DETECTING THE UNDETECTABLE: ASSESSING THE EFFICACY OF CURRENT SPOOF DETECTION METHODS AGAINST SEAMLESS SPEECH EDITS

Sung-Feng Huang[1,2], *Heng-Cheng Kuo*[2,3], *Zhehuai Chen*[1], *Xuesong Yang*[1], *Chao-Han Huck Yang*[1], *Yu Tsao*[3], *Yu-Chiang Frank Wang*[1], *Hung-yi Lee*[2], *Szu-Wei Fu*[1]

[1]NVIDIA [2]National Taiwan University [3]Acedemia Sinica

ABSTRACT

Neural speech editing advancements have raised concerns about their misuse in spoofing attacks. Traditional partially edited speech corpora primarily focus on cut-and-paste edits, which, while maintaining speaker consistency, often introduce detectable discontinuities. Recent methods, like A^{3}T and Voicebox, improve transitions by leveraging contextual information. To foster spoofing detection research, we introduce the Speech INfilling Edit (SINE) dataset, created with Voicebox. We detailed the process of re-implementing Voicebox training and dataset creation. Subjective evaluations confirm that speech edited using this novel technique is more challenging to detect than conventional cut-and-paste methods. Despite human difficulty, experimental results demonstrate that self-supervised-based detectors can achieve remarkable performance in detection, localization, and generalization across different edit methods. The dataset and related models will be made available at: `https://jasonswfu.github.io/SINE_dataset/index.html`

Index Terms— Neural Speech Editing, Audio Spoofing Detection, Seamless Speech Edit Corpus

1. INTRODUCTION

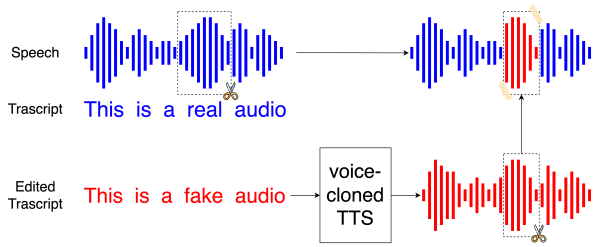
In the rapidly evolving realm of digital communication, the emergence of speech editing technologies offers considerable advantages, such as enhancing podcast quality and refining virtual assistant voices. However, alongside these benefits come notable security challenges, particularly the potential for the creation of convincing speech deepfakes [1, 2]. As accessibility to these techniques increases, so does the urgency for effective spoofing detection methods [3, 4, 5]. The creation of speech corpora tailored for identifying partial edits represents a significant step forward in the fight against deep fake manipulation [6, 7, 8]. However, existing corpora primarily focus on cut-and-paste (CaP) editing, a technique frequently associated with deepfakes, leading to detectable discontinuities at edit points. Consequently, these discontinuities have prompted many leading detection models to focus primarily on such irregularities [9, 10, 11, 12], potentially overlooking edits seamlessly woven into the audio. This limitation underscores the necessity for research into more sophisticated detection strategies to effectively safeguard against advanced spoofing attacks.

Traditional speech editing methods often rely on a CaP approach. This involves identifying keywords or phrases within the original speech that require modification and replacing them with speech segments conveying different or opposing meanings. However, these substitutions often fail to maintain continuity with the audio segments preceding and following the edit, leading to inevitable discontinuities at the edit boundaries. Recent advancements in speech editing technology have marked significant progress, with A^{3}T [13] and Voicebox [14] emerging as particularly powerful tools. Unlike traditional methods, A^{3}T and Voicebox exemplify speech infilling models, capable of generating speech conditioned on both the text and the surrounding audio. By masking the section of the speech audio that needs editing and replacing the corresponding keyword in the transcript, these models can seamlessly infill the masked speech, resulting in smooth speech edits. These novel speech edits not only closely match the timbre of the original recording but also avoid noticeable discontinuities at the edit boundaries, posing a new challenge for existing edit detectors.

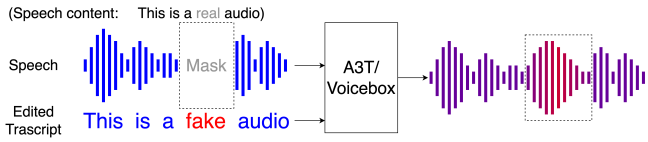
This paper aims to advance research in the detection of seamless speech edit to prevent the misuse of such technologies. In Section 2, we discuss two main methods of speech editing: cut-and-paste and Voicebox's seamless speech editing technique. Section 3 details how we re-implemented Voicebox and the generation of four types of audio (two types of edited speech and two types of genuine audio) to build our Speech INfilling Edit (SINE) dataset. Finally, in Section 4, we introduce four state-of-the-art (SOTA) detectors and present the experimental results, analysis, and findings of these top detectors on our new dataset.

In conclusion, this paper makes several significant contributions:

- To the best of our knowledge, we are the pioneers in proposing a corpus specifically designed for seamless speech editing detection.
- Through our experiments, we evaluated the performance of existing SOTA detectors on this novel dataset.



(a) Cut-and-paste speech editing.



(b) Seamless speech editing.

Fig. 1: Cut-and-paste and seamless speech editing.

- Despite the increased difficulty for human detection, a self-supervised learning (SSL) based detector consistently demonstrates robust detection capabilities across various speech editing scenarios.

- In the spirit of fostering further advancements in anti-spoofing research, we are committed to publicly releasing both the SINE dataset and the detector models for the benefit of the research community.

2. SPEECH EDITING METHODS

Advancements in speech processing and synthesis have enriched our lives but also expanded attack vectors, raising security threats. To address these, challenges like ASVspoof [15, 16, 17, 18] and ADD [19, 20] have been launched to promote defensive research. Initially focusing on synthetic speech detection amidst noise and degraded audio quality, the scope has expanded to include deepfake and speech editing technologies. These newer challenges emphasize the importance of detecting partially manipulated speech, where only segments are altered, presenting a subtler form of fakery that is more challenging to identify.

This paper focuses on such partial fake speech editing setups. In this section, we categorize speech editing methods into two types: CaP and seamless speech editing, detailed in sections 2.1 and 2.2, respectively. We also discuss how the differences between these two speech editing methods might affect the effectiveness of existing partial fake speech detection approaches in Section 2.3.

2.1. Cut-and-paste (CaP) speech editing

CaP stands out as one of the most straightforward methods for editing speech, ensuring minimal alterations to the original content while substantively replacing specific keywords or segments to alter the meaning of the sentence. To ensure the edited speech sounds natural, the replacement segments typically originate from another speech utterance spoken by the same speaker or are synthesized using voice-cloned text-to-speech (TTS) technology.

A speech spoofing corpus named HAD [7] employs this method to generate speech editing data. HAD is derived from the Chinese speech dataset AISHELL-3 [21], where each utterance's transcript is modified by substituting a word with its antonym or replacing a named entity with another, thus creating a new transcript with a modified meaning. It then utilizes GST-Tacotron [22] for zero-shot TTS synthesis based on the new transcript, followed by cutting and pasting the synthesized speech's corresponding segment into the original speech at the designated replacement point. This CaP technique has also been utilized in the partial fake detection tracks at ADD 2022 and 2023 to edit speech.

2.2. Seamless speech editing

Models trained for speech infilling have emerged in recent years, employing diverse masking strategies to enable functionalities like zero-shot TTS, elimination of background noise, and partial speech edits. During the training phase of speech infilling, these models partially mask the input speech and simultaneously take a text transcript as input, prompting the model to reconstruct the masked speech sections. This method allows models to learn to seamlessly fill in the gaps in speech, maintaining the integrity of the surrounding audio. Figure 1 illustrates the process of seamless editing using a speech-infilling model compared to traditional CaP methods. Prominent models in this domain include A^3T [13] and Voicebox [14], with their main difference being the use of regression loss by A^3T and conditional flow matching loss by Voicebox in their training processes.

2.3. Potential risk of seamless speech edit

This novel speech editing technique highlights the potential challenge in detecting partially fake speech, as speech edits often involve a small fraction of the total speech, possibly producing undetectable changes for current detectors. Existing challenges such as ADD 2022 and 2023 [19, 20] focus on CaP speech editing detection, leaving the effectiveness of current detection methods against novel seamless speech editing technologies in question.

To ascertain whether the novel seamless speech editing poses a threat to current partial fake speech detectors, we decided to construct a seamlessly edited speech corpus and test the performance of the existing top-performing detectors on

this new corpus. In Section 3, we will explain how we re-implemented and trained our own Voicebox model, describe how we used this Voicebox model to generate the speech edit corpus, and present subjective evaluation results of our edited speech.

3. SPEECH INFILLING EDIT (SINE) DATASET

This section outlines the configuration and generation pipeline of our Speech INfilling Edit (SINE) dataset. As Voicebox has not made their code and model parameters public, we begin by detailing our re-implementation and training approach. We then utilize Voicebox[1] to generate two types of edited speech as depicted in Section 2. Following this, we introduce our transcript editing process. We conclude with an overview of our dataset creation pipeline and a human evaluation of speech edit quality.

3.1. Re-implementation of Voicebox

We re-implemented Voicebox and conducted training on the LibriLight [23] medium set. As LibriLight does not provide transcripts, we resorted to a third-party version, LibriHeavy [24], for the requisite transcripts. Following the procedures outlined in the Voicebox paper, we applied the Montreal Forced Aligner [25] to achieve alignment between each utterance and its corresponding transcript. This process enabled us to gather critical training data, including phonemized transcripts and corresponding phone durations.

The Voicebox model utilizes mel-spectrogram for acoustic feature representation. During training, a masked mel-spectrogram serves as input, and the model is trained to reconstruct the entire utterances, encompassing both masked and unmasked segments. To transform the mel-spectrogram back into a waveform, the Vocos vocoder [26], which matches the performance of leading vocoders, is employed.

To enhance the editing process, we modified certain protocols from the original paper: our approach utilizes the utterance-level reconstruction loss rather than masked reconstruction loss, ensuring both seamless generation of the edited segment and high-quality reconstruction of the unedited parts.

3.2. Audio type settings

To evaluate the performance of various detectors against Voicebox's speech editing features, we employed four audio generation methods to compile our dataset: Real, Resynthesized-real (Resyn), CaP, and Infilling (Infill). Real and Resyn are categorized as genuine audio, while CaP and Infill are classified as partially fake.

Real audio comprises unaltered LibriLight speech, trimmed to match the LibriHeavy transcript's locations and durations. Resynthesized-real audio undergoes the process of transforming waveform into a mel-spectrogram and then back to waveform through a Vocos vocoder, but it is still classified as genuine in our experiments to test the detectors' capacity to differentiate based on speech edit patterns rather than vocoding patterns.

CaP and Infilling represent partially fabricated audio. CaP employs Voicebox for zero-shot TTS, generating speech from edited transcripts where the interested parts of the synthesized speech are cut and pasted to replace the original speech segment. Infilling, on the other hand, masks the input audio segment to be edited and uses Voicebox to generate an infilled mel-spectrogram according to the edited transcript. The primary difference between these methods lies in how Voicebox generates the editing part and whether the unedited audio segments are original or reconstructed by Voicebox.

In our detection experiments, we pair a genuine audio type with a partially fake one to train the detectors and analyze the distinguishable patterns learned by the detectors.

3.3. Transcript editing

We employed a method akin to the setup used in HAD: selecting a word or named entity from each sentence and substituting with its antonym or alternative named entity, thereby altering the semantic content of the entire sentence. Unlike HAD, we did not have a predefined pool of named entities or word/antonym pairs at our disposal. Instead, we aimed for more diverse editing outcomes. To achieve this, we generated edited transcripts by prompting the zephyr-7B-beta [27] large language model (LLM). However, the generated transcripts may not always meet the expected format. In such cases, we identified substitutions using word-level Levenshtein distance [28], randomly selecting a word for replacement as our method of edit manipulation.

3.4. Dataset generation pipeline

To ensure the quality of speech edits and accumulate a sufficient number of audio samples for our SINE dataset, we utilized the LibriLight medium set as the real audio source. For audio files in LibriLight, we matched each transcript to its segment in the LibriHeavy recordings, retaining only those between 6 to 8 seconds to fit our editing needs, resulting in approximately 95k clips. These clips were also resynthesized to construct the Resyn set. To generate edited speech, we first prompted the zephyr-7B-beta to edit transcripts and identify replacement keywords. Subsequently, we employed our trained Voicebox for two different speech editing techniques: CaP and Infilling.

In the splitting process into training, validation, and test sets, we ranked each speaker by the total number of their audio files in the dataset. Speakers with a higher count of audio samples were allocated to the training set, followed

[1] Voicebox can also be used as zero-shot TTS for cut-and-paste edits, as shown in Voicebox demo page: https://voicebox.metademolab.com/.

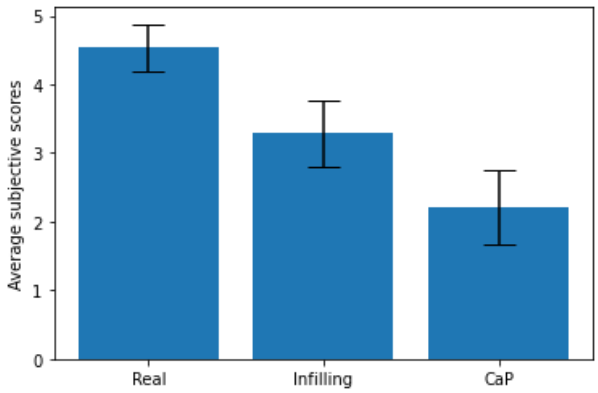


Fig. 2: Subjectivev Scores of different speech edit methods.

by sequential allocation to the validation and test sets. We maintained a proportional distribution of audio files across the train, validation, and test sets, approximately following a 6:2:2 ratio, ensuring the quantity of audio files in our validation/test sets was roughly equivalent to that of HAD. As our allocation was based on the number of audio samples per speaker, we anticipated that the test set would contain a significant number of unseen speakers, effectively testing the detectors' ability to generalize on a speaker-wise basis.

3.5. Subjective evaluation of edited speech

To compare the performance of different speech editing methods, we conducted a subjective evaluation using a 5-point scale test. Participants assessed audio files, which could be either real or edited, with a score of 5 indicating highly natural and unaltered audio, and a score of 1 signifying a strong conviction of partial editing (please refer to the supplementary materials for the experimental details). Initially, 20 samples were randomly selected from the Real condition, and then the corresponding speech from the Infilling and CaP set was chosen, resulting in a total of $20 \times 3 = 60$ utterances for each listener to evaluate. The order of playing the speech was randomized, and 17 listeners participated in the study. The experimental results are depicted in Fig 2, indicating that the Infilling edit method is more challenging for humans to detect than the CaP. Figure 3 shows the instructions of subjective evaluation.

3.6. SINE dataset statistics and demo files

Statistics of SINE dataset are shown in Table 1. Real/Resyn share the same statistics, and Infill/CaP also share the same statistics.

We randomly sampled several files for demo purpose, which are under supplementary material's `SINE_demo/` folder. Based on the editing quality, we split these demo audios into two folders: `SINE_demo/Good/` contains successfully edited audios, and `SINE_demo/Poor/` contains poorly edited audios.

Table 1: Statistics of SINE dataset.

Audio Types	subsets	# of Samples	# of Speakers	Durations (h)	Audio Lengths (s)	
					min	max
Real/ Resyn	train	26,547	70	51.82	6.00	8.00
	val	8,676	100	16.98	6.00	8.00
	test	8,494	900	16.60	6.00	8.00
Infill/ CaP	train	26,546	70	51.98	5.40	9.08
	val	8,686	100	16.99	5.45	8.76
	test	8,493	903	16.64	5.49	8.85

4. EXPERIMENTS

4.1. Partial-fake speech detectors

We introduce four of the most advanced partial-fake speech detectors currently available, which include the top two performers (SSL-Linear [9] and Res2d-Trans [10]) in the ADD 2022 challenge, as well as the second and third place (Res1d-BLSTM [29] and VGG-BGRU [30]) entrants in the ADD 2023 challenge. To avoid data leakage using LibriLight data, we use wav2vec2-base-960 [31] as our pre-trained SSL-Linear front-end, which is only trained on LibriSpeech [32]. Furthermore, since ADD2022/2023 datasets are not publicly available, we also evaluate our re-implementing results on HAD dataset [7] to validate their capability on existing spoofing corpus. A brief summary of the re-implemented detectors and the evaluation results can be found in Table 2.

4.2. Experimental setup

In the detection experiments, we randomly pair a type of real audio with a type of partial fake audio to analyze the origin of the distinguishable patterns learned by the detectors. Our experiments focused on the following real-fake audio type pairs: Real-CaP, Real-Infill, and Resyn-Infill[1]. Among these, Resyn-Infill ensures that all audio samples undergo a vocoding process, thereby minimizing the variability introduced by the vocoder.

As introduced in Section 4.1, we implemented four types of SOTA detectors to evaluate their performance on our proposed speech edit dataset. All of the detectors were trained with at least 100 epochs and terminated when saturated. Consistent with ADD 2023, we employed two metrics for assessment: frame-level F1 score and utterance-level accuracy.

Parts of the speech you will listen to may have been **edited**. Your task is to assess how confident you are that the speech has been **partially edited**, focusing solely on the speaker's coherent, and **naturalness** of the speech. Please do not consider noise in the speech, grammar, semantics, or other linguistic factors in your evaluation. Use the following scale to rate your confidence:

- **1**. I am completely convinced, based on clear evidence, that the speech has been partially edited.
- **2**. I suspect that the speech has been partially edited, noting specific inconsistencies or unnatural elements.
- **3**. I neither lean towards believing the speech has been partially edited nor fully convinced of its realness, due to a lack of compelling evidence in either direction.
- **4**. I suspect that the speech is real, but have some minor doubts about its realness.
- **5**. I am completely convinced that the speech is real and unedited.

Each audio should only be replayed at most **twice**.

Fig. 3: Instruction of the subjective evaluation.

Table 2: Summary of SOTA detectors. ASP stands for "attentive statistic pooling". Results on the HAD test set are also shown. (evaluated with frame-level $F1$-score / utterance-level accuracy)

Model	Front-End	Backbone	Frame Classifier	Utter Classifier	# parameters	HAD results
SSL-Linear [9]	SSL	linear	linear	ASP+linear	95M	99.90 / 99.94
Res2d-Trans [10]	mel-spectrogram	SE-ResNet(Conv2D) + Trans. block	linear	ASP+linear	34M	94.14 / 96.79
Res1d-BLSTM [29]	mel-spectrogram	ResNet(Conv1D) + BiLSTM	linear	# fake frames < 3	2.3M	95.39 / 96.21
VGG-BGRU [30]	mel-spectrogram	VGG(Conv2D) + 2*BiGRU	linear	linear softmax	1.0M	98.88 / 99.19

4.3. Experimental results

The primary experimental results are presented in Table 3. As illustrated in the table, SSL-Linear achieved high scores across various evaluation metrics and settings, indicating its effective capability to detect a range of speech editing configurations. In utterance-level prediction, other detectors also demonstrated good performance, with a notable generalization gap observed in the Real-CaP setting, while showing commendable results in both Real-Infill and Resyn-Infill scenarios. On the frame-level prediction aspect, SSL-Linear exhibited exceptional performance, whereas Res2d-Trans failed to learn, even on the training dataset. The training performance of the remaining detectors was superior in the Real-CaP setting but struggled with learning in Real-Infill and Resyn-Infill; their performance during testing also revealed a larger generalization gap.

We argue that the superior performance of SSL-Linear over other detectors can be attributed to its self-supervised pre-training on a large-scale real dataset. This exposure likely facilitated the learning of more generalized features. Conversely, other detection models lack pre-training and exhibit tendencies to overfit at the frame-level prediction stage. This overfitting suggests that these models may be focusing on irrelevant patterns or noise within the training set, rather than assimilating the true spoofing pattern essential for achieving robust performance on unseen data.

Given that only SSL-Linear was successfully trained and could generalize to testing, it becomes the preferred detector for analyzing the properties of our dataset. Therefore, in the following sections, its generalization ability to different distortions and across different editing methods are investigated.

4.4. Evaluation of robustness to distortions

To test the robustness of SSL-Linear detector to different audio distortions, we choose 2 common types: adding background noise and audio compression (i.e., MP3, bit-rate = 32kbps). The background noise comes from the MUSAN dataset [33], and we mix the noise with the clean speech in our test set in two different conditions: high SNR (0 to 10 dB) and low SNR (-10 to 0 dB).

The detection results are shown in Table 4. From the table we can observe that SSL-linear is quite robust in detecting the edited speech at the utterance level. However, as expected, adding severe background noise (low SNR condition) will significantly degrade the model's ability to localize the edited frames. Please note that this is a **zero-shot** evaluation, no audio distortions are intentionally added during the training of the detector.

[1]Resyn-Cap is not included because it's not a realistic setting.

Table 3: Detector's train/test results on our proposed dataset. Res2d-Trans failed to learn frame-level detection on all cases. The redder background color refers to lower scores.

Model	Audio Types	Frame-F1↑ (%) Train	Frame-F1↑ (%) Test	Utt-acc↑ (%) Train	Utt-acc↑ (%) Test
SSL -Linear [9]	Real-CaP	99.11	97.38	99.99	99.55
	Real-Infill	98.61	92.87	100.0	99.59
	Resyn-Infill	97.56	92.68	99.98	99.32
Res2d -Trans [10]	Real-CaP	38.89	21.31	97.60	66.92
	Real-Infill	62.82	29.85	99.66	94.21
	Resyn-Infill	49.83	27.95	99.64	92.48
Res1d -BLSTM [29]	Real-CaP	92.35	40.00	99.24	66.22
	Real-Infill	69.88	33.95	96.62	86.39
	Resyn-Infill	78.41	33.70	95.46	82.51
VGG -BGRU [30]	Real-CaP	90.38	68.03	99.86	83.66
	Real-Infill	66.04	46.52	99.47	94.40
	Resyn-Infill	83.17	55.28	99.91	92.66

Table 4: Robustness evaluation of SSL-Linear detector. Rows refer to the audio types for both training and testing, while columns refer to the testing distortions. The redder background color refers to lower scores.

Audio Types	Distortions Low SNR	Distortions High SNR	Distortions MP3	Clean
frame-level F1↑ (%)				
Real-CaP	34.11	58.50	94.86	97.38
Real-Infill	25.38	44.23	86.40	92.87
Resyn-Infill	23.33	43.89	85.74	92.68
utterance-level Accuracy↑ (%)				
Real-CaP	81.93	88.60	98.67	99.55
Real-Infill	86.48	90.10	99.50	99.59
Resyn-Infill	68.89	76.86	97.84	99.32

4.5. Cross-editing-method evaluation

In the main experiment, both our training and testing sets employed identical audio-type settings, and SSL-Linear demonstrated strong performance across various configurations. In this experiment, we will conduct cross-evaluation among different training and testing sets to analyze whether training on specific audio-type pairs can be generalized to other types.

Experimental results are presented in Table 5. Among all type configurations, the detector trained on **Resyn-Infill** setting achieves the best generalization ability across different audio pairs. This success can be attributed to the use of resynthesized-real audio, which avoids interference from vocoding patterns. This directs the focus of detectors toward identifying more universal spoofing patterns instead of being distracted by boundary discontinuities.

Table 5: Cross-editing-method evaluation with SSL-Linear. Rows refer to the train sets, and columns refer to the test sets. The redder background color refers to lower scores.

Train \Test	Real-CaP	Real-Infill	Resyn-Infill
frame-level F1 (%)			
Real-CaP	97.38	51.57	40.42
Real-Infill	62.18	92.87	52.93
Resyn-Infill	86.56	92.02	92.68
utterance-level Accuracy (%)			
Real-CaP	99.55	86.83	69.58
Real-Infill	67.29	99.59	52.81
Resyn-Infill	91.66	98.78	99.32

4.6. Generalizability to detect edits from unseen editing models

We also evaluated whether our trained detector can generalize to detect speech edited by other speech editing/synthesis models. HAD test set [7], and a new test set based on VoiceCraft [34] are considered in this experiment. The audio characteristics of these unseen test sets are **very different** from SINE training set, including languages (HAD is in Mandarin while SINE is in English), sampling rates, vocoders, speech editing models and methods (VoiceCraft reformulates the infilling method into a language-model-based next-token-prediction problem), etc. Our experimental results reveal that the current detector trained on the SINE dataset is difficult to generalize to detect these two unseen models' edits.

Based on the experimental results from both Section 4.5 and 4.6, we can conclude that the current detector can learn more generalized patterns across different editing methods by training with the Resyn-Infill setting of the SINE dataset, however, it still struggles to learn patterns generated by unseen editing models.

5. CONCLUSION

In this paper, we introduce the first large-scale seamless spoofing dataset, SINE, to advance anti-spoofing research. In our experiments with current SOTA detectors, we observed that only SSL-Linear was capable of successfully identifying spoofs in this new dataset, while the remaining detectors struggled to generalize well on the test set. Furthermore, a cross-editing-methods experiment suggests that detectors trained on Resyn-Infill can generalize to different edit methods. This may imply that the detection is based on some common spoofing patterns. Although the current detector is difficult to generalize to unseen edit models (e.g., VoiceCraft), we believe this work will foster research toward a universally applicable detector to prevent the malicious usage of different speech edit/synthesis technologies.

6. REFERENCES

[1] NBC News, ""why AI-generated audio is so hard to detect"," Feb 2024.

[2] Rohan Kumar Das, Xiaohai Tian, Tomi Kinnunen, and Haizhou Li, "The Attacker's Perspective on Automatic Speaker Verification: An Overview," in *Proc. Interspeech 2020*, 2020, pp. 4213–4217.

[3] Lin Zhang, Xin Wang, Erica Cooper, Junichi Yamagishi, Jose Patino, and Nicholas Evans, "An initial investigation for detecting partially spoofed audio," in *Interspeech 2021*. ISCA, 2021, pp. 4264–4268.

[4] Jiangyan Yi, Chenglong Wang, Jianhua Tao, Xiaohui Zhang, Chu Yuan Zhang, and Yan Zhao, "Audio deepfake detection: A survey," *arXiv preprint arXiv:2308.14970*, 2023.

[5] Haibin Wu, Jiawen Kang, Lingwei Meng, Helen Meng, and Hung-yi Lee, "The defender's perspective on automatic speaker verification: An overview," 2023.

[6] Lin Zhang, Xin Wang, Erica Cooper, Nicholas Evans, and Junichi Yamagishi, "The partialspoof 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.

[7] Jiangyan Yi, Ye Bai, Jianhua Tao, Haoxin Ma, Zhengkun Tian, Chenglong Wang, Tao Wang, and Ruibo Fu, "Half-truth: A partially fake audio detection dataset," in *Proc. Interspeech*, 2021, pp. 1654–1658.

[8] Haoxin Ma, Jiangyan Yi, Chenglong Wang, Xinrui Yan, Jianhua Tao, Tao Wang, Shiming Wang, and Ruibo Fu, "Cfad: A chinese dataset for fake audio detection," *arXiv preprint arXiv:2207.12308*, 2022.

[9] Zhiqiang Lv, Shanshan Zhang, Kai Tang, and Pengfei Hu, "Fake audio detection based on unsupervised pretraining models," in *Proc. ICASSP*, 2022, pp. 9231–9235.

[10] Haibin Wu, Heng-Cheng Kuo, Naijun Zheng, Kuo-Hsuan Hung, Hung-Yi Lee, Yu Tsao, Hsin-Min Wang, and Helen Meng, "Partially fake audio detection by self-attention-based fake span discovery," in *Proc. ICASSP*, 2022, pp. 9236–9240.

[11] Zexin Cai, Weiqing Wang, and Ming Li, "Waveform boundary detection for partially spoofed audio," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[12] Zexin Cai and Ming Li, "Integrating frame-level boundary detection and deepfake detection for locating manipulated regions in partially spoofed audio forgery attacks," *Computer Speech & Language*, vol. 85, pp. 101597, 2024.

[13] He Bai, Renjie Zheng, Junkun Chen, Mingbo Ma, Xintong Li, and Liang Huang, "A^3T: Alignment-aware acoustic and text pretraining for speech synthesis and editing," in *Proc. ICML*, 2022, pp. 1399–1411.

[14] Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al., "Voicebox: Text-guided multilingual universal speech generation at scale," in *Proc. NeurIPS*, 2023.

[15] Zhizheng Wu, Junichi Yamagishi, Tomi Kinnunen, Cemal Hanilçi, Mohammed Sahidullah, Aleksandr Sizov, Nicholas Evans, Massimiliano Todisco, and Hector 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.

[16] Tomi Kinnunen, Md Sahidullah, Héctor Delgado, Massimiliano Todisco, Nicholas Evans, Junichi Yamagishi, and Kong Aik Lee, "The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," in *Proc. Interspeech*, 2017, pp. 2–6.

[17] Massimiliano Todisco, Xin Wang, Ville Vestman, Md Sahidullah, Héctor Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi Kinnunen, and Kong Aik Lee, "ASVspoof 2019: Future horizons in spoofed and fake audio detection," in *Proc. Interspeech*, 2019, pp. 1008–1012.

[18] Junichi Yamagishi, Xin Wang, Massimiliano Todisco, Md Sahidullah, Jose Patino, Andreas Nautsch, Xuechen Liu, Kong Aik Lee, Tomi Kinnunen, Nicholas Evans, et al., "Asvspoof 2021: accelerating progress in spoofed and deepfake speech detection," in *ASVspoof 2021 Workshop-Automatic Speaker Verification and Spoofing Coutermeasures Challenge*, 2021.

[19] Jiangyan Yi, Ruibo Fu, Jianhua Tao, Shuai Nie, Haoxin Ma, Chenglong Wang, Tao Wang, Zhengkun Tian, Ye Bai, Cunhan Fan, Shan Liang, Shiming Wang, Shuai Zhang, Xinrui Yan, Le Xu, Zhengqi Wen, and Haizhou Li, "ADD 2022: the first audio deep synthesis detection challenge," in *Proc. ICASSP*, 2022, pp. 9216–9220.

[20] Jiangyan Yi, Jianhua Tao, Ruibo Fu, Xinrui Yan, Chenglong Wang, Tao Wang, Chu Yuan Zhang, Xiaohui Zhang, Yan Zhao, Yong Ren, et al., "ADD 2023: the second audio deepfake detection challenge," in *IJCAI*

2023 Workshop on Deepfake Audio Detection and Analysis (DADA 2023), 2023.

[21] Yao Shi, Hui Bu, Xin Xu, Shaoji Zhang, and Ming Li, "AISHELL-3: A Multi-Speaker Mandarin TTS Corpus," in *Proc. Interspeech 2021*, 2021, pp. 2756–2760.

[22] Daisy Stanton, Yuxuan Wang, and RJ Skerry-Ryan, "Predicting expressive speaking style from text in end-to-end speech synthesis," in *Proc. IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 595–602.

[23] Jacob Kahn, Morgane Riviere, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al., "Libri-light: A benchmark for asr with limited or no supervision," in *Proc. ICASSP*, 2020, pp. 7669–7673.

[24] Wei Kang, Xiaoyu Yang, Zengwei Yao, Fangjun Kuang, Yifan Yang, Liyong Guo, Long Lin, and Daniel Povey, "Libriheavy: a 50,000 hours asr corpus with punctuation casing and context," in *Proc. ICASSP*, 2024.

[25] Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger, "Montreal forced aligner: trainable text-speech alignment using kaldi," in *Proc. Interspeech*, 2017, pp. 498–502.

[26] Hubert Siuzdak, "Vocos: Closing the gap between time-domain and Fourier-based neural vocoders for high-quality audio synthesis," in *Proc. of ICLR*, 2024.

[27] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al., "Zephyr: Direct distillation of LM alignment," Tech. Rep., The H4 Team at Hugging Face, October 2023.

[28] Vladimir I Levenshtein et al., "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*. Soviet Union, 1966, vol. 10, pp. 707–710.

[29] Jie Liu, Zhiba Su, Hui Huang, Caiyan Wan, Quanxiu Wang, Jiangli Hong, Benlai Tang, and Fengjie Zhu, "TranssionADD: A multi-frame reinforcement based sequence tagging model for audio deepfake detection," in *Proceedings of IJCAI 2023 Workshop on Deepfake Audio Detection and Analysis*, 2023, vol. 750, pp. 113–118.

[30] Kang Li, Xiao-Min Zeng, Jian-Tao Zhang, and Yan Song, "Convolutional recurrent neural network and multitask learning for manipulation region location," in *Proceedings of IJCAI 2023 Workshop on Deepfake Audio Detection and Analysis*, 2023, vol. 750, pp. 18–22.

[31] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12449–12460, 2020.

[32] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev 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.

[33] David Snyder, Guoguo Chen, and Daniel Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[34] Puyuan Peng, Po-Yao Huang, Daniel Li, Abdelrahman Mohamed, and David Harwath, "VoiceCraft: Zero-shot speech editing and text-to-speech in the wild," *arXiv preprint arXiv:2403.16973*, 2024.