



PartialEdit: Identifying Partial Deepfakes in the Era of Neural Speech Editing

You Zhang^{*1}, Baotong Tian^{*1}, Lin Zhang², Zhiyao Duan¹

¹Audio Information Research Lab, University of Rochester, Rochester, USA

²Speech@FIT, Brno University of Technology, Brno, Czechia

{you.zhang, baotong.tian, zhiyao.duan}@rochester.edu, zlin@ieee.org

Abstract

Neural speech editing enables seamless partial edits to speech utterances, allowing modifications to selected content while preserving the rest of the audio unchanged. This useful technique, however, also poses new risks of deepfakes. To encourage research on detecting such partially edited deepfake speech, we introduce PartialEdit, a deepfake speech dataset curated using advanced neural editing techniques. We explore both detection and localization tasks on PartialEdit. Our experiments reveal that models trained on the existing Partial-Spoof dataset fail to detect partially edited speech generated by neural speech editing models. As recent speech editing models almost all involve neural audio codecs, we also provide insights into the artifacts the model learned on detecting these deepfakes. Further information about the PartialEdit dataset and audio samples can be found on the project page: <https://zyyouzhang.com/PartialEdit/index.html>.

Index Terms: speech deepfake detection, neural speech editing, partial deepfake audio, anti-spoofing, dataset

1. Introduction

Recent advances in text-to-speech (TTS) and voice-conversion (VC) technologies have enabled the generation of audio that is virtually indistinguishable from genuine human speech [1, 2]. The risk of their misuse by attackers to spread misinformation or attack security systems has grown significantly. Hence, deepfake detection has become an important area of research [3].

Most existing work on speech deepfake detection targets cases in which entire utterances are synthesized by TTS or VC systems. However, in recent years, speech editing algorithms have been emerging [4], where users can generate audio in high quality by *partially* modifying existing speech, rather than generating from scratch [5–10]. Although partial deepfakes have been discussed in some literature [11–13], they mainly focused on scenarios in which modified speech segments were generated using vocoder-based TTS and VC methods, then spliced back into the original utterance using basic signal processing techniques. However, the effectiveness of existing partial deepfake detection systems [12, 14] remains unverified against advanced speech editing techniques, where edited regions are generated through in-context learning, making them potentially more difficult to detect.

Additionally, recent speech generation models have transitioned to a neural codec-based paradigm, employing a flexible end-to-end generation pipeline, and are evolving rapidly. Unlike traditional vocoder-based approaches, modern models leverage neural audio codecs to represent speech as discrete to-

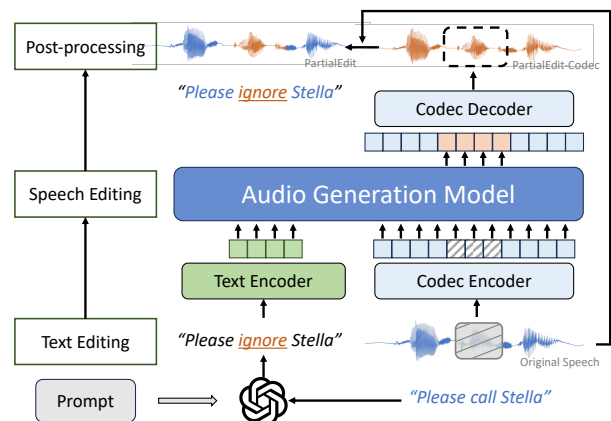


Figure 1: *PartialEdit* curation process, illustrated using the utterance p226_001 from the VCTK dataset. The original speech is modified to produce “Please ignore Stella,” where the speech segment corresponding to “ignore” is synthesized, making it the **partial deepfake** within the newly edited utterance.

kens, which audio language models can process. This enables seamless speech editing through techniques such as prompting and infilling, making speech editing more natural and contextually coherent while preserving natural prosody and speaker traits. Although prior studies [15–17] explored deepfake detection in neural codec-based speech generation, they overlooked speech editing models, which modify real recordings rather than generate fully synthetic speech. Since real-world misuse would often involve editing bona fide speech, our study addresses the unexplored challenge of the detection and localization of partially edited deepfakes in the era of neural speech editing.

We introduce **PartialEdit**, a new partial deepfake dataset that involves neural speech editing, with its curation process illustrated in Figure 1. PartialEdit is built on various modern speech editing models, including VoiceCraft [9], SSR-Speech [10], Audiobox-Speech, and Audiobox [18]. A previous study [19] is close to ours, but they only considered Voicebox (a vocoder-based method that preceded Audiobox) and did not handle more advanced neural codec-based speech editing methods. We conduct partial deepfake detection and localization on PartialEdit and find that detectors trained on existing partially spoofed audio fail to generalize to PartialEdit. We also perform deepfake localization comparisons across different speech editing methods and find that audio partially edited by VoiceCraft and SSR-Speech is harder to detect compared to Audiobox-Speech and Audiobox. Moreover, we discuss the impact of post-processing where stitching artifacts are introduced under codec-processed artifacts, and argue for a clearer defini-

* Equal Contribution

tion of deepfake regions that focuses solely on content-edited segments, regardless of codec-introduced artifacts. Our results provide new insights into the detection and localization of partially deepfake audio in the era of neural speech editing.

2. PartialEdit Dataset

In this section, we briefly overview the neural speech editing models used in curating our PartialEdit dataset, describe the dataset curation process, and present dataset statistics.

2.1. Neural speech editing models used in PartialEdit

We adapt the following codec-based speech generation models designed for or capable of speech editing: VoiceCraft [9], SSR-Speech [10], Audiobox-Speech and Audiobox [18]. All of them utilize Encodec [20] but with different configurations. To maintain consistency, we downsample all samples generated by Audiobox from 24 kHz to 16 kHz.

VoiceCraft (E1) [9] formulates sequence infilling (for speech editing) by rearranging tokens from the neural audio codec. The original speech is first converted to discrete codec tokens by the Encodec [20] encoder. A subset of tokens is masked and shifted to the end of the sequence. The target transcript, together with these processed tokens, is fed into a decoder-only Transformer, which autoregressively predicts the masked tokens. Surrounding frames are slightly modified to ensure smooth transitions, and the predicted tokens are rearranged and decoded back into audio by the Encodec decoder.

SSR-Speech (E2) [10] is built on VoiceCraft with key improvements. In particular, SSR-Speech can automatically detect the type of edit, whether insertion, deletion, or substitution, and apply the appropriate modifications accordingly, whereas VoiceCraft only allows one edit type provided by the user. Note that for both VoiceCraft and SSR-Speech, instead of using the original transcripts of VCTK, we follow the original structure of SSR-Speech to apply WhisperX¹ [21] to produce transcriptions as well as word-level alignment.

Audiobox-Speech (E3) [18] is an Encodec-based speech generation model based on flow-matching. It fine-tunes the self-supervised generative pre-training foundation model for in-context TTS using transcribed speech. For speech editing, the original transcript and the speech are aligned using a forced aligner of character-level char-units [22]. Given the target transcript, a new alignment is made with a preset masked duration of the edited region. The duration of the edited region is sampled using a pre-trained flow-matching duration model and then serves as conditional input for the audio flow.

Audiobox (E4) [18] is a unified model capable of generating both speech and general audio, conditioned on text descriptions or audio examples. The generation process remains the same as in Audiobox-Speech, and it can be considered a variant with different parameter weights due to comprehensive training objectives, but its additional capabilities for generating sound are not activated for this neural speech editing application.

2.2. PartialEdit curation process

We use the VCTK [23] dataset as the source of bona fide speech, consistent with several widely used deepfake datasets [15, 24], and the partial deepfake dataset PartialSpoof [12]. To generate high-quality, partially edited deepfake speech, we follow a three-step process as illustrated in Figure 1:

¹<https://github.com/m-bain/whisperX>










Name	Description	Method	Sample
VCTK	Original bonafide speech		
CodecFake	Codec-processed speech		
PartialEdit-Codec	Partially edited codec-processed speech	 + 	
PartialEdit	Partially edited speech		

Figure 2: Overview of different cases considered in this study. Our analysis decouples codec processing from generation.

Step 1: Text editing. To ensure naturalness after text modifications, we adopt an approach inspired by LlamaPartialSpoof [25]. Specifically, we iteratively input each transcript from VCTK [23] into GPT-4o-mini, prompting it to modify one word² while preserving grammatical correctness and fluency.

Step 2: Neural speech editing. As shown in Figure 1, our pipeline first encodes the modified transcripts from Step 1 into text tokens by a text encoder. Next, we align the original speech to its transcript and compare word-level differences to identify edit regions. The speech waveform is then converted to discrete tokens via a neural codec encoder. Tokens corresponding to designated edit regions are masked, while the audio generation model predicts the masked tokens conditioned on both text tokens and unmasked speech context. We preserve the unmasked tokens to maintain consistency in content-unedited segments. Finally, the modified token sequence—comprising both the newly predicted tokens and the retained original tokens—is decoded into an audio waveform using the neural codec decoder, producing the edited speech output.

Step 3: Post-processing. In speech editing models, although only the content-edited regions are intentionally generated, the entire output audio, including content-unedited regions, undergoes neural codec processing. To avoid additional artifacts introduced by neural audio codecs, we introduce a cut-and-paste post-processing step rather than directly using the output from speech editing models. Specifically, with alignment information derived from the speech editing model, we extract the edited parts from the generated speech and cut and paste them back into the original audio. This cut-and-paste operation, also adopted in PartialSpoof [12], ensures that the original bona fide speech is preserved in the resulting output. Detailed discussions on the artifacts introduced by neural codecs are in Section 5.1.

As shown in Figure 1, we construct **PartialEdit** using audio produced from the final post-processing step, where only the content-edited regions are generated by the speech editing model. We also retain an intermediate version of the dataset, **PartialEdit-Codec**, which contains deepfake speech from Step 2 without post-processing. The key distinction between PartialEdit and PartialEdit-Codec is whether the content-unedited parts have passed through the neural codec. In particular, although the content for the content-unedited region remains unchanged in both cases, the PartialEdit-Codec version introduces additional neural codec processing. We direct readers to Figure 2 for an intuitive comparison by visualization.

²Although the prompt instructs GPT to modify only a single word, maintaining sentence naturalness occasionally requires modifying two words, such as changing “included in” to “excluded from.”

Table 1: Duration (hours) and predicted mean opinion score (MOS) for PartialEdit and (PartialEdit-Codec). Duration report as train/dev/eval splits and shares between both versions.

Subset	Duration (h)	MOS
VCTK [23]	7.80 / 8.18 / 25.13	3.88±0.28
VoiceCraft (E1)	8.28 / 8.06 / 27.79	3.80±0.32 (3.60±0.38)
SSR-Speech (E2)	7.82 / 7.64 / 26.26	3.83±0.30 (3.71±0.34)
Audiobox-Speech (E3)	7.94 / 7.96 / 25.69	3.90±0.32 (3.53±0.32)
Audiobox (E4)	8.14 / 7.96 / 26.44	3.90±0.33 (3.54±0.32)

2.3. Dataset statistics information

We maintain the same utterance across all subsets generated by different speech editing models. After removing unsuccessfully edited utterances from all subsets, the dataset consists of 108 speakers and 43,358 partially edited utterances from each speech editing model. Following the speaker setup of ASVspoof2019 [24] and PartialSpoof [12], we split speakers and utterances into three disjoint partitions: 20 speakers (8,258 utterances), 20 speakers (7,915 utterances), and 68 speakers (27,185 utterances) for training, validation, and evaluation sets, respectively. We applied a pretrained DistillMOS [26] model to estimate the mean opinion score (MOS) as a measure of the naturalness of speech generated by each editing model.

Table 1 shows the duration and predicted MOS for both PartialEdit and PartialEdit-Codec subsets. Consistently, PartialEdit achieves a higher MOS than PartialEdit-Codec across all speech editing models, suggesting that the artifacts introduced by cutting and pasting the edited region back into the original speech are less noticeable than those introduced by neural codec processing. The overall MOS of PartialEdit is similar to that of bona fide speech from VCTK. This indicates that partially edited deepfake speech in our curated PartialEdit dataset achieves perceptual closeness to bona fide speech; This matches what we subjectively noticed while curating the dataset.

3. Detection on partially edited deepfakes

Partial deepfake detection involves two complementary tasks: utterance-level detection and segment-level localization. In this section, we focus on utterance-level detection—determining whether an entire speech sample is bona fide or contains partial edits. We will then discuss localization in Section 4.

3.1. Experimental setup

Datasets. Besides PartialEdit and PartialEdit-Codec datasets (where all speech generation models E1-E4 are included), we also incorporate PartialSpoof [12] dataset in this study. PartialSpoof [12] is a widely used dataset for partial deepfake detection, where random speech segments from real utterances are replaced with deepfake speech. Both PartialSpoof and our PartialEdit share the VCTK [23] corpus as the bona fide source.

Model configuration. We select XLSR-SLS [27] to perform deepfake detection, as it achieves top performance on various audio deepfake detection benchmarks [28, 29]. It adopts a large-scale self-supervised learning (SSL) representation XLSR³ [30] as the front-end and incorporates a sensitive layer selection (SLS) module as the back-end. We use the same set of hyperparameters for training following [27]. We train each model respectively for 10 epochs and set 3 for early stopping.

³https://dl.fbaipublicfiles.com/fairseq/wav2vec/xlsr2_300m.pt

Table 2: EERs (%) of deepfake detection on PartialEdit and existing datasets. Rows correspond to training data for the model, while columns correspond to test data. The same applies to the following tables.

Train \ Test	I	II	III
PartialSpoof (I)	2.55	12.95	23.72
PartialEdit-Codec (II)	14.54	0.13	27.59
PartialEdit (III)	23.06	0.41	2.14
I + III	3.00	0.64	2.61

Evaluation metrics. We use the equal error rate (EER) to present the performance of deepfake detection.

3.2. Utterance-level deepfake detection

Our results are presented in Table 2. The model trained on PartialSpoof (Row I) fails to generalize to our PartialEdit dataset, as indicated by the high EERs in Columns II and III. This suggests that our dataset presents new challenges for partial deepfake detection. Not surprisingly, training on PartialEdit-Codec (Row II) or on PartialEdit (Row III) shows prominently better performance on their same test data. Interestingly, training on PartialEdit generalizes well to PartialEdit-Codec (Row III, Column II) but not the other way around (Row II, Column III). This seems to suggest that artifacts presented in PartialEdit-Codec (mainly codec-related) are also presented in PartialEdit (codec-related and stitching-related), but not the other way around.

Additionally, the EERs on PartialEdit (Column III) are consistently higher than those on PartialEdit-Codec (Column II), indicating that partial deepfakes where unedited segments remain identical to the original are more challenging for detection systems. This suggests that the artifacts introduced by post-processing with cutting and pasting are less detectable than those introduced by codec processing. This finding also aligns with their predicted MOS values discussed in Section 2.3.

However, neither of the models trained on our PartialEdit datasets (Rows II, III) generalizes to PartialSpoof (Column I), indicating that PartialSpoof represents a different paradigm compared to PartialEdit. When mixing PartialSpoof and PartialEdit to train the model (I+III), we observe promising results on all datasets, underscoring the need for anti-spoofing systems to tackle both cutting-edge deepfakes and conventional ones.

4. Localization on partially edited deepfakes

This section focuses on the deepfake localization task, aiming to locate the edited regions within partially edited deepfakes.

4.1. Experimental setup

Datasets. We conduct localization experiments on both the entire PartialEdit dataset and individual subsets generated by different speech editing models (E1-E4 in Table 1) to examine how training on deepfakes generated by different speech editing methods affects the final localization performance.

Model configuration. We adopt BAM [14] given its state-of-the-art performance on deepfake localization on the PartialSpoof dataset. Following the configuration in [14], we utilize WavLM-Large⁴ [31] as the front-end and train the model at a 20 ms resolution. The training speech length is fixed at 4 seconds. We employ the Adam optimizer with an initial learning

⁴<https://github.com/microsoft/unilm/tree/master/wavlm>

Table 3: Frame-level EERs (%) of localization with cross-algorithm evaluation on different editing algorithms of PartialEdit.

Train \ Test	E1	E2	E3	E4	PartialEdit
E1	3.80	3.61	6.79	7.75	7.10
E2	6.50	3.57	9.17	9.33	9.51
E3	22.35	20.86	0.11	0.14	15.26
E4	16.32	15.32	0.17	0.11	11.77
PartialEdit	4.07	3.30	0.18	0.16	2.77

rate 10^{-5} that is then halved every 10 epochs. We also employ early stopping if the validation loss fails to reduce for 3 epochs. **Evaluation metrics.** We use frame-level EER with a 20 ms resolution to measure the performance of deepfake localization.

4.2. Localization across different speech editing algorithms

The results for localization are presented in Table 3. Similar to our findings in utterance-level spoof detection discussed in Section 3.2, models perform best when trained on data that match the test data. Their performance degrades when testing on data generated by unseen models. For example, VoiceCraft (E1) and SSR-Speech (E2) share similar technology, while both E3 and E4 are based on Audiobox. Models trained on data generated by E1 or E2 achieve lower EERs on those subsets but perform worse on audio generated by Audiobox (E3 and E4), and vice versa. In particular, models trained on E3 or E4 achieve very good EERs on E3 and E4, but they cannot generalize to E1 or E2. Furthermore, training on the entire PartialEdit dataset with E1-E4 pooled together (last row) achieves good performance across all test sets. While this result is not surprising, it reaffirms the conclusion we reached from Section 3: Diversity of training data matters.

5. Discussion

5.1. Impact of post-processing step of PartialEdit curation

As introduced in Section 2.2, PartialEdit applies an additional cut-and-paste post-processing step on top of PartialEdit-Codec to mitigate artifacts introduced by neural codecs on content-unedited regions. The key difference between PartialEdit-Codec and PartialEdit, therefore, is whether the content-unedited regions are processed by a neural codec or directly stitched from the original audio. Although results from Section 3.2 indicate the superior performance when combining both stitching and codec artifacts compared to only including codec artifacts (i.e. PartialEdit vs. PartialEdit-Codec), it remains unclear how those two operations affect the localization of edited regions in PartialEdit. This section conducts an experiment to examine deepfake localization on two settings in Table 4. We include CodecFake [15] as it is the codec-processed version of VCTK, and we select the SSR-Speech-edited (E2) subset among PartialEdit, as it is one of the most recent approaches and is hard to detect according to Section 4. To clarify, we define deepfake as content-edited regions, regardless of whether the segments have undergone codec processing, with the assumption that the detection target is malicious generation.

The results are presented in Table 4. As expected, the model can easily locate the content-edited region when the bona fide subset in the training data matches the content-unedited region in the test data. In such cases, the model achieves EERs lower than 6% on the diagonal. Notably, we observe a high EER close

Table 4: Comparison of localization EER (%) on PartialEdit-Codec with different settings. \triangle indicates datasets used as bona fide, while \circ represents datasets used as deepfake. (CFE: CodecFake [15]-Encodec; PEC: PartialEdit-Codec)

	Train on				Test on	
	VCTK	CFE	PEC	PartialEdit	I	II
I	\triangle			\circ	3.57	47.14
II		\triangle	\circ		10.73	5.30

to 50% when training on I and testing on II, indicating that if content-unedited codec-processed regions are not seen in partially edited audio during training, it becomes difficult to accurately locate the content-edited regions surrounded by content-unedited codec-processed segments. This suggests that artifacts introduced by codec processing may mislead the model if they are not recognized as bona fide during training. Crucially, when including Encodec-resynthesized data as a bona fide subset in row II, the result improves when testing on I, suggesting an approach to mitigate the misleading effects of these artifacts.

5.2. Limitations

We acknowledge a few limitations in this study. **1) On diverse speech editing models.** Although we discussed more speech editing models compared with [19], this is still limited. With the advancement of neural audio codecs and audio language models, further analysis of more sophisticated speech editing models is worth exploring. **2) On variations in text editing.** The localization task in our study focuses solely on identifying substitution regions. It does not address the localization of deletion operations, as detecting deletion requires further methodological design. However, SSR-Speech is capable of providing deletion or addition, though not in our PartialEdit-E2 (generated by SSR-Speech), which could be explored in future work.

6. Conclusion

In this study, we introduced PartialEdit, a partially edited deepfake dataset tailored for speech deepfake detection against neural speech editing. Unlike traditional deepfake datasets, PartialEdit consists of speech utterances where segments are modified by advanced speech editing algorithms and seamlessly stitched back into the original recording. Additionally, we include PartialEdit-Codec, where the unedited regions are also processed through a neural codec, reflecting the common operations of modern speech editing models.

Using PartialEdit, we investigated both deepfake detection and deepfake localization tasks. Our experiments reveal that models trained on PartialSpooF struggle to detect partially edited speech generated by neural speech editing models. Notably, among all models, VoiceCraft and SSR-Speech present greater challenges for detection.

Furthermore, we clearly define bona fide and deepfake segments in partial deepfake localization: Deepfake segments should only refer to those whose content has been modified, while bona fide segments refer to content-unedited regions, regardless of whether they undergo codec processing. This definition respects the original intention of neural codec models—achieving more effective compression. Our experiments also show that including codec-processed but content-unedited utterances as bona fide examples during training can improve performance in localizing content-edited regions in partial deepfakes.

7. Acknowledgments

This work is supported in part by National Institute of Justice Graduate Research Fellowship Award 15PNIJ-23-GG-01933-RESS, Intelligence Advanced Research Projects Activity ARTS Program, a New York State Center of Excellence in Data Science Award, and Meta Audiobox Responsible Generation Grant. The GPU resources are supported by NAIRR Pilot Project #240152, ACCESS #ELE240019, and NCSA Delta. Part of this material is based on work supported by Audiobox License, Copyright© Meta Platforms, Inc. All Rights Reserved.

The authors would also like to thank Puyuan Peng (UT Austin) and Helin Wang (JHU) for their brief discussions and for open-sourcing their work on VoiceCraft and SSR-Speech.

8. References

- [1] X. Tan, T. Qin, F. Soong, and T.-Y. Liu, “A survey on neural speech synthesis,” *arXiv preprint arXiv:2106.15561*, 2021.
- [2] 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,” in *Proc. International Conference on Machine Learning (ICML)*, 2024.
- [3] M. Li, Y. Ahmadiadi, and X.-P. Zhang, “A survey on speech deepfake detection,” *ACM Computing Surveys*, 2025.
- [4] T. Kässmann, Y. Liu, and D. Liu, “Speech editing—a summary,” *arXiv preprint arXiv:2407.17172*, 2024.
- [5] Z. Jin, G. J. Mysore, S. Diverdi, J. Lu, and A. Finkelstein, “Voco: Text-based insertion and replacement in audio narration,” *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 1–13, 2017.
- [6] M. Morrison, L. Rencker, Z. Jin, N. J. Bryan, J.-P. Caceres, and B. Pardo, “Context-aware prosody correction for text-based speech editing,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.
- [7] D. Tan, L. Deng, Y. T. Yeung, X. Jiang, X. Chen, and T. Lee, “EditSpeech: A text based speech editing system using partial inference and bidirectional fusion,” in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 626–633.
- [8] T. Wang, J. Yi, R. Fu, J. Tao, and Z. Wen, “CampNet: Context-aware mask prediction for end-to-end text-based speech editing,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 2241–2254, 2022.
- [9] P. Peng, P.-Y. Huang, S.-W. Li, A. Mohamed, and D. Harwath, “VoiceCraft: Zero-shot speech editing and text-to-speech in the wild,” in *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024, pp. 12 442–12 462.
- [10] H. Wang, M. Yu, J. Hai, C. Chen, Y. Hu, R. Chen, N. Dehak, and D. Yu, “SSR-Speech: Towards stable, safe and robust zero-shot text-based speech editing and synthesis,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.
- [11] L. Zhang, X. Wang, E. Cooper, J. Yamagishi, J. Patino, and N. Evans, “An initial investigation for detecting partially spoofed audio,” in *Proc. Interspeech*, 2021, pp. 4264–4268.
- [12] L. Zhang, X. Wang, E. Cooper, N. Evans, and J. Yamagishi, “The partialspoofer database and countermeasures for the detection of short fake speech segments embedded in an utterance,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 813–825, 2022.
- [13] J. Yi, J. Tao, R. Fu, X. Yan, C. Wang, T. Wang, C. Y. Zhang, X. Zhang, Y. Zhao, Y. Ren *et al.*, “ADD 2023: the second audio deepfake detection challenge,” in *Proc. IJCAI Workshop on Deepfake Audio Detection and Analysis (DADA)*, 2023.
- [14] J. Zhong, B. Li, and J. Yi, “Enhancing partially spoofed audio localization with boundary-aware attention mechanism,” in *Proc. Interspeech*, 2024, pp. 4838–4842.
- [15] H. Wu, Y. Tseng, and H. yi Lee, “CodecFake: Enhancing anti-spoofing models against deepfake audios from codec-based speech synthesis systems,” in *Proc. Interspeech*, 2024.
- [16] Y. Xie, Y. Lu, R. Fu, Z. Wen, Z. Wang, J. Tao, X. Qi, X. Wang, Y. Liu, H. Cheng, L. Ye, and Y. Sun, “The codecfake dataset and countermeasures for the universally detection of deepfake audio,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 33, pp. 386–400, 2025.
- [17] J. Du, X. Chen, H. Wu, L. Zhang, I. Lin, I. Chiu, W. Ren, Y. Tseng, Y. Tsao, J.-S. R. Jang *et al.*, “CodecFake-Omni: A large-scale codec-based deepfake speech dataset,” *arXiv preprint arXiv:2501.08238*, 2025.
- [18] A. Vyas, B. Shi, M. Le, A. Tjandra, Y.-C. Wu, B. Guo, J. Zhang, X. Zhang, R. Adkins, W. Ngan *et al.*, “Audiobox: Unified audio generation with natural language prompts,” *arXiv preprint arXiv:2312.15821*, 2023.
- [19] S.-F. Huang, H.-C. Kuo, Z. Chen, X. Yang, C.-H. H. Yang, Y. Tsao, Y.-C. F. Wang, H.-y. Lee, and S.-W. Fu, “Detecting the undetectable: Assessing the efficacy of current spoof detection methods against seamless speech edits,” in *Proc. IEEE Spoken Language Technology Workshop (SLT)*, 2024.
- [20] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” *Transactions on Machine Learning Research*, 2023.
- [21] M. Bain, J. Huh, T. Han, and A. Zisserman, “WhisperX: Time-accurate speech transcription of long-form audio,” in *Proc. Interspeech*, 2023, pp. 4489–4493.
- [22] L. Barrault, Y.-A. Chung, M. C. Meglioli, D. Dale, N. Dong, M. Duppenhaler, P.-A. Duquenne, B. Ellis, H. Elsahar, J. Haheim *et al.*, “Seamless: Multilingual expressive and streaming speech translation,” *arXiv preprint arXiv:2312.05187*, 2023.
- [23] C. Veaux, J. Yamagishi, and K. MacDonald, “CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit,” *The Centre for Speech Technology Research (CSTR)*, 2016.
- [24] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee *et al.*, “ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech,” *Computer Speech and Language*, vol. 64, p. 101114, 2020.
- [25] H.-T. Luong, H. Li, L. Zhang, K. A. Lee, and E. S. Chng, “Llama-PartialSpoofer: An LLM-driven fake speech dataset simulating disinformation generation,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.
- [26] B. Stahl and H. Gamper, “Distillation and pruning for scalable self-supervised representation-based speech quality assessment,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.
- [27] Q. Zhang, S. Wen, and T. Hu, “Audio deepfake detection with self-supervised xls-r and sls classifier,” in *Proc. ACM International Conference on Multimedia*, 2024, pp. 6765–6773.
- [28] Speech Arena, “Speech arena: Speech deepfake leaderboard,” <https://huggingface.co/spaces/Speech-Arena-2025/Speech-DF-Arena>, 2025, accessed: February 2025.
- [29] Y. Zhang, Y. Zang, J. Shi, R. Yamamoto, T. Toda, and Z. Duan, “SVDD 2024: The inaugural singing voice deepfake detection challenge,” in *Proc. IEEE Spoken Language Technology Workshop (SLT)*, 2024, pp. 782–787.
- [30] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino, A. Baevski, A. Conneau, and M. Auli, “XLS-R: Self-supervised cross-lingual speech representation learning at scale,” in *Proc. Interspeech*, 2022, pp. 2278–2282.
- [31] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, “WavLM: Large-scale self-supervised pre-training for full stack speech processing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.