection from the resulting synthesized audio. However, in the context of zero-shot VC-resistant watermarking, such methods face several limitations:

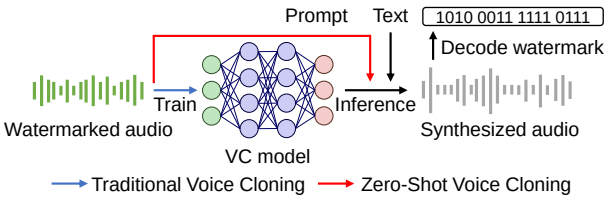


Figure 1: *Comparison of data flow for embedding watermarks in traditional voice cloning and zero-shot voice cloning.*

- **Inability to embed watermarks via training**: Zero-shot VC models eliminate the possibility of embedding watermarks into model parameters through training, causing existing VC-resistant watermarking methods to fail.
- **Failure to retain watermarks during synthesis**: Audio synthesized by zero-shot VC models exhibits significant differences from the prompt, including changes in content, length, and speed. These transformations disrupt or filter out the watermark, making it undetectable in the synthesized audio.

These limitations hinder the direct application of existing watermarking methods for traceability in zero-shot VC, exposing individuals' voices to the risk of unauthorized cloning.

To mitigate these risks, we aim to design a robust zero-shot VC-resistant watermarking method that does not rely on training VC models yet enables traceability for zero-shot VC. We observe that in zero-shot VC, to clone a speaker's voice, models must extract speaker-specific information (timbre, pitch, prosody, etc.) while discarding content from the audio prompt. This process typically occurs in the latent space compressed by neural codec-based methods [15, 16, 17] where the model implicitly or explicitly disentangles the speaker-specific latents to synthesize cloned audio. This implies that for higher speaker similarity, the model must consistently transfer speaker-specific latents from the original audio prompt to the synthesized audio.

Therefore, the key idea of our work is to leverage speaker-specific latents as the watermark carrier by adopting a neural codec model to disentangle them, embed the watermark, and reconstruct watermarked audio. The watermark is then transferred along with the speaker-specific latents to the synthesized

*Corresponding authors.

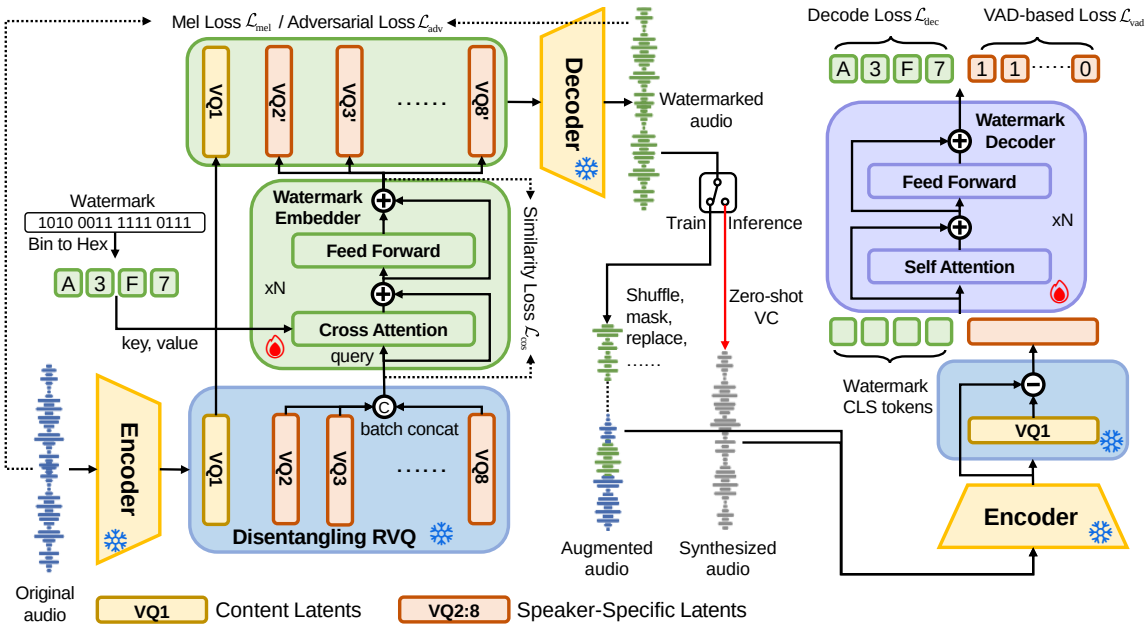


Figure 2: *The overall architecture of our proposed VoiceMark.*

audio during the zero-shot VC inference, thereby enabling robust zero-shot VC-resistant watermarking. To achieve this goal, we must address two primary challenges: 1) Speaker-specific latents span only speech-containing frames, while a large portion of the audio consists of silence and voiceless frames, which lack such latents. This makes conventional frame-by-frame embedding methods ineffective in leveraging speaker-specific latents. 2) The VC process alters content, duration, and speed while also distorting speaker-specific latents, causing the watermark to remain in only certain frames and potentially be incomplete. This makes accurate watermark detection challenging.

In this paper, we propose VoiceMark, the first zero-shot VC-resistant watermarking method that significantly improves resistance across multiple zero-shot VC models compared to state-of-the-art (SOTA) audio watermarking methods. Our main contributions are summarized as follows:

- VoiceMark introduces a watermark embedder model based on speaker-specific latents. It disentangles speaker-specific latents using a pre-trained residual vector quantization (RVQ) model, embed the watermark into these latents, and generates watermarked audio. We design a voice activity detection (VAD)-based loss function to guide the model in identifying frames containing speaker-specific latents and adaptively embedding the watermark through cross-attention mechanisms.
- VoiceMark proposes a robust watermark decoder model. To simulate the potential distortions of the zero-shot VC synthesis on the watermark, its training incorporates augmentation techniques, such as masking, shuffling, replacing, and neural encoding/decoding. Using a global transformer, the decoder recovers the watermark from the speaker-specific latents of the entire audio to enable robust detection.
- We conduct extensive experiments on multiple zero-shot VC models and traditional audio editing methods to evaluate the effectiveness of VoiceMark. Our results demonstrate that VoiceMark achieves over 95% accuracy in watermark detection after zero-shot VC synthesis, which significantly outperforms existing watermarking methods that reach around 50%.

2. Methodology

VoiceMark consists of three main components: an encoder-decoder RVQ model, a cross-attention-based watermark embedder, and a transformer-based watermark decoder, as shown in Figure 2. The RVQ model disentangles speaker-specific latents and reconstructs watermarked audio. The embedder embeds the watermark into speaker-specific latents. The decoder extracts the watermark from latents of the synthesized audio.

2.1. Disentangling Speaker-Specific Latents

The RVQ model, inspired by SpeechTokenizer [17], leverages HuBERT [18] latents as a semantic teacher to distill content latents into the first VQ layer, thereby disentangling speaker-specific latents into the remaining layers (VQ 2 to 8).

Given an input audio x, the encoder compresses it into a latent sequence $l = E(x)$, where $l \in \mathbb{R}^{t \times d}$, t is the number of frames, and d is the latent dimension. The sequence is then quantized by an 8-layer RVQ model, producing a set of quantized latents $\{z_1, z_2, \ldots, z_8\}$, where z_1 represents content latents and $\{z_2, \ldots, z_8\}$ correspond to speaker-specific latents. The content latents z_1 remain unmodified, while the speaker-specific latents are used for subsequent watermark embedding.

2.2. Watermark Embedding

Given the speaker-specific latents $\{z_2, \ldots, z_8\}$ and an n-bit watermark, the watermark is first converted into a hexadecimal sequence $w \in \{0, 1, \ldots, 15\}^{n/4}$ to reduce its length. The sequence w is then projected into the latent space of dimension d, resulting in $w' \in \mathbb{R}^{(n/4) \times d}$. The speaker-specific latents $z_2, \ldots, z_8$ and w' are then fed into a 4-layer cross-attention embedder $W_e(\cdot, \cdot)$ based on the transformer decoder [19], which produces the watermarked speaker-specific latents:

$$\{\hat{z}_2, \ldots, \hat{z}_8\} = W_e(\{z_2, \ldots, z_8\}, w'). \quad (1)$$

Here, $\{z_2, \ldots, z_8\}$ serve as queries, while w' serves as keys and values. After embedding the watermark, all latents are ag-

gregated as $\hat{z} = z_1 + \sum_{i=2}^{8} \hat{z}_i$, where $\hat{z}$ represents the watermarked latents. These are then passed into the decoder $D(\hat{z})$ to reconstruct the watermarked audio $\hat{x}$.

2.3. VC-Simulated Augmentation

The zero-shot VC synthesis process can significantly distort watermarked audio. To enhance the model's robustness, we introduce several augmentations that simulate these distortions:

1. **No watermark in silent frames**. We mask certain frames to zero with a probability of 20%.
2. **Completely different content**. We randomly shuffle the audio in 50 ms windows with a 50% probability.
3. **Partial watermark filtering**. We replace 50 ms segments with the original audio at a probability of 50%.
4. **Neural codec encoding and decoding**. We encode and decode the audio using EnCodec [15].
5. **Audio perturbation**. We apply speed perturbation, amplitude scaling, filtering, or resampling with 10% probability.

This process generates augmented audio $\tilde{x}$ to train the model and improve its robustness against distortions.

2.4. Watermark Decoding

The watermark decoder, denoted as $W_d(\cdot)$, is a transformer model for extracting the watermark from speaker-specific latents. During inference, it processes audio synthesized by zero-shot VC models, while during training, it learns from augmented audio $\tilde{x}$ without exposure to any VC models. The input audio is first compressed into latents l. The content latents from the first VQ layer (z_1) are subtracted from l to obtain the speaker-specific latents, denoted as $l_s = l - z_1$.

For watermark decoding, we use learnable CLS tokens, denoted as $c \in \mathbb{R}^{(n/4)\times d}$, corresponding to the $n/4$-length hexadecimal watermark. These tokens are concatenated with l_s and fed into $W_d(\cdot)$, which employs self-attention to extract watermark information from l_s:

$$(\hat{w}, \hat{p}) = W_d(c \oplus l_s), \tag{2}$$

where $\oplus$ denotes the concatenation operation, $\hat{w} \in \mathbb{R}^{(n/4)\times 16}$ is the decoded hexadecimal watermark, with each row representing a softmax probability distribution over 16 categories, $\hat{p} \in \mathbb{R}^t$ is the sigmoid probability of a frame containing both speech and a watermark ($\hat{p} = 1$) or neither ($\hat{p} = 0$). The decoded hexadecimal watermark is first processed with argmax, then converted into an n-bit binary sequence.

2.5. Training Loss

We design multiple loss functions to ensure accurate watermark detection while preserving audio quality:

VAD-based Loss. We use a dual-threshold VAD method [20] to compute binary cross entropy $\mathcal{L}_{\text{vad}}$ on $\hat{p}$. Frames containing both speech and a watermark are labeled as 1, while silent, masked or replaced frames are labeled as 0. This guides the embedder and decoder to focus on watermarked speech.

Quality Loss. We preserve speaker consistency with a cosine similarity loss $\mathcal{L}_{\text{cos}}$ between speaker-specific latents before and after watermark embedding. For perceptual quality, we apply a multi-scale Mel spectrogram loss $\mathcal{L}_{\text{mel}}$ and an adversarial loss $\mathcal{L}_{\text{adv}}$ to refine the reconstructed audio $\hat{x}$ [15].

Decoding Loss. The watermark decoding loss $\mathcal{L}_{\text{dec}}$ is computed using the cross-entropy loss on $\hat{w}$, ensuring accurate recovery of the hexadecimal watermark.

The total loss is a weighted sum of these components:

$$\mathcal{L} = \lambda_{\text{vad}}\mathcal{L}_{\text{vad}} + \lambda_{\text{cos}}\mathcal{L}_{\text{cos}} + \lambda_{\text{mel}}\mathcal{L}_{\text{mel}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}} + \lambda_{\text{dec}}\mathcal{L}_{\text{dec}}, \tag{3}$$

where $\lambda_{\text{vad}}, \lambda_{\text{cos}}, \lambda_{\text{mel}}, \lambda_{\text{adv}}, \lambda_{\text{dec}}$ are hyperparameters.

3. Experiments

3.1. Implementation Details

For the RVQ model, we use the pretrained SpeechTokenizer*. W_e employs a 4-layer, 1-head, 256-dimensional Transformer Decoder, while W_d adopts an 8-layer, 1-head, 512-dimensional Transformer Encoder [19]. The watermark bit length is set to 16. Hyperparameters are set as: $\lambda_{\text{vad}} = 1$, $\lambda_{\text{cos}} = 2$, $\lambda_{\text{mel}} = 2$, $\lambda_{\text{adv}} = 1$, and $\lambda_{\text{dec}} = 1$, with larger weights for λ_{cos} and λ_{mel} to preserve audio quality. The model is trained for 30 epochs using Adam [21] optimizer with a learning rate of $5e^{-5}$.

3.2. Baselines

We compare VoiceMark with three SOTA watermarking methods: AudioSeal [8], WavMark [7], and Timbre [1]. The related method [9] is not included since the code is not public.

3.3. Datasets

We use the VCTK [22] (train and test) and Librispeech [23] (test) datasets. The test set consists of 2,000 unseen VCTK samples and 2,600 Librispeech samples, totaling 4,600 samples from 150 speakers.

3.4. Metrics

We evaluate watermark detection using two metrics: bitwise accuracy (ACC) and False Attribution Rate (FAR). ACC is the ratio of correctly decoded bits to the total number of bits. FAR simulates real-world multi-candidate identification by comparing the Hamming distance of each decoded watermark to 100 candidates (1 ground truth and 99 random). A false attribution occurs if the closest match is not the ground truth.

For audio quality, we assess objective metrics including perceptual evaluation of speech quality (PESQ) [24], scale-invariant signal-to-noise ratio (SI-SNR), and short-time objective intelligibility (STOI) [25]. Subjective evaluation is conducted using similarity mean opinion score (SMOS), where 15 subjects rate 40 samples on a 1-5 scale. A SMOS of 4 or higher indicates high similarity to the original audio.

3.5. Performance Evaluation

Table 1 compares VoiceMark's watermark detection performance with other SOTA methods under zero-shot VC models (CosyVoice [2], F5-TTS [3], and MaskGCT [4]) and traditional audio editing. The text prompts for zero-shot VC are randomly selected from the test set.

For zero-shot VC models, VoiceMark consistently outperforms all methods. Other approaches exhibit an ACC close to 0.5 and an FAR near 1.0, indicating near-random decoding in zero-shot VC scenarios. This demonstrates that existing methods fail to retain the watermark in zero-shot VC, whereas VoiceMark ensures robust watermark traceability.

For traditional audio editing, we use AudioCraft's implementation† with default parameters, where Amplitude is the average of boost and duck, and Filter is the average of band, high,

*https://huggingface.co/fnlp/SpeechTokenizer

†https://github.com/facebookresearch/audiocraft

Table 1: *Performance Evaluation on Zero-Shot VC Models and Traditional Editing.*

Method	Zero-Shot VC Models (ACC ↑ / FAR ↓)			Traditional Editing (ACC ↑ / FAR ↓)					
	CosyVoice [2]	F5-TTS [3]	MaskGCT [4]	EnCodec [15]	Resample	Amplitude	Filter	White	MP3
AudioSeal (2024) [8]	0.508/0.979	0.513/0.977	0.506/0.973	0.936/0.052	1.000/0.000	1.000/0.000	1.000/0.000	1.000/0.000	1.000/0.000
WavMark (2023) [7]	0.499/1.000	0.499/1.000	0.499/1.000	0.498/1.000	1.000/0.000	1.000/0.000	0.711/0.289	0.938/0.058	1.000/0.000
Timbre (2024) [1]	0.499/0.981	0.539/0.954	0.527/0.966	0.696/0.726	1.000/0.000	1.000/0.000	0.989/0.022	1.000/0.000	1.000/0.000
VoiceMark (Ours)	**0.964/0.112**	**0.979/0.070**	**0.957/0.141**	**0.985/0.044**	0.988/0.049	0.995/0.014	0.987/0.036	0.965/0.109	0.973/0.102

and low-pass filters. The results show that VoiceMark performs comparably to other methods and outperforms them after EnCodec processing. We observe that VoiceMark doesn't achieve 1.0 ACC, likely due to our training solely on the VCTK dataset. VCTK has many speakers suitable for VC-related tasks, but it is a relatively small, clean dataset, and the distorted audio from editing may degrade our performance.

Table 2: *Ablation Study*

Method	ACC ↑	FAR ↓
AS-Emb + VM-Dec	0.663	0.906
VoiceMark	**0.964**	**0.112**
w/o VAD-base Loss	0.478	0.980
w/o Augmentation	0.626	0.924

3.6. Ablation Study

VoiceMark incorporates three key innovations: speaker-specific latents watermarking, VC-simulated augmentation, and VAD-based loss. We conduct ablation studies on CosyVoice [2] to validate their necessity, as shown in Table 2.

To assess the impact of speaker-specific latents watermarking, we keep other modules unchanged but replace the watermark embedder with the AudioSeal embedder [8] (AS-Emb + VM-Dec), which is based on a generic architecture, where the synthesized watermark is directly added to the original waveform. Results show that our latent-based design significantly enhances zero-shot VC resistance. Additionally, removing VC-simulated augmentation and VAD-based loss causes a performance drop, confirming the necessity for robust watermarking.

3.7. Audio Quality Assessment

Table 3 presents the audio quality evaluation results for different methods. AudioSeal directly adds the generated watermark to the original audio waveform, while EnCodec and SpeechTokenizer are neural codec models that reconstruct the audio from a latent space. VoiceMark, leveraging a neural codec architecture for watermarking, integrates the watermarking process within the codec framework.

The results show that while VoiceMark's audio quality is lower than AudioSeal's, it performs comparably to SpeechTokenizer and significantly outperforms EnCodec.

3.8. Case Study

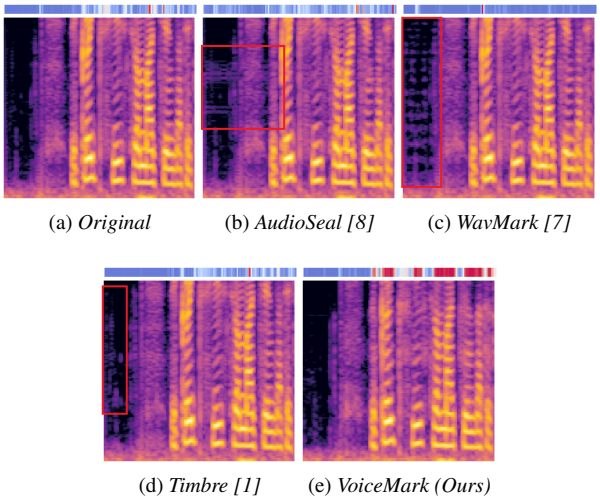
To directly observe the impact of watermarking on audio, we visualize the mel spectrograms of different watermarking methods, as shown in Figure 3.

VoiceMark embeds the watermark within speaker-specific latents, subtly altering harmonics and formants, making it more difficult for attackers to detect. In contrast, spectrograms from other methods exhibit visible watermarking artifacts in certain frequency bands (highlighted in red boxes), which attackers can exploit to detect or remove the watermark.

In addition, we visualize watermark probability detected by VoiceMark as a top color band, with red for high and blue for low probability. In VoiceMark's watermarked audio, the detected probability aligns precisely with speech segments, while no watermark appears in other samples, confirming its effective embedding within speech features.

Table 3: *Audio Quality Assessment. (W): General watermarking directly added to original waveforms. (N): Neural codec models. (W-N): Watermarking using neural codec architecture.*

Method	PESQ ↑	SI-SNR ↑	STOI ↑	SMOS ↑
AudioSeal (W) [8]	4.32	26.69	0.99	4.67±0.10
EnCodec (N) [15]	1.62	-0.62	0.80	2.21±0.15
SpeechTokenizer (N) [17]	2.58	1.64	0.89	4.63±0.10
VoiceMark (W-N)	2.20	2.01	0.89	4.25±0.13



(a) *Original* (b) *AudioSeal [8]* (c) *WavMark [7]*

(d) *Timbre [1]* (e) *VoiceMark (Ours)*

Figure 3: *Visualization of Mel Spectrograms.*

4. Conclusion

In this work, we propose VoiceMark, the first zero-shot VC-resistant watermarking method that embeds watermarks into speaker-specific latents to achieve resistance to zero-shot VC models. Additionally, we incorporate VC-simulated augmentations and VAD-based loss to further enhance the robustness of VoiceMark. Experimental results show that VoiceMark significantly outperforms SOTA watermarking methods in retaining watermarks across multiple zero-shot VC models.

5. Acknowledgements

This work is supported by National Natural Science Foundation of China (62076144).

6. References

[1] C. Liu, J. Zhang, T. Zhang, X. Yang, W. Zhang, and N. Yu, "Detecting voice cloning attacks via timbre watermarking," in *Network and Distributed System Security Symposium*, 2024.

[2] Z. Du, Q. Chen, S. Zhang, K. Hu, H. Lu, Y. Yang, H. Hu, S. Zheng, Y. Gu, Z. Ma *et al.*, "Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens," *arXiv preprint arXiv:2407.05407*, 2024.

[3] Y. Chen, Z. Niu, Z. Ma, K. Deng, C. Wang, J. Zhao, K. Yu, and X. Chen, "F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching," *arXiv preprint arXiv:2410.06885*, 2024.

[4] Y. Wang, H. Zhan, L. Liu, R. Zeng, H. Guo, J. Zheng, Q. Zhang, X. Zhang, S. Zhang, and Z. Wu, "Maskgct: Zero-shot text-to-speech with masked generative codec transformer," in *ICLR*. OpenReview.net, 2025.

[5] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Systems Journal*, p. 313–336, Jan 1996. [Online]. Available: http://dx.doi.org/10.1147/sj.353.0313

[6] D. Gruhl, A. Lu, and W. Bender, *Echo hiding*, Jan 1996, p. 295–315. [Online]. Available: http://dx.doi.org/10.1007/3-540-61996-8_48

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

[8] R. San Roman, P. Fernandez, H. Elsahar, A. Défossez, T. Furon, and T. Tran, "Proactive detection of voice cloning with localized watermarking," in *International Conference on Machine Learning*, vol. 235, 2024.

[9] Q. Li and X. Lin, "Proactive audio authentication using speaker identity watermarking," in *2024 21st Annual International Conference on Privacy, Security and Trust (PST)*. IEEE, 2024, pp. 1–10.

[10] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5530–5540.

[11] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," in *International Conference on Learning Representations*.

[12] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[13] H. Zhang, T. Yuan, J. Chen, X. Li, R. Zheng, Y. Huang, X. Chen, E. Gong, Z. Chen, X. Hu *et al.*, "Paddlespeech: An easy-to-use all-in-one speech toolkit," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: System Demonstrations*, 2022, pp. 114–123.

[14] "Voice cloning app," https://github.com/BenAAndrew/Voice-Cloning-App, 2023.

[15] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.

[16] Z. Ju, Y. Wang, K. Shen, X. Tan, D. Xin, D. Yang, E. Liu, Y. Leng, K. Song, S. Tang *et al.*, "Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models," in *Forty-first International Conference on Machine Learning*.

[17] X. Zhang, D. Zhang, S. Li, Y. Zhou, and X. Qiu, "Speechtokenizer: Unified speech tokenizer for speech large language models," *arXiv preprint arXiv:2308.16692*, 2023.

[18] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 29, pp. 3451–3460, 2021.

[19] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[20] L. R. Rabiner, *Digital processing of speech signals*. Pearson Education India, 1978.

[21] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] J. Yamagishi, "English multi-speaker corpus for cstr voice cloning toolkit," 2012. [Online]. Available: https://datashare.ed.ac.uk/handle/10283/3443

[23] 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)*. IEEE, 2015, pp. 5206–5210.

[24] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221)*, vol. 2. IEEE, 2001, pp. 749–752.

[25] 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.