



CodecFake: Enhancing Anti-Spoofing Models Against Deepfake Audios from Codec-Based Speech Synthesis Systems

Haibin Wu^{1,2}, Yuan Tseng^{1,2}, Hung-yi Lee^{1,2}

¹Speech Processing and Machine Learning Laboratory, National Taiwan University

²Graduate Institute of Communication Engineering, National Taiwan University

f07921092@ntu.edu.tw

Abstract

Current state-of-the-art (SOTA) codec-based audio synthesis systems can mimic anyone’s voice with just a 3-second sample from that specific unseen speaker. Unfortunately, malicious attackers may exploit these technologies, causing misuse and security issues. Anti-spoofing models have been developed to detect fake speech. However, the open question of whether current SOTA anti-spoofing models can effectively counter deepfake audios from codec-based speech synthesis systems remains unanswered. In this paper, we curate an extensive collection of contemporary SOTA codec models, employing them to re-create synthesized speech. This endeavor leads to the creation of CodecFake, the first codec-based deepfake audio dataset. Additionally, we verify that anti-spoofing models trained on commonly used datasets cannot detect synthesized speech from current codec-based speech generation systems. The proposed CodecFake dataset empowers these models to counter this challenge effectively.

Index Terms: deepfake detection, speech synthesis, neural audio codec, speech language model, anti-spoofing

1. Introduction

Neural audio codecs were first proposed to allow better compression of audio data, thus reducing file storage sizes and improving transmission latency significantly. Researchers have also found that neural audio codecs serve as effective tokenizers for converting continuous speech into discrete codes suitable for autoregressive speech generation tasks, akin to the role of language modeling in natural language processing. Consequently, the past three years have seen substantial development of codec-based audio synthesis systems [1–13]. Current state-of-the-art codec-based audio synthesis systems [5–10, 14] have advanced to the point where they can accurately mimic someone’s voice with just a 3-second recording from an unseen speaker. Unfortunately, these advancements raise concerns about potential misuse and security issues, as malicious attackers may exploit these technologies to impersonate others’ speech.

The ASVspoof challenge, as cited in [15–18], serves the purpose of attracting international teams to participate in mitigating synthetic speech using anti-spoofing models. Anti-spoofing models are designed to perform binary classification tasks, specifically identifying and discarding fake audio. In recent years, there has been remarkable progress in the development of high-performance anti-spoofing models [15–21] for traditional speech synthesis systems. However, codec-based speech synthesis systems are relatively new, and whether current state-of-the-art anti-spoofing models can effectively counter deepfaked audios generated by codec-based speech synthesis systems remains an open question.

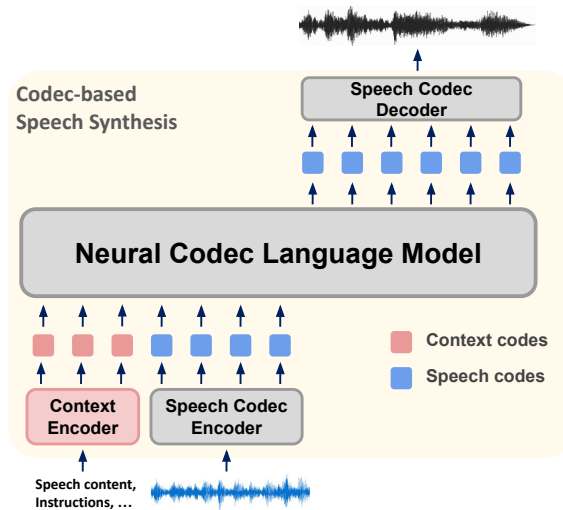


Figure 1: Codec-based speech generation pipeline.

In this paper, we assemble an exhaustive collection of state-of-the-art contemporary codec models, employing them to re-synthesize speech. The collection includes six diverse mainstream codec papers, each characterized by distinct training implementations, yielding a total of 15 unique codec models. This undertaking results in CodecFake, which marks the inception of the first-ever codec-based deepfake audio dataset.¹ Furthermore, we conduct a comprehensive analysis, carefully assessing the effectiveness of cutting-edge anti-spoofing models when confronted with synthesized speech generated by cutting-edge neural audio codec models. In addition, we also leverage the CodecFake dataset to create enhanced anti-spoofing models for detecting codec-based deepfaked audios.

We find that: (1) Models trained on the ASVspoof dataset struggle to detect not only the synthesized speech from codec models but also from current codec-based speech generation models. (2) Our proposed CodecFake dataset effectively enhances models’ ability to tackle synthesized speech from current state-of-the-art codec-based speech generation systems (equal error rate 0.4% for synthesized speech from VALL-E, a state-of-the-art codec-based speech generation model).

We will release all code and data², and we aspire that this study and the CodecFake dataset will contribute to mitigating malicious attacks stemming from current advanced codec-based speech synthesis systems.

¹We note that a concurrent preprint with similar ideas [22] was submitted to arXiv during the Interspeech 2024 anonymity period.

²<https://codecfake.github.io>

Table 1: Details of adopted neural audio codec models. "A-F" represent different codec frameworks, N_c represents the number of codebooks, SR represents audio sampling rate (in kHz), and BPS represents bits per second.

	Name	Training Data	N_c	SR	BPS
A	16k	Librispeech	8	16	4
B1	hifi_16k_320d	LibriTTS	4	16	2
B2	hifi_16k_320d_large_uni	VCTK	4	16	2
B3	hifi_24k_320d	AISHHELL	4	24	3
C	24k_320d	Valentini	8	24	6.4
D1	16k	Common Voice, DAPS	12	16	6
D2	24k	MUSDB, VCTK	32	24	24
D3	44k	AudioSet, Jamendo	9	44.1	8
E	24k	Common Voice DAPS, Jamendo AudioSet, FSD50K	8	24	1.5
F1	en_libritts_16k_gr1nq32ds320		32	16	16
F2	en_libritts_16k_gr8nq32ds320	LibriTTS	32	16	16
F3	en_libritts_16k_nq32ds320	Subset	32	16	16
F4	en_libritts_16k_nq32ds640		32	16	8
F5	zh_en_16k_nq32ds320	In-house, 25k hrs	32	16	16
F6	zh_en_16k_nq32ds640	(en & zh-cn)	32	16	8

2. CodecFake

This section presents the dataset creation procedure, design principles, and the rationale behind the CodecFake dataset.

2.1. Codec-based speech generation

As depicted in Figure 1, codec-based speech generation starts with converting context information, such as the desired speech content or textual instructions, into discrete context codes while encoding audio into discrete speech codes at the same time. These context and speech codes are then fed into the codec language model to generate a target speech code sequence. Lastly, the target speech code sequence is sent to the speech codec decoder to synthesize the final output waveform.

Various designs exist for codec language models (center gray block in Figure 1) [2–13], resulting in numerous variations. To mitigate the impact of these variations and preserve the sole invariant factor of the codec-based speech generation, we exclusively utilize the codec resynthesis procedure to generate our proposed CodecFake dataset. This process involves encoding speech into discrete codes and then decoding these codes back into speech. Experimental results in Table 3 verify that the CodecFake dataset, generated according to this principle, enhances anti-spoofing models to catch deepfaked speech produced by state-of-the-art codec-based speech synthesis systems.

2.2. Codec adoption

To ensure our generated dataset achieves extensive coverage and comprehensiveness, we systematically gather all current open-source codec models spanning six distinct types, each with its unique training settings. This effort yields a total of 15 distinct codec models, summarized in Table 1.

Encodec (E) [23] stands as one of the pioneering implementations of neural codec models, embodying a classic neural codec architecture comprising encoder, quantizer, and decoder mod-

Table 2: Statistics of various data sources used.

Data Source		Training	Validation	Testing
		# utts. / # spks.		
<i>Dataset Generation</i>				
CodecFake	VCTK	42752 / 103	735 / 2	755 / 2
	VCTK	42752 / 103	735 / 2	755 / 2
<i>Zero-shot Evaluation</i>				
ASVspooF	ASVspooF 2019	N/A	N/A	71237 / 67
	VALL-E	N/A	N/A	755 / 2
VALL-E X	Demo Page	N/A	N/A	36 / 36
SpeechX	Demo Page	N/A	N/A	12 / 12

ules. It further augments its capabilities by integrating supplementary LSTM layers and harnessing a Transformer-based language model to model residual vector-quantized codes, thereby amplifying its sequence modeling performance.

SpeechTokenizer (A) [24] is a unified speech tokenizer designed for speech-language models. Similar to Encodec, it also implements an encoder-decoder architecture enhanced with residual vector quantization (RVA). By integrating semantic and acoustic tokens, SpeechTokenizer hierarchically separates various facets of speech information across different RVQ layers. Specifically, SpeechTokenizer is designed to regularize the first RVQ layer to learn Hubert tokens [25]. The authors claim that employing such techniques can lead to improved disentanglement of information across various RVQ layers.

AcademiaCodec (B) [26] proposes a group-residual vector quantization technique, tailored explicitly for generation tasks. It aims to enhance the reconstruction performance using a limited number of codebooks, consequently achieving an impressively low bit rate per second (BPS). This low BPS is of utmost significance as it effectively addresses the challenge of lengthy speech tokens in speech-language modeling, resulting in reduced sequence lengths.

AudioDec (C) [27] represents an enhanced version of Encodec, implementing a group convolution mechanism to facilitate real-time operation of the streamable network, while also harnessing the capabilities of HiFiGAN [28] to effectively generate high-fidelity audio at a sampling rate of 48 kHz.

Descript Audio Codec (D) [29] combines a multitude of training techniques, such as periodic activation functions [30], enhanced residual vector quantization using factorized and L2-normalized codes, random quantizer dropout to preserve audio reconstruction quality, as well as refining adversarial and reconstruction loss during the training process. Out of all the techniques employed, they emphasize the pivotal role played by the periodic activation function.

FunCodec (F) [31] proposes a frequency-domain codec. The authors claim they can achieve comparable performance with fewer parameters and lower computation complexity. Meanwhile, they also find incorporating semantic information in the codec tokens improves speech quality at low bit rates.

2.3. Dataset generation pipeline

The most famous anti-spoofing dataset, the ASVspooF 2019 database [16], is based on the Voice Cloning Toolkit (VCTK) corpus [32]. Following the methodology of ASVspooF 2019, we generate the CodecFake dataset based on the VCTK corpus. The set of 107 speakers is divided into three speaker-disjoint

		Test Subset															
		A	B1	B2	B3	C	D1	D2	D3	E	F1	F2	F3	F4	F5	F6	
Training Subset	AASIST	43.71	49.01	50.86	47.42	36.42	49.40	46.62	15.89	5.56	34.04	40.26	35.63	50.86	48.21	48.48	60
	AASIST-L	43.18	36.29	40.66	43.71	47.42	50.01	54.30	40.53	13.51	41.59	42.38	36.56	55.89	45.96	47.15	60
	A	0.00	1.06	0.66	12.98	5.43	16.16	45.96	22.78	0.66	5.03	19.47	9.80	5.96	16.95	17.09	50
	B1	36.16	1.46	3.58	4.64	30.99	36.56	46.62	42.12	29.27	46.36	39.60	25.03	24.90	34.57	34.57	50
	B2	40.13	3.44	2.52	6.23	34.97	40.40	49.93	45.83	39.74	54.97	54.57	33.11	33.51	46.36	46.62	40
	B3	51.66	5.43	6.75	4.64	67.42	50.07	49.14	49.93	52.19	57.48	51.79	49.01	59.34	50.33	50.86	40
	C	35.89	27.28	32.58	50.60	0.00	40.00	52.58	47.02	6.89	14.04	19.07	24.77	2.91	26.75	26.75	40
	D1	40.26	3.58	7.68	20.53	52.85	0.00	16.29	6.36	13.25	4.64	31.66	24.11	1.72	42.25	42.78	30
	D2	47.02	19.87	34.70	26.49	50.33	1.85	0.93	12.45	25.43	8.74	29.67	4.24	1.59	15.10	14.83	30
	D3	26.89	0.40	0.66	2.78	54.44	0.13	6.23	0.13	5.83	5.56	58.28	61.46	14.83	78.94	79.07	30
	E	35.23	31.79	33.64	34.44	19.87	50.20	50.60	50.86	0.00	12.72	34.44	39.34	40.79	60.26	60.53	20
	F1	45.03	49.80	43.44	58.94	26.62	56.56	53.51	57.35	1.32	0.00	8.21	45.03	32.05	41.32	41.19	20
	F2	50.07	43.31	45.03	56.56	20.00	65.70	52.19	61.32	14.97	5.43	0.00	1.19	3.44	10.99	10.73	10
	F3	41.59	52.05	53.64	54.04	30.86	38.81	40.13	51.92	27.42	28.48	2.52	0.00	0.13	2.52	2.52	10
	F4	45.96	34.17	36.56	39.07	19.21	44.90	48.21	50.73	26.09	44.24	23.71	6.75	0.00	43.84	44.24	10
	F5	51.52	52.98	55.36	56.42	35.10	49.01	37.22	48.74	60.26	17.35	5.70	0.79	2.91	0.00	0.00	10
F6	58.01	60.40	61.72	60.40	38.54	61.72	38.81	54.70	71.39	18.68	6.09	0.53	4.37	0.00	0.00	10	

Figure 2: Cross-testing results of anti-spoofing models on CodecFake test subsets. We report equal error rate (EER), where lower is better. Row AASIST and Row AASIST-L are baselines trained on the ASVspoof 2019 logical access training set using different architectures, while rows A-F6 are AASIST-L models trained on one training subset of CodecFake. Columns A-F6 indicate the specific CodecFake test subset evaluated.

sets for training, validation, and testing. Speakers p226 and p229 are allocated for generating validation data, while speakers p227 and p228 are reserved for generating testing data. The remaining speakers are used to generate training data.

We use 6 different neural audio codec frameworks, with a total of 15 different pre-trained codec models to re-synthesis speech based on VCTK. As shown in 1, a typical neural audio codec model consists of an encoder and a decoder. Using a pre-trained codec encoder, we first encode real speech into discrete codes. With these discrete codes as input, we re-synthesize speech using each respective codec decoder. CodecFake contains 15 subsets, generated by an existing pre-trained neural audio codec model in Table 1. Each subset shares the same statistics as VCTK, with training, validation, and testing data. Statistics are shown in *Dataset Generation* part of Table 2.

3. Experimental setup

3.1. Implementation details

We adopt AASIST-L [33]³, a state-of-the-art deepfake detection model, as our backbone model to train the anti-spoofing models. We choose AASIST-L rather than AASIST as the backbone, because AASIST-L is less prone to overfitting. This model architecture is leveraged to get well-trained models on

³<https://github.com/clovaai/aasist>

both the ASVspoof and CodecFake data. For the ASVspoof data, we utilize the author’s released checkpoints, as they have already conducted a thorough hyperparameter search. For CodecFake data, we do a careful hyperparameter search for the batch size and learning rate, to achieve optimal results. The final learning rate is 0.001 with an Adam optimizer, and the batch size is 16. During training, we combine one of the 15 CodecFake subsets with the corresponding original VCTK utterances as training data. Then, we employ the respective CodecFake validation subset for model selection.

3.2. Evaluation details

We consider four evaluation sets: (1) ASVspoof 2019 evaluation set; (2) CodecFake evaluation set; (3) VCTK evaluation set synthesized using an open-source reimplementation of VALL-E [5]⁴; (4) Synthesized and ground truth audio samples collected from the demo pages of codec-based speech synthesis systems, including VALL-E X [6]⁵ and SpeechX [10]⁶. We resort to collecting audio samples from VALL-E X and SpeechX demo pages because they do not release pretrained models for inference due to concerns about potential malicious use. Potential malicious deepfake attacks may even impact open-source

⁴https://huggingface.co/amphion/valle_librilight_6k

⁵<https://www.microsoft.com/en-us/research/project/vall-e-x/>

⁶<https://www.microsoft.com/en-us/research/project/speechx/>

Table 3: Comparison of fake speech detection performance (EER) of models trained on ASVspoof and CodecFake datasets. Models that outperform those trained on ASVspoof (verified by tests of significance) are highlighted in bold. Best-performing models are colored in gray background.

	ASVspoof (All)		CodecFake														
	AASIST	AASIST-L	A	B1	B2	B3	C	D1	D2	D3	E	F1	F2	F3	F4	F5	F6
VALL-E*	9.93	13.91	0.53	31.79	35.50	58.94	5.30	4.50	16.95	2.65	0.40	1.99	32.72	41.85	21.06	42.52	56.29
VALL-E X†	36.11	33.33	25.00	52.78	58.33	50.00	41.67	11.11	22.22	13.89	5.56	22.22	44.44	41.67	50.00	50.00	58.33
SpeechX†	33.33	25.00	50.00	66.67	58.33	58.33	41.67	16.67	50.00	41.67	8.33	16.67	41.67	50.00	50.00	50.00	50.00

* Implementation from https://huggingface.co/amphion/valle_librilight_6k; † Audio collected from respective demo pages.

efforts, underscoring the significance of our research. Once deepfake attacks become detectable, researchers may become more willing to release their models. For demo page audio samples, we concentrate solely on tasks that generate speech with minimal noise, excluding audio generation and samples with significant distortion. Our evaluation primarily focuses on samples intended for speaker-conditioned text-to-speech synthesis, speaker-conditioned speech-to-speech translation, speech editing, voice conversion, etc. We collect pairs of real and fake speech. Real speech comes from the ground truth or the speaker prompt, whereas model-generated speech is categorized as fake speech. The statistics of the four evaluation sets used for testing are shown in the *Zero-shot Evaluation* part of Table 2. We adopt equal error rate for evaluation, as it is the mainstream metric to evaluate anti-spoofing models [16].

4. Experimental results

4.1. Detecting speech generated by neural audio codecs

The cross-testing results (EERs. Lower is better) are shown in Figure 2. The rows denote the training data subsets (CodecFake and ASVspoof), and the columns denote the CodecFake testing data subsets. We have the following observations:

- Models (AASIST and AASIST-L) trained on the ASVspoof dataset exhibit significantly high EERs (usually higher than 30%) across most of the CodecFake testing subsets. This suggests that models trained on ASVspoof struggle to detect synthesized speech from codec models.
- Typically, training on a specific CodecFake training subset leads to the lowest EER for the corresponding CodecFake testing subset (e.g., row A as training and column A as testing; row B1 as training and column B1 as testing). Training on a particular type of CodecFake training subset generally results in relatively low EERs for the corresponding type of CodecFake testing subset (e.g., row B1 as training, column B1, B2 and B3 as testing). A possible reason is that the in-distribution testing conditions will perform better than out-of-distribution testing conditions.
- The model trained on training subset A (row A) demonstrates good generalizability to most of the testing subsets (columns A to F6). In SpeechTokenizer (A) [24], authors introduce a technique to allocate different types of information across different layers of codebooks. This disentanglement of information may contribute to the above generalizability.

4.2. Detecting speech generated by codec-based speech synthesis systems

Table 3 illustrates the testing results for models trained by different kinds of training datasets. The columns denote which training set is adopted for training. For ASVspoof, all the train-

ing data are adopted to train two model variants, AASIST and AASIST-L. For the CodecFake dataset, each column denotes which training subset is adopted to train the AASIST-L model. The rows denote the sources of testing sets. We have the following observations:

- Models trained with the ASVspoof dataset do not yield satisfactory performance on the three codec-based speech synthesis systems. This is likely because ASVspoof is comprised of speech generated by traditional speech synthesis models, which differ significantly from current codec-based systems. This underscores the urgent need for a codec-based deepfake detection dataset.
- VALL-E claims to be able to replicate anyone’s voice with just three seconds of speech, which raises serious concerns about potential malicious impersonation. However, we find that model trained from CodecFake train subsets A and E effectively detect fake speech generated by VALL-E, with remarkably low equal error rates (0.53% and 0.4%). For demo samples from VALL-E X and SpeechX, models trained on CodecFake train subset E achieve the best EER. We also note that models trained on subsets D1 and F1 also exhibit better detection performance than ASVspoof baselines.
- We perform statistical tests to verify results in Table 3 are statistically significant. For each testing source, we compare classifier predictions from models trained on each CodecFake train subset to prediction from the best performing ASVspoof classifier. With a null hypothesis of identical equal error rates, we conduct two tests, paired sample t-test and Wilcoxon signed-rank test, using a p-value threshold of 0.01. For VALL-E and VALL-E X, models highlighted in bold significantly outperform the classifiers trained on ASVspoof. For SpeechX, due to the small number of data samples, we cannot verify whether gains achieved are meaningful.

5. Conclusion

In this study, we assembled a comprehensive collection of contemporary codec models, resulting in the creation of the CodecFake dataset. Our extensive experiments verify that while anti-spoofing models trained on the ASVspoof dataset struggle to detect synthesized speech from codec models and current codec-based speech generation models, our CodecFake dataset significantly enhances anti-spoofing models’ capability to do so. We achieved an equal error rate of 0.4% for synthesized speech from VALL-E, a leading codec-based speech generation model. We will release all codes and data, aiming to contribute to mitigating malicious attacks from current advanced codec-based speech synthesis systems.

Acknowledgement: We thank the National Center for High-performance Computing (NCHC) of National Applied Research Laboratories (NARLabs) in Taiwan for providing computational and storage resources.

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] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi *et al.*, "Audiolm: a language modeling approach to audio generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.
- [3] P. K. Rubenstein, C. Asawaroengchai, D. D. Nguyen, A. Bapna, Z. Borsos, F. d. C. Quiry, P. Chen, D. E. Badawy, W. Han, E. Kharitonov *et al.*, "Audiopalm: A large language model that can speak and listen," *arXiv preprint arXiv:2306.12925*, 2023.
- [4] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzett, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi *et al.*, "Musiclm: Generating music from text," *arXiv preprint arXiv:2301.11325*, 2023.
- [5] 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.
- [6] Z. Zhang, L. Zhou, C. Wang, S. Chen, Y. Wu, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li *et al.*, "Speak foreign languages with your own voice: Cross-lingual neural codec language modeling," *arXiv preprint arXiv:2303.03926*, 2023.
- [7] T. Wang, L. Zhou, Z. Zhang, Y. Wu, S. Liu, Y. Gaur, Z. Chen, J. Li, and F. Wei, "Viola: Unified codec language models for speech recognition, synthesis, and translation," *arXiv preprint arXiv:2305.16107*, 2023.
- [8] 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.
- [9] Q. Chen, Y. Chu, Z. Gao, Z. Li, K. Hu, X. Zhou, J. Xu, Z. Ma, W. Wang, S. Zheng *et al.*, "Lauragpt: Listen, attend, understand, and regenerate audio with gpt," *arXiv preprint arXiv:2310.04673*, 2023.
- [10] X. Wang, M. Thakker, Z. Chen, N. Kanda, S. E. Eskimez, S. Chen, M. Tang, S. Liu, J. Li, and T. Yoshioka, "Speechx: Neural codec language model as a versatile speech transformer," *arXiv preprint arXiv:2308.06873*, 2023.
- [11] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," *arXiv preprint arXiv:2306.05284*, 2023.
- [12] G. L. Lan, V. Nagaraja, E. Chang, D. Kant, Z. Ni, Y. Shi, F. Iandola, and V. Chandra, "Stack-and-delay: a new codebook pattern for music generation," *arXiv preprint arXiv:2309.08804*, 2023.
- [13] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, "Audiogen: Textually guided audio generation," *arXiv preprint arXiv:2209.15352*, 2022.
- [14] H. Wu, H.-L. Chung, Y.-C. Lin, Y.-K. Wu, X. Chen, Y.-C. Pai, H.-H. Wang, K.-W. Chang, A. H. Liu, and H.-y. Lee, "Codec-superb: An in-depth analysis of sound codec models," *arXiv preprint arXiv:2402.13071*, 2024.
- [15] J. Yamagishi, X. Wang *et al.*, "Asvspoof 2021: accelerating progress in spoofed and deepfake speech detection," *arXiv preprint arXiv:2109.00537*, 2021.
- [16] M. Todisco, X. Wang *et al.*, "Asvspoof 2019: Future horizons in spoofed and fake audio detection," *arXiv preprint arXiv:1904.05441*, 2019.
- [17] T. Kinnunen, M. Sahidullah *et al.*, "The asvspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," 2017.
- [18] Z. Wu, T. Kinnunen *et al.*, "Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.
- [19] R. K. Das, X. Tian, T. Kinnunen, and H. Li, "The attacker's perspective on automatic speaker verification: An overview," *arXiv preprint arXiv:2004.08849*, 2020.
- [20] H. Wu, J. Kang, L. Meng, H. Meng, and H.-y. Lee, "The defender's perspective on automatic speaker verification: An overview," *arXiv preprint arXiv:2305.12804*, 2023.
- [21] J. Yi, R. Fu, J. Tao, S. Nie, H. Ma, C. Wang, T. Wang, Z. Tian, Y. Bai, S. Liang, S. Wang, S. Zhang, X. Yan, L. Xu, and H. Li, "Add 2022: the first audio deep synthesis detection challenge," in *IEEE ICASSP*, 2022.
- [22] Y. Xie, Y. Lu, R. Fu, Z. Wen, Z. Wang, J. Tao, X. Qi, X. Wang, Y. Liu, H. Cheng *et al.*, "The codefake dataset and countermeasures for the universally detection of deepfake audio," *arXiv preprint arXiv:2405.04880*, 2024.
- [23] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.
- [24] X. Zhang, D. Zhang, S. Li, Y. Zhou, and X. Qiu, "Spechtok-enizer: Unified speech tokenizer for speech large language models," *arXiv preprint arXiv:2308.16692*, 2023.
- [25] W.-N. Hsu *et al.*, "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.
- [26] 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.
- [27] Y.-C. Wu, I. D. Gebru, D. Marković, and A. Richard, "Audiodec: An open-source streaming high-fidelity neural audio codec," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [28] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 022–17 033, 2020.
- [29] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, "High-fidelity audio compression with improved rvqgan," *arXiv preprint arXiv:2306.06546*, 2023.
- [30] L. Ziyin, T. Hartwig, and M. Ueda, "Neural networks fail to learn periodic functions and how to fix it," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1583–1594, 2020.
- [31] Z. Du, S. Zhang, K. Hu, and S. Zheng, "Funcodec: A fundamental, reproducible and integrable open-source toolkit for neural speech codec," *arXiv preprint arXiv:2309.07405*, 2023.
- [32] J. Yamagishi, C. Veaux, K. MacDonald *et al.*, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92)," *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, 2019.
- [33] J.-w. Jung, H.-S. Heo, H. Tak, H.-j. Shim, J. S. Chung, B.-J. Lee, H.-J. Yu, and N. Evans, "Aasist: Audio anti-spoofing using integrated spectro-temporal graph attention networks," in *IEEE ICASSP*, 2022.