國立臺北科技大學
National Taipei University of Technology

資訊工程系碩士班

碩士學位論文

Department of Computer Science and Information Engineering

Master Thesis

用於語音防偽之中文局部深偽語音資料集建構與評估

A Mandarin Partial Deepfake Speech Dataset for Audio Anti-Spoofing

研究生：張維恩

Researcher: WEI-EN ZHANG

指導教授：陳香君博士

Advisor: SHIANG-JIUN CHEN, Ph.D.

August, 2026

國立臺北科技大學
研究所碩士學位論文口試委員會審定書

本校 資訊工程 研究所 張維恩 君

所提論文，經本委員會審定通過，合於碩士資格，特此證明。

學位考試委員會

委　　員：吳承恩

馬奕薇

陳香君

指導教授：陳香君

所　　長：劉建宏

中 華 民 國 115 年 7 月 28 日

摘要

關鍵詞：局部深度偽造語音偵測、中文語音資料集、基於神經編解碼的語音編輯

隨著由深度學習驅動的語音生成技術快速發展，文字轉語音 (Text-to-Speech, TTS) 與語音轉換 (Voice Conversion, VC) 技術現在已能產生高度自然的合成語音。然而，這些技術也帶來了與深度偽造 (deepfake) 語音相關的安全風險。相較於完全由模型生成、整體皆為合成的偽造語音，部分深度偽造 (partial deepfake) 語音——即僅修改特定片段，同時保留大部分真實錄音特徵——更具隱蔽性，因此對現有的偵測模型來說更難以辨識。

目前關於部分深度偽造語音的研究主要集中在英語語音上，針對中文語音環境的研究則相對有限。因此，本研究藉由建構一個中文部分深度偽造語音資料集，來探討中文部分深度偽造語音的偵測問題。本研究之資料生成流程遵循 PartialEdit 的方法，採用語義上合理的文本編輯策略，並結合基於神經編解碼器 (neural codec) 的語音編輯模型，以生成自然且在聲學上連續的部分偽造語音。此過程旨在模擬現實應用中可能出現的真實中文語音竄改情境。

與現有的中文部分深度偽造資料集相比，本研究提出的資料集有效地減少了拼接邊界與人工編輯的痕跡，使得偽造語音在聲學上更接近真實錄音。基於此資料集，本研究進一步訓練並評估了部分深度偽造偵測模型，分析它們在中文語音場景下的偵測能力與泛化表現。此外，本研究也比較了部分偽造與完全偽造語音對偵測模型的影響。

實驗結果顯示，對現有模型而言，部分深度偽造語音比完全偽造語音更難被偵測，且中文語音的特徵確實會影響語音偽造的生成與偵測表現。整體而言，本研究提出的中文部分深度偽造資料集與相關分析，可作為未來中文語音深度偽造偵測研究的基礎，同時也有助於在語音安全與多媒體鑑識領域中，增進對部分語音偽造攻擊的理解與防禦能力。

TAIPEI TECH

Abstract

Keywords: Partial Deepfake Speech Detection, Mandarin Speech Dataset, Neural Codec-based Speech Editing

With the rapid advancement of deep learning-driven speech generation technologies, Text-to-Speech (TTS) and Voice Conversion (VC) technologies are now capable of producing highly natural deepfake speech. However, these technologies have also introduced security risks associated with deepfake speech. Compared to fully deepfake speech generated entirely by models, partial deepfake speech where only specific segments are modified while most authentic recording characteristics are preserved is more difficult to detect and therefore more challenging for existing detection models to identify.

Current research on partial deepfake speech primarily focuses on English, while corresponding studies on Mandarin speech remain relatively limited. Therefore, this study addresses the problem of Mandarin partial deepfake speech detection by constructing a Mandarin partial deepfake speech dataset. The data generation pipeline in this study follows the methodology of PartialEdit, employing semantically plausible text editing strategies combined with a neural codec-based speech editing model to generate natural and acoustically continuous partially fake speech. This process aims to simulate realistic Mandarin speech tampering scenarios that may occur in real-world applications.

Compared with existing Mandarin partial deepfake datasets, the dataset proposed in this study effectively reduces splicing boundaries and manual editing artifacts, making the deepfake speech acoustically closer to genuine recordings. Based on this dataset, this study further trains and evaluates partial deepfake detection models, analyzing their detection capability and generalization performance in Mandarin speech scenarios. In addition, the study compares the impact of partial fake and fully fake speech on detection models.

Experimental results show that partial deepfake speech is more difficult for existing models to detect than fully deepfake speech. The results also suggest that language-related char-

TAIPEI TECH

acteristics may influence deepfake speech generation and detection performance. Overall, the Mandarin partial deepfake dataset and related analyses presented in this study can serve as a foundation for future research on Mandarin speech deepfake detection, while also contributing to improved understanding and defense against partial speech deepfake attacks in the fields of speech security and multimedia forensics.



誌謝

論文順利完成，我很感激這一路上陪伴與支持我的每一個人。最深的謝意要獻給我的老師陳香君教授，謝謝教授在我實驗和撰寫論文時耐心地指導，並給予我明確的方向。當我面對困難與不確定時，教授的指引和溫暖的回饋對我來說至關重要，是我研究道路上的一盞明燈。

我也要感謝口試委員吳和庭教授與馬奕葳教授。口試時專業的提問與詳細的建議都是我寶貴的收穫，大大地拓展了我的學術視野，也使我的論文能更加完善。

我還要謝謝林鈺翔同學和王少雷同學，感謝兩位同學在我剛進實驗室時的關照，也謝謝一路上的陪伴。

最後我特別感謝我的家人，進入研究所後我經歷了不少挫折，但有家人的鼓勵才使我能夠安心地完成學業。

TAIPEI TECH

Table of Contents

摘要 i

Abstract ii

誌謝 iv

Table of Contents v

List of Figures vii

List of Tables viii

Chapter 1 Introduction 1

Chapter 2 Related Work 3

2.1 Speech Deepfake Detection 3

2.2 Partial Deepfake Speech Detection 4

2.3 Partial Deepfake Dataset Generation 5

2.4 Neural Audio Codec and Speech Editing 6

2.5 LLM-based Text Manipulation 8

2.6 Mandarin Speech Dataset 9

2.7 Summary 9

Chapter 3 Methodology 11

3.1 AISHELL-3 Dataset 12

3.2 Preprocessing 12

3.3 Text Editing 14

3.3.1 Text Deletion 15

3.3.2 Text Insertion 17

3.3.3 Text Substitution 18

3.4 Edited Text Validation 20

3.4.1 Deleted Text Validation 21

3.4.2 Inserted Text Validation 21

3.4.3 Substituted Text Validation 22

3.5 Speech Infilling 22

TAIPEI TECH

3.5.1 Speech Deletion . . . 23
3.5.2 Speech Insertion . . . 23
3.5.3 Speech Substitution . . . 23
3.6 Post-processing . . . 23
3.7 Dataset Curation . . . 24
Chapter 4 Dataset Utility Evaluation . . . 25
4.1 Evaluation Framework . . . 25
4.2 Baseline Model . . . 26
4.3 Dataset Preparation for Training . . . 27
4.4 Experimental Configuration . . . 28
4.5 Evaluation Metrics . . . 29
Chapter 5 Results and Analysis . . . 31
5.1 In-domain and Cross-dataset Evaluation . . . 31
5.1.1 In-domain Evaluation . . . 32
5.1.2 Cross-dataset Evaluation . . . 32
5.2 Edit-type Evaluation . . . 33
Chapter 6 Conclusion and Future Work . . . 36
6.1 Conclusion . . . 36
6.2 Future Work . . . 37
References . . . 38

TAIPEI TECH

List of Figures

3.1 Deepfake Speech Generation Pipeline 11

3.2 Expanded Preprocessing Subprocess 13

TAIPEI TECH

List of Tables

4.1 Summary of Dataset Splits for MPDS, HAD, and PartialSpoof 28

4.2 Summary of Experimental Setup and Hyperparameters 29

5.1 In-domain and Cross-dataset Evaluation Results 32

5.2 Evaluation Results across Different Edit Types on MPDS Subsets 35

TAIPEI TECH

Chapter 1 Introduction

Recent improvements in Text-to-Speech (TTS)[1] and Voice Conversion (VC)[2] models produce deepfake speech that closely resembles human fluency, timbre, and prosody. While beneficial for applications like automated reading and multimedia dubbing, these technologies lower the barrier to speech deepfake, increasing the risks of identity impersonation and fraud. Consequently, detecting deepfake speech has become a critical focus in multimedia forensics.

Most deepfake speech research focuses on fully fabricated utterances. However, attackers often manipulate only specific keywords or semantically critical segments to distort the original meaning. This partial deepfake speech [3] preserves the acoustic characteristics and background noise of the original recording, making it harder for both human listeners and detection models to identify. While some studies incorporate partial deepfakes into their evaluation protocols, systems designed primarily for fully deepfake speech often experience performance degradation when facing partial or mixed attacks.

To simulate realistic speech attack environments, datasets like PartialSpoof [4] simulate attacks by splicing synthesized segments into bona fide speech, showing the increased difficulty of detecting partial deepfakes. Furthermore, PartialEdit [5] demonstrated that context-aware neural speech editing generates more natural boundaries, further complicating detection. Yet, these datasets primarily focus on English. Their methods and data characteristics may not directly transfer to Mandarin.

Mandarin differs from English in syllable structure, tones, and grammar, which directly affect the acoustic characteristics of partial deepfakes. Relying on English-based findings may not accurately reflect model performance in Mandarin scenarios. Existing Mandarin partial deepfake datasets, such as Half-Truth: A partially fake audio detection dataset (HAD)[6], typically rely on single-word substitutions from predefined entity pools. This approach limits semantic variation, and conventional splicing methods often leave audible traces. The community still lacks a Mandarin dataset featuring diverse manipulation patterns based on modern speech editing techniques.

In this study, a dataset construction pipeline is proposed that combines a large language

model with VoiceCraft-X [7], a neural codec-based speech editing model. The dataset generated through this pipeline is referred to as Mandarin Partial Deepfake Speech (MPDS). Unlike single-substitution approaches, the proposed pipeline includes deletion, insertion, and substitution. Targeted lexical manipulations, such as modifying adjectives or removing negations, are applied to alter the semantic meaning while maintaining grammatical plausibility. The generated segments are then seamlessly blended with the original audio, minimizing conventional splicing artifacts to simulate realistic partial spoofing attacks.

The generated dataset is evaluated using AASIST [8] as the baseline model under three primary setups: in-domain evaluation, cross-dataset generalization, and performance analysis across specific edit types (deletion, insertion, and substitution). The detection performance and generalization of anti-spoofing models in Mandarin partial deepfake scenarios are analyzed, demonstrating the utility of MPDS. This dataset and the accompanying experimental analysis provide insight into the acoustic traits of Mandarin partial deepfakes and serve as a valuable baseline for future research in Mandarin speech forensics.

The remainder of this thesis is organized as follows. Chapter 2 provides a review of speech deepfake detection techniques, alongside existing partial deepfake datasets and their construction methodologies. Chapter 3 details the generation pipeline for MPDS. Chapter 4 outlines the experimental setup, baseline model, and data preprocessing. Chapter 5 analyzes the experimental results, focusing on in-domain detection performance, out-of-domain generalization, and the model's sensitivity to these three edit types. Finally, Chapter 6 summarizes the main contributions of this study and discusses potential directions for future research.

TAIPEI TECH

Chapter 2 Related Work

This chapter reviews related research on speech deepfake detection and describes the research background and problems addressed in this study from the perspectives of evolving attack forms and data generation methods. First, existing speech deepfake detection methods primarily developed for fully deepfake speech are reviewed, with a focus on their mainstream architectures, evaluation benchmarks, and limitations in generalization. Next, this chapter focuses on partial deepfake speech, which has received increasing attention in recent years, explains why it is more difficult to detect than fully deepfake speech, and reviews related detection research and evaluation campaigns. Finally, this chapter examines methods for constructing partial deepfake datasets, ranging from conventional waveform splicing to recent speech editing techniques based on neural codecs and contextual modeling. Differences among existing datasets in terms of generation strategies and language coverage are also compared to highlight the limitations of current research and identify opportunities for further investigation.

2.1 Speech Deepfake Detection

With the rapid advancement of text-to-speech (TTS) and voice conversion (VC) technologies, speech deepfake detection has become a central research topic in speech security and multimedia forensics. Most existing studies formulate the problem as a binary classification task that aims to distinguish deepfake speech generated entirely by generative models from bona fide speech. Several recent survey papers have summarized the technological development of this field and indicated that mainstream approaches can generally be divided into architectures that combine learned feature extractors with classifiers and end-to-end deep learning models that directly take raw waveforms or spectral representations as input. These approaches have achieved significant progress on competition datasets such as ASVspoof [9][10] and ADD [11][12]. However, their performance often depends heavily on the training data and attack types, and their ability to generalize to unknown attacks or cross-dataset scenarios remains limited. In their survey, Menglu Li et al. [13] pointed out that most existing speech deepfake detection studies still

focus on fully deepfake scenarios and predominantly use English speech samples for evaluation, resulting in a gap between current research settings and the diverse and complex attacks encountered in the real world. Similarly, Jiangyan Yi et al. [14] noted that the detection performance of existing methods decreases substantially when they encounter unknown generative models or noise from real-world environments, demonstrating the importance of data diversity.

In recent years, researchers have begun to place greater emphasis on model generalization. Speech DF Arena: A Leaderboard for Speech DeepFake Detection Models [15] established a unified evaluation platform that integrates multiple public datasets and attack scenarios and compares the performance of different speech deepfake detection systems using consistent evaluation metrics. The results showed that even models that perform well on standard datasets may exhibit substantial performance degradation in cross-dataset or out-of-domain scenarios. This finding further demonstrates that existing speech deepfake detection techniques have not yet achieved the robustness and generalization required for practical deployment. These studies also suggest that detecting only fully deepfake speech is no longer sufficient to address real-world threats, highlighting the critical need to investigate more insidious attack forms, such as partial deepfakes.

2.2 Partial Deepfake Speech Detection

Compared with attacks in which an entire utterance is generated, partial deepfake speech, in which only selected segments of an utterance are manipulated, is more difficult to detect and more closely resembles realistic attack scenarios. Such attacks typically replace only keywords, temporal information, or semantically critical segments. Consequently, most of the utterance retains characteristics of the original recording, including environmental noise and artifacts introduced by channel compression, substantially increasing the difficulty of detection for both human listeners and speech deepfake detection systems. Jiangyan Yi et al. [14] identified partial deepfakes as an important emerging category of audio forgery. They noted that partial deepfake speech typically exhibits greater temporal consistency and more natural speech continuity, making it more difficult to detect than fully deepfake speech and presenting a major challenge

to existing speech deepfake detection models.

In recent years, several studies and competitions have begun incorporating partial deepfakes into their evaluation protocols. Examples include the PF task and segment localization task in the ADD challenge, which have encouraged the research focus to shift from simple binary classification toward more fine-grained analysis. However, the large-scale evaluation conducted in Speech DF Arena indicated that most existing detection systems are still primarily designed for fully deepfake attacks. Consequently, their performance is notably unstable on datasets containing partial deepfakes or mixed attack types, revealing persistent deficiencies in current methods for detecting partial deepfakes. Furthermore, the results of the ASVspoof5 challenge [16] showed that although multi-model fusion and data augmentation can reduce error rates on specific datasets, model generalization remains limited in cross-dataset partial deepfake scenarios and under different proportions of manipulated speech. Overall, multiple studies have confirmed both the difficulty and importance of partial deepfake detection. Nevertheless, substantial gaps remain in dataset scale and systematic analysis.

2.3 Partial Deepfake Dataset Generation

To advance research on partial deepfake detection, several datasets have recently been proposed to simulate attack scenarios in which only certain segments of an utterance are manipulated while the remaining content is preserved from a bona fide recording. The PartialSpoof Database, proposed by Lin Zhang et al. [4], is one of the earliest representative datasets in this area. It was constructed by inserting short speech segments generated by TTS or VC systems into bona fide utterances while controlling the duration of the manipulated segments and their proportion within each utterance. PartialSpoof also provides segment annotations at multiple temporal resolutions, allowing researchers to conduct both utterance-level classification and manipulated-segment localization. It has therefore become an important benchmark for subsequent research on partial deepfake detection.

However, the speech editing process used to construct PartialSpoof still relies on conventional splicing techniques, in which synthetic or converted speech segments are directly inserted

into the original recordings. Recent studies have indicated that although this approach can effectively simulate certain attack scenarios, advances in neural codec-based models have gradually reduced the discontinuity artifacts that previously appeared at editing boundaries in partial deepfake speech. Recent models such as VoiceCraft [17] and SSR-Speech [18], for example, can perform highly natural local editing of existing speech while preserving speaker characteristics and speech continuity. In particular, the RealEdit dataset introduced alongside VoiceCraft highlights the seamlessness of such neural speech editing—making partial deepfakes increasingly difficult for human listeners to distinguish—but its lack of anti-spoofing labels and standardized evaluation protocols limits its utility as a security benchmark for speech deepfake detection research.

Against this background, You Zhang et al. [5], together with Lin Zhang, an author of PartialSpoof [4], proposed the large-scale partial deepfake dataset PartialEdit. Their study indicated that neural codec-based speech editing models such as VoiceCraft and SSR-Speech [18] possess contextual modeling capabilities. These models can generate coherent and fluent edited segments according to the original speech and its surrounding context, allowing the acoustic characteristics of the generated speech to transition naturally into the existing content. Consequently, the generated speech does not require additional smoothing procedures or other methods for eliminating audio gaps before the edited segment is inserted into the original utterance through a cut-and-paste operation. Experimental results further demonstrated that context-aware end-to-end models can effectively obscure editing boundaries and reduce obvious forgery artifacts introduced at these boundaries. As a result, speech deepfake detection models can no longer rely primarily on editing-boundary artifacts to identify partially manipulated speech.

2.4 Neural Audio Codec and Speech Editing

The construction of speech deepfake datasets has gradually shifted from conventional waveform splicing toward the direct use of neural speech generation systems. Large-scale datasets such as SpeechFake [19] incorporate diverse TTS, VC, and neural vocoder systems to cover a broader range of speakers, languages, and generation methods. In contrast, CodecFake [20]

focuses on neural audio codecs and generates synthetic samples by encoding and reconstructing bona fide speech with multiple codec models. These studies illustrate how speech generation systems can be used as controllable data-generation tools, allowing dataset developers to vary the synthesis architecture and generation process represented in the resulting data.

Neural audio codecs support this process by converting continuous speech waveforms into compact sequences of discrete acoustic tokens. These tokens can be predicted by sequence models and subsequently reconstructed into waveforms through a codec decoder. This formulation provides an efficient representation for large-scale speech generation and enables the effects of different quantization and decoding configurations to be examined systematically. Nevertheless, neural codec processing alone does not introduce contextual awareness. In codec resynthesis methods such as CodecFake, the original linguistic content is generally preserved, and the generated samples mainly reflect characteristics introduced during codec encoding, quantization, and reconstruction.

Building on neural codec representations, recent speech generation and editing models have further introduced context-aware mechanisms to improve the consistency between generated and retained speech regions. Rather than generating a replacement segment independently, these models condition generation on information from the surrounding utterance, such as neighboring speech tokens, textual content, speaker characteristics, and recording conditions. The SINE dataset [21], for example, was constructed using Voicebox [22], which generates a selected speech region according to both textual and acoustic context. Its subjective evaluation showed that the resulting samples were more difficult to distinguish than conventional cut-and-paste manipulations. PartialEdit similarly employs neural codec-based speech editing models, including VoiceCraft and SSR-Speech, to construct partially manipulated utterances. By incorporating contextual information during generation, these models can better preserve speaker characteristics, prosody, and recording conditions across generated and original regions, thereby reducing obvious acoustic discontinuities.

The use of contextual information makes these generation methods particularly suitable for constructing realistic partial deepfake samples. Because the manipulated region is generated in relation to an existing utterance rather than as an isolated speech segment, detection systems

are less able to depend solely on waveform-splicing artifacts and must instead identify subtler traces associated with the underlying generative process. VoiceCraft-X [7] further extends neural codec-based speech generation and editing to multiple languages, including Mandarin, demonstrating the applicability of these generation paradigms to multilingual contexts.

Overall, neural audio codecs and context-aware generation contribute to speech dataset construction at different stages of the generation process. Neural codecs provide the discrete representations used to model and reconstruct speech, whereas context-aware mechanisms determine how information from the surrounding utterance is incorporated when generating a selected region. Neural codec-based models such as VoiceCraft, VoiceCraft-X, and SSR-Speech combine these two components, enabling the construction of partial deepfake samples with greater contextual and acoustic consistency. Therefore, generated speech datasets should be compared not only by their underlying codec architectures, but also by their conditioning strategies, manipulation scopes, and reconstruction procedures.

2.5 LLM-based Text Manipulation

Large language models have recently been used to automate textual content preparation for speech dataset construction. Noroozi et al. [23] employed Mixtral 8×7B [24] to generate question–answer pairs from existing speech transcripts. Cornell et al. [25] used Llama 3 8B [26] Instruct to produce domain-specific conversational transcripts from a small number of examples. Although these studies primarily generated new textual content rather than modifying existing utterances, they demonstrated that LLMs can efficiently produce diverse and contextually appropriate text for large-scale speech data generation.

More closely related to partial deepfake dataset construction, LlamaPartialSpoof [27] used Llama-3-8B-Instruct to automatically modify existing transcripts. Its prompting strategies covered both local lexical changes and broader modifications to sentence meaning. PartialEdit [5] subsequently applied a similar approach, utilizing GPT-4o-mini [28] to modify selected words in bona fide transcripts while preserving grammatical correctness and fluency. Compared with manually designed replacement rules, these LLM-based approaches provide greater flexibility

in controlling the location and semantic effect of textual modifications, while supporting the systematic generation of linguistically plausible partial deepfake samples.

2.6 Mandarin Speech Dataset

Compared with partial deepfake datasets that primarily focus on English, Half-Truth: A Partially Fake Audio Detection Dataset (HAD)[6] is one of the few partial deepfake speech datasets developed for Mandarin Chinese. HAD uses bona fide speech from AISHELL-3 [29] as its foundation. The original transcriptions are first modified, after which a small number of keywords or local segments are replaced with synthetic speech, while the remaining portions of the original recordings are preserved. Although this dataset construction method produces samples that more closely resemble scenarios involving localized content manipulation, the replacement content must conform to the semantic and syntactic structure of the original sentence. Consequently, the number of words that can be modified and the diversity of the generated samples are restricted to some extent. HAD provides both utterance-level and segment-level annotations, allowing researchers to separately investigate overall classification and local manipulation localization in partial deepfake speech. The experimental results reported in the HAD paper also demonstrated that Mandarin partial deepfake speech containing only a small number of synthetic segments is more difficult to detect than fully deepfake speech. This finding reflects the detection challenges introduced by mixing bona fide and manipulated content and highlights the importance of dataset construction and detection research for Mandarin partial deepfake speech.

2.7 Summary

The studies reviewed above demonstrate that speech deepfake detection research has gradually expanded from the conventional detection of fully deepfake speech to partial deepfakes that more closely resemble realistic attack scenarios. At the same time, partial deepfake data generation techniques have evolved from conventional speech-splicing methods toward speech editing architectures that integrate neural codecs and contextual modeling capabilities. Exist-

ing studies generally indicate that partial deepfakes preserve a substantial amount of bona fide speech characteristics, making the generated speech sound more natural and significantly increasing the difficulty for detection models. Partial deepfake speech has therefore become an important issue that cannot be overlooked in the field of speech security.

Nevertheless, existing research remains predominantly focused on English speech, resulting in relatively limited language coverage. In particular, the generation characteristics, dataset construction methods, and detection performance associated with Mandarin partial deepfake speech remain insufficiently investigated. Although a small number of Mandarin partial deepfake datasets have been proposed, they remain limited in terms of speech naturalness, dataset scale, and the integration of speech editing techniques. These limitations indicate that substantial opportunities remain for the systematic construction and expansion of Mandarin speech datasets. At present, there is also a lack of a dataset capable of effectively simulating realistic Mandarin speech manipulation scenarios, together with a comprehensive experimental analysis of such scenarios. Therefore, this study integrates the data generation methods of PartialEdit and HAD and adopts a speech editing approach based on neural codecs and contextual modeling to construct a Mandarin partial deepfake speech dataset. Furthermore, this study investigates the detection capability and generalization performance of speech deepfake detection models in Mandarin partial deepfake scenarios, thereby addressing existing limitations in Mandarin speech deepfake detection research.

TAIPEI TECH

Chapter 3 Methodology

This study presents a partial deepfake speech dataset tailored for semantic manipulation in Mandarin Chinese. Partial deepfakes preserve most of the original recording and modify only a small amount of text to simulate realistic and deceptive attack scenarios. Building upon the PartialEdit framework, the proposed pipeline utilizes a speech generation model to reconstruct localized audio, effectively eliminating the boundary artifacts typical of manual editing. Moreover, the scope of manipulation is expanded beyond basic substitution to include both deletion and insertion. To prevent these structural changes from yielding nonsensical sentences, the text is modified based on Chinese part-of-speech and semantic rules during the editing phase. These modifications ensure that the manipulated samples remain grammatically coherent and acoustically continuous.

TAIDEI

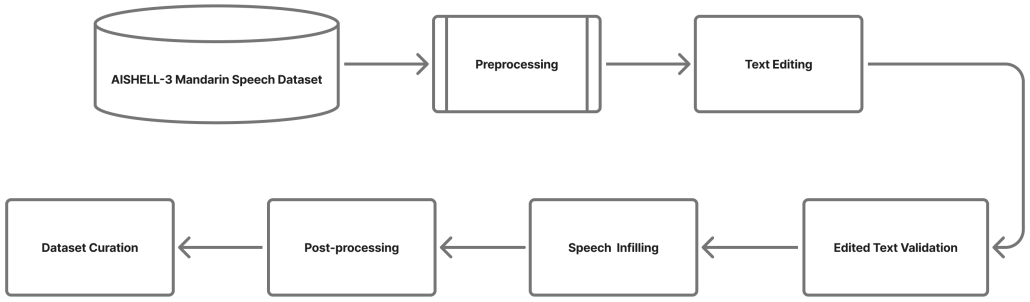


Figure 3.1 Deepfake Speech Generation Pipeline

The overall pipeline of this study is illustrated in Figure 3.1. This chapter describes the seven components of the pipeline in sequence. First, AISHELL-3, the source of the original Mandarin speech data, is introduced. During the Preprocessing stage, consistency checking between the text and speech, forced alignment, Chinese word segmentation, and part-of-speech tagging are sequentially performed to establish structured data for partial editing. Subsequently, during the Text Editing stage, rigorous sample selection and modification rules are defined for the three editing types: deletion, insertion, and substitution.

During the Edited Text Validation stage, the modified text is examined and adjusted for grammatical plausibility. During the Speech Infilling stage, the model generates speech at the specified location according to the text and the start and end times of each character, and blends the generated segment into the original audio to ensure spectral continuity. Subsequently, during the Post-processing stage, the reconstructed speech is realigned, and anomaly detection is applied to directly exclude defective samples. Finally, during the Dataset Curation stage, all valid samples are consolidated, and structured metadata is created to complete the construction of the final Mandarin partial deepfake speech dataset.

3.1 AISHELL-3 Dataset

Following HAD, this study uses AISHELL-3 [29] as the source corpus for constructing a Mandarin partial deepfake speech dataset. AISHELL-3 is a Mandarin Chinese dataset specifically developed for research on multi-speaker text-to-speech systems. The corpus contains approximately 85 hours of speech from 218 native Mandarin speakers, comprising a total of 88,035 utterances. In addition, AISHELL-3 provides transcripts aligned with the corresponding speech recordings, facilitating subsequent text editing.

3.2 Preprocessing

At this stage, the audio recordings and their corresponding transcripts are processed to obtain the textual and temporal information required for subsequent procedures. As shown in Figure 3.2, the preprocessing pipeline consists of three components: transcript verification, forced alignment, Mandarin word segmentation and part-of-speech tagging.

To ensure the correctness of the dataset, Whisper [30] is used to transcribe the original speech recordings, and the generated transcripts are compared with the reference texts provided by AISHELL-3. Samples containing inconsistencies between the two transcripts are discarded. Filtering out these mismatched samples prevents erroneous utterances from affecting the subsequent text-editing and speech-reconstruction processes.

After verification, the transcripts are synchronized with their corresponding audio recordings using the Montreal Forced Aligner (MFA) [31]. This alignment process identifies the start and end times of each character. These temporal boundaries serve as the basis for determining the regions used for speech masking and infilling.

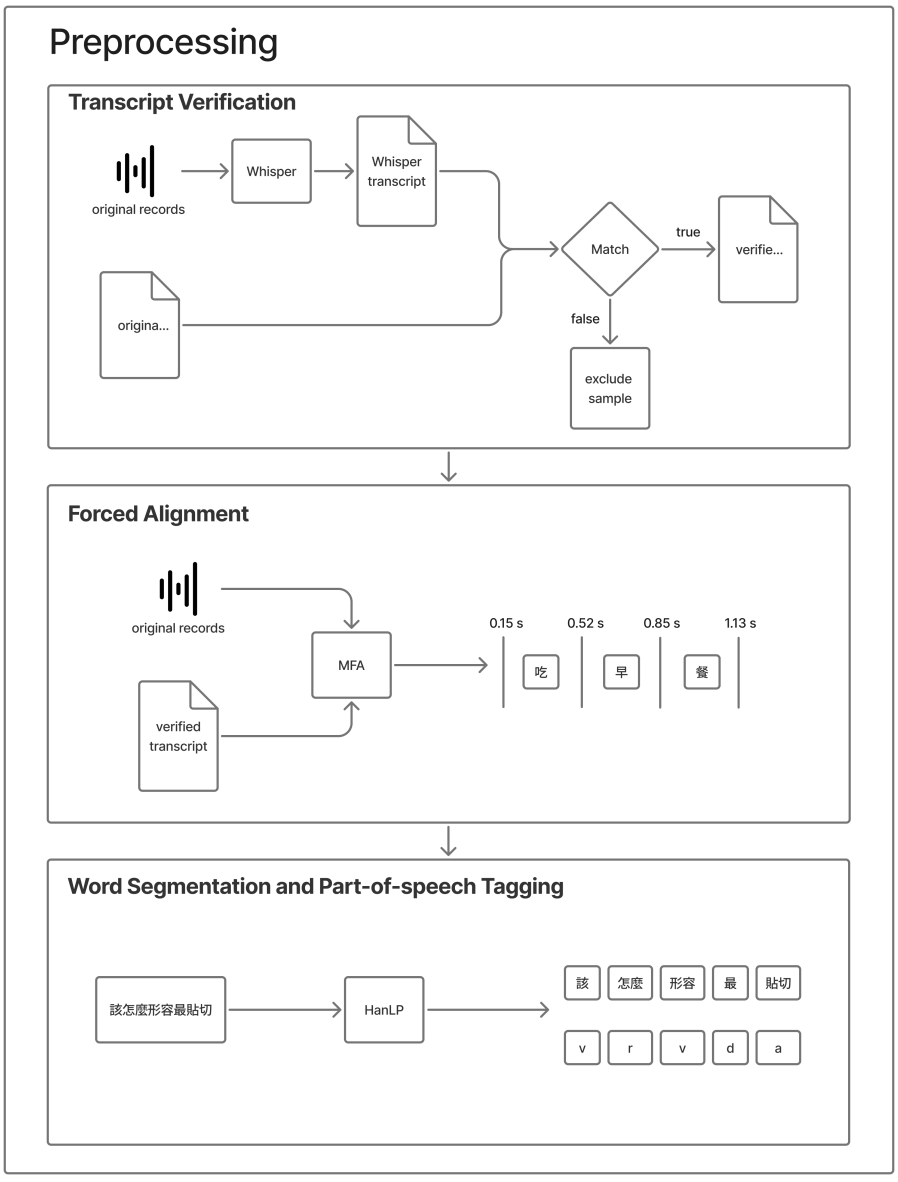


Figure 3.2 Expanded Preprocessing Subprocess

Because written Chinese does not contain explicit word boundaries, HanLP [32] is applied for word segmentation and part-of-speech tagging, including the identification of nouns, verbs, and adjectives. Compared with arbitrarily selecting individual characters, using semantically

complete words as the targets for partial speech editing better preserves the original grammatical structure of the text. Therefore, this part-of-speech-based approach can significantly reduce the generation of semantically flawed speech.

3.3 Text Editing

After data preprocessing, different sample selection and editing rules are defined for the three editing types: deletion, insertion, and substitution. Their target proportions are approximately 10%, 30%, and 60% of the complete dataset, respectively. Because deletion and insertion have received relatively limited attention in existing studies, few operational procedures and implementation examples are available for reference. Therefore, this study adopts more explicit and conservative rules for these two editing operations. Among them, deletion has the strictest eligibility criteria and consequently yields the smallest number of applicable samples. Based on this principle, deletion samples are first selected from the complete corpus, followed by insertion and substitution samples sequentially selected from the remaining unmodified samples. The large language model Phi-3 [33] is then used to generate the text required for the insertion and substitution operations.

The actual processing results show that samples satisfying the deletion rules account for approximately 13% of the complete dataset, which remains close to the predefined target. Because some samples are expected to be excluded during subsequent grammatical validation, this proportion is temporarily retained at this stage. The proportions of insertion and substitution samples can be adjusted according to the amount of remaining data and the dataset construction requirements. In the experiments, 30% and 60% are still used as the primary target proportions. After each type of text editing is completed, the complete modified sentence is written to the edited text field. The original text, editing type, editing location, word-segmented sequence with part-of-speech annotations, and temporal information are also retained for use in subsequent stages. The sample selection criteria and text modification rules for the three editing operations are described below.

TAIPEI TECH

3.3.1 Text Deletion

Algorithm 1 Text Deletion Procedure

edited_text ← None
for all *segment* in word segments of *utterance* **do**
    if POS(*segment*) = Adverb **then**
        negation ← FindNegationCharacter(*segment*)
        position ← Position(*utterance*, *negation*)
        if *negation* exists **and** *position* $\notin \{0, \text{Length}(utterance)\}$ **then**
            edited_text ← RemoveCharacter(*utterance*, *position*)
            break
        end if
    end if
end for
if *edited_text* = None **then**
    numeral ← FindChineseNumeral(*utterance*)
    if *numeral* exists **then**
        if *numeral* is at the beginning of *utterance* **then**
            edited_text ← RemoveNumeralEnding(*utterance*, *numeral*)
        else if *numeral* is at the end of *utterance* **then**
            edited_text ← RemoveNumeralBeginning(*utterance*, *numeral*)
        else
            edited_text ← RemoveNumeralEnding(*utterance*, *numeral*)
        end if
    end if
end if
return *edited_text*

Deletion is relatively uncommon in existing partial deepfake speech datasets. Therefore, this study adopts a simple and explicitly rule-based procedure to reduce the risk of unnatural text or unstable speech boundaries after deletion. This operation is performed entirely according to predefined part-of-speech, lexical, and positional rules, without using a large language model to modify the text.

To ensure that the locally manipulated segment created through deletion has clear and identifiable boundaries, words located at the beginning or end of an utterance are not deleted. If the only eligible deletion target in an utterance occurs at the beginning or end of the sentence and no other deletable segment exists internally, no deletion operation is applied. The utterance is instead retained for subsequent editing operations.

This study first searches for utterances containing adverbial word segments and then determines whether these adverbs contain specific negation characters. This design is based on the observation that Chinese negation words are generally tagged as adverbs. Therefore, part-

of-speech annotations can first narrow the search scope, after which the characters are checked to determine whether they meet the deletion conditions. The negation characters considered in this study are “不、沒、無、非、否、未、莫、勿、毋、甭”. When an adverbial word segment contains one of these negation characters and is not located at the beginning or end of the sentence, the corresponding negation character is deleted, and the complete modified sentence is written to the edited text field. For example, in the sentence “這句話不好笑”, the negation character “不” can be deleted to produce “這句話好笑”.

Samples obtained solely by deleting a negation character account for approximately 7% of the complete AISHELL-3 dataset, which does not reach the target proportion for deletion. Therefore, deletion is additionally performed on text containing numbers. Because numerical values in the AISHELL-3 transcripts are represented using Chinese numerals, such as “三十六萬五千一百八十”, this study deletes a numeral component from the beginning or end of a numeral string according to the compositional structure of Chinese numerals. This operation changes the original numerical value while preserving the remaining content.

In Chinese numerals, “十、百、千、萬、億” represent place values. Therefore, the component to be deleted must be determined according to the place-value character and its adjacent numeral character. At the beginning of a numeral string, the first place-value character and the numeral character immediately preceding it may be deleted together. For example, deleting “三十” from “三十六萬五千一百八十” produces “六萬五千一百八十”. At the end of a numeral string, the unit numeral character is preferentially deleted. If no numeral character follows the final place value character, the final place value character and the numeral character immediately preceding it are deleted together. For example, “三十六萬五千一百八十” ends with “八十”, and no units digit follows “十”. Therefore, the numeral component “八十” can be deleted to produce “三十六萬五千一百”.

The direction of numeral deletion also depends on the position of the numeral within the sentence. If the numeral occurs at the beginning of the sentence, only part of the value at the end of the numeral string is removed to avoid deleting audio content at the beginning of the complete utterance. If the numeral occurs at the end of the sentence, only part of the value at the beginning of the numeral string is removed to preserve the sentence ending. If the numeral

occurs in the middle of the sentence, deletion is performed from the end of the numeral string, provided that the actual deletion interval does not extend to the beginning or end boundary of the complete sentence. If no modification satisfying these rules can be produced without deleting content at the beginning or end of the sentence, deletion is not performed on the text.

By applying the two deletion rules based on negation characters and Chinese numerals, deletion samples account for slightly more than 10% of the complete dataset, reaching 13%. Therefore, no additional deletion operations are applied to other lexical categories.

3.3.2 Text Insertion

Algorithm 2 Text Insertion Procedure

edited_text ← None
inserted_index_list ← [ ]
for all *segment* in word segments of *utterance* **do**
    previous_segment ← PreviousSegment(*utterance*, *segment*)
    if POS(*segment*) ∈ {Adjective, Verb} **and** POS(*previous_segment*) ≠ Adverb **then**
        position ← Position(*utterance*, *segment*)
        inserted_word ← GenerateAdverb(*utterance*, *position*)
        edited_text ← InsertBefore(*utterance*, *position*, *inserted_word*)
        inserted_index_list ← [*position*]
        break
    else if POS(*segment*) = Noun **and** POS(*previous_segment*) ≠ Adjective **then**
        position ← Position(*utterance*, *segment*)
        inserted_word ← GenerateAdjective(*utterance*, *position*)
        edited_text ← InsertBefore(*utterance*, *position*, *inserted_word*)
        inserted_index_list ← [*position*]
        break
    end if
end for

Following the selection of deletion samples, utterances satisfying the insertion conditions are identified from the remaining unmodified data. Compared with deletion, insertion can be applied to a wider range of part-of-speech combinations, and a sufficient number of eligible utterances are available. It is therefore easier to achieve the predefined data proportion.

Two insertion conditions are defined. The first applies to sentences in which the word segment preceding an adjective or verb is not an adverb. The second applies to sentences in which the word segment preceding a noun is not an adjective. For the first type of sample, an adverb is inserted before the target adjective or verb to add information concerning degree, scope, negation, or other forms of modification. For the second type, an adjective is inserted

before the target noun to modify or supplement the description of a person, object, or concept.

After the insertion position has been determined, the language model generates an appropriate adverb or adjective based on the surrounding context and target position, and the newly generated word is inserted at the target location. This method first constrains the eligible insertion positions according to part-of-speech conditions and then uses the language model to generate a contextually appropriate modifier. This reduces inconsistencies between the inserted content and the original syntactic or semantic structure. The complete modified sentence is subsequently written to edited text, and an inserted index list field is created to record the position of the newly inserted word segment. The information provided by this field is used during subsequent text validation to support further text editing.

Because a sufficient number of utterances satisfy the two conditions described above, this operation continues until insertion samples account for approximately 30% of the complete AISHELL-3 dataset. Once the predefined proportion is reached, the remaining unmodified samples are retained for substitution.

3.3.3 Text Substitution

Substitution accounts for the largest proportion among the three editing types in this study, with a target proportion of approximately 60% of the complete dataset. Compared with deletion and insertion, substitution does not alter the original sentence structure and can be applied to a wider variety of lexical categories. It is therefore easier to generate a sufficient number of modified texts with complete syntactic structures.

A priority order is established according to lexical category when selecting substitution targets. Proper nouns, including personal names, place names, and organization names, are selected first. Replacing such words can directly change the participants, locations, or related entities in an event while preserving the original syntactic structure. If a sentence contains no proper nouns, common nouns and verbs are selected sequentially. Replacing a noun can change the person, object, concept, or object of an event, whereas replacing a verb can alter the action, state, or event described in the original utterance.

After the lexical-category filtering described above, if any text remains unassigned to an

Algorithm 3 Text Substitution Procedure

edited_text ← None
substituted_index_list ← [ ]
target_segment ← None
for all *segment* in word segments of *utterance* **do**
    if POS(*segment*) = ProperNoun **then**
        target_segment ← *segment*
        break
    end if
end for
if *target_segment* = None **then**
    for all *segment* in word segments of *utterance* **do**
        if POS(*segment*) = CommonNoun **then**
            target_segment ← *segment*
            break
        end if
    end for
end if
if *target_segment* = None **then**
    for all *segment* in word segments of *utterance* **do**
        if POS(*segment*) = Verb **then**
            target_segment ← *segment*
            break
        end if
    end for
end if
if *target_segment* = None **then**
    target_segment ← RandomSelect(*utterance*)
end if
position ← Position(*utterance*, *target_segment*)
replacement_word ← GenerateReplacement(*utterance*, *position*)
edited_text ← ReplaceSegment(*utterance*, *position*, *replacement_word*)
substituted_index_list ← [*position*]

TAIPEI TECH

editing type, one word segment is randomly selected from its word segmented sequence as the substitution target to obtain the required number of substitution samples. Because the target word segment is randomly selected for some substitution samples, these operations are more likely to produce invalid lexical collocations, anomalous syntactic structures, or contextual inconsistencies. Therefore, after substitution is completed, the grammatical correctness and contextual plausibility of the modified text are further checked, and samples that do not satisfy the requirements are excluded.

For each type of substitution target described above, the language model generates an semantically consistent replacement word based on the target word segment and its context. The generated word is used to replace the original target segment, and the complete modified sentence is written to the edited text field. A substituted index list field is also created to record the position of the replaced word segment in the original word-segmented sequence. This information is subsequently used during text validation to support further text editing.

3.4 Edited Text Validation

After text editing is completed, NaSGEC-BART[34] is used to perform grammatical validation on edited text and determine whether deletion, insertion, or substitution has introduced anomalies into the grammatical structure. If the modified text passes validation, the sample is retained and proceeds to the subsequent speech generation pipeline. If the text contains grammatical errors, an appropriate reprocessing strategy is applied according to its original editing type. Subsequently, the proportions of deletion, insertion, and substitution among the valid samples that pass validation and are retained are 10%, 28%, and 62%, respectively.

During this process, inserted index list and substituted index list are used to record the editing positions previously attempted for insertion and substitution, respectively. This prevents repeated selection of the same position and allows the search to continue for alternative candidate word segments. The processing strategy for each editing type is described below.

3.4.1 Deleted Text Validation

Deletion is performed entirely according to predefined rules, without using a large language model to generate or modify the text. Because applying the same deletion rules to the same text again would produce the same result, the deletion operation is not repeated when a deletion sample fails grammatical validation.

For such samples, substitution, which has fewer constraints, is instead applied to the original text. After substitution is completed, the sample enters the substitution validation pipeline. This approach avoids repeatedly generating the same erroneous result.

3.4.2 Inserted Text Validation

Algorithm 4 Inserted Text Validation Procedure

```
1:  if GrammaticalValidation(edited_text) = Fail then
2:      candidate_positions ← FindInsertionPositions(utterance)
3:      candidate_positions ← RemoveAttemptedPositions(candidate_positions, inserted_index_list)
4:      if candidate_positions is not empty then
5:          insertion_position ← First(candidate_positions)
6:          edited_text ← TextInsertion(utterance, insertion_position)
7:          Append(inserted_index_list, insertion_position)
8:      else
9:          TerminateInsertion
10:         TrySubstitution(utterance)
11:     end if
12: end if
```

If an insertion sample fails grammatical validation, previously attempted insertion positions are first excluded according to inserted index list, and the next candidate is selected from the remaining positions that satisfy the insertion conditions. After a new insertion position is selected, a contextually appropriate word is regenerated and edited text is updated.

Each attempted insertion position is recorded in inserted index list to prevent the same position from being selected again in subsequent attempts. This process continues until the text passes grammatical validation or no other eligible insertion position remains in the sentence. If all available insertion positions have been attempted but the modified text still contains grammatical errors, the insertion operation is terminated. Substitution is then performed on the original text, and the sample proceeds to the substitution validation pipeline.

3.4.3 Substituted Text Validation

Algorithm 5 Substituted Text Validation Procedure

```
if GrammaticalValidation(edited_text) = Fail then
    candidate_positions ← FindSubstitutionPositions(utterance)
    candidate_positions ← RemoveAttemptedPositions(candidate_positions, substituted_index_list)
    if candidate_positions is not empty then
        substitution_position ← RandomSelect(candidate_positions)
        edited_text ← TextSubstitution(utterance, substitution_position)
        Append(substituted_index_list, substitution_position)
    else
        MarkSampleInvalid
    end if
end if
```

If a substitution sample fails grammatical validation, previously attempted word segments are excluded according to substituted index list, and a new substitution target is randomly selected from the remaining word segments that have not yet been attempted. After the target is selected, a new replacement word is generated and edited text is updated.

The position of each word segment selected as a substitution target is recorded in substituted index list to prevent repeated attempts at the same position. This process continues until the text passes grammatical validation or all candidate word segments in the sentence have been attempted. If all candidates have been attempted but the modified text still contains grammatical errors, the sample is marked as invalid and excluded from the subsequent speech generation and dataset construction pipelines.

3.5 Speech Infilling

After text validation is completed, VoiceCraft-X [7] is used to perform local speech reconstruction. Based on the original speech, original text, edited text, and specified editing position, VoiceCraft-X generates a local speech segment that transitions naturally into the preceding and following audio. This method ensures that the generated audio remains highly consistent with the original speech in terms of speaker timbre, prosody, and background noise.

Because deletion, insertion, and substitution introduce different changes at the textual level, their corresponding speech masking ranges and reconstruction procedures also differ. The speech

infilling procedures for the three editing operations are described below.

3.5.1 Speech Deletion

Although text editing deletes only the target word, directly removing the corresponding audio introduces spectral discontinuities. To prevent this, the masked region for speech reconstruction via VoiceCraft-X encompasses both the target word and its preceding word segment. Because the preceding word segment is retained in the edited text, VoiceCraft-X performs speech reconstruction on this segment to maintain continuous local acoustic boundaries.

3.5.2 Speech Insertion

The speech reconstruction procedure for insertion is simpler than that for deletion. VoiceCraft-X generates coherent speech according to the context. Because this process requires no audio splicing, it does not disrupt spectral continuity at the speech boundaries.

3.5.3 Speech Substitution

The speech reconstruction procedure for substitution is similar to that for insertion. However, VoiceCraft-X first masks the target region to generate the replacement speech, which similarly requires no audio splicing.

3.6 Post-processing

Post-processing includes speech-text realignment and anomaly detection. To avoid introducing additional artifacts, anomalous samples are directly excluded rather than repaired artificially.

MFA performs forced alignment between the generated speech and the modified text. Because the earlier stages alters the speech duration, realignment provides updated timestamps to reannotate the manipulated segments.

Anomaly detection comprises both acoustic and semantic evaluations. First, audio boundaries are checked for anomalous pauses, intervals, or excessive volume fluctuations, and samples exhibiting these defects are directly excluded. Furthermore, Whisper transcribes the generated speech to verify content consistency against the target text. Any mismatched samples are also discarded. This overall exclusion mechanism prevents subsequent models from overfitting to artifacts during training.

3.7 Dataset Curation

Upon completion of the validation and post-processing stages, the partial deepfake speech samples that pass the screening are consolidated, and corresponding metadata is generated to support subsequent model training and analysis. Each metadata record includes the utterance identifier, speaker identifier, text content before and after editing, editing operation type, part-of-speech annotation sequence, character-level start and end times, start and end times of the manipulated segment, and a sample-validity indicator.

The final dataset retains only those samples that pass the text validation, speech-content consistency, and anomaly detection procedures. Supported by this structured metadata, the dataset can support both utterance-level deepfake speech detection and segment-level manipulation localization.

TAIPEI TECH

Chapter 4 Dataset Utility Evaluation

This chapter outlines the framework for evaluating dataset utility, assessing the capacity of MPDS to train models for detecting partially manipulated speech. The HAD and PartialSpoof datasets serve as comparative benchmarks, utilizing AASIST [8] as the baseline model.

The subsequent sections detail the evaluation framework: dataset preparation and partitioning, the AASIST network architecture and fine-tuning strategy, and the training configurations. Finally, the evaluation metrics for distinguishing bona fide from spoofed speech are defined to standardize the performance analysis.

4.1 Evaluation Framework

This study establishes a three-dimensional evaluation framework. For the control setup, MPDS, HAD, and PartialSpoof serve as independent training datasets, resulting in three corresponding speech deepfake detection models: MPDS-AASIST, HAD-AASIST, and PartialSpoof-AASIST.

To isolate the effect of the training data, all other variables are kept constant. The three models share the identical AASIST architecture, audio preprocessing pipeline, training hyperparameters, and selection criteria during fine-tuning. Additionally, a single global random seed is applied throughout the experiments to ensure reproducibility.

This controlled configuration ensures that performance variations among the models are directly attributable to their respective training datasets. Utilizing these three models, the evaluation framework proceeds with the following three principal testing scenarios.

- **In-domain Evaluation**: Each model is evaluated on the test set of its source domain. This stage examines whether each training dataset enables the corresponding model to effectively learn the relevant manipulation characteristics. The results serve as the reference baseline for the subsequent cross-dataset experiments.

- **Cross-dataset Evaluation**: To analyze the cross-dataset generalization ability of each model when confronted with different languages, data distributions, and unknown spoof-

ing attacks, each model is evaluated on target test sets that do not originate from its training dataset. By comparing model performance under in-domain and cross-dataset conditions, this evaluation investigates whether the characteristics learned from different training datasets can be effectively transferred to unseen data and attack scenarios.

- **Edit-type Evaluation**: Given the technical diversity of partial manipulation, this evaluation framework specifically examines how different training datasets affect model performance in detecting three types of local editing: deletion, insertion, and substitution. By independently evaluating test subsets corresponding to specific editing types, the sensitivity of each model to different partial manipulation techniques can be assessed.

4.2 Baseline Model

To evaluate the utility of the MPDS dataset for training speech deepfake detection models, this study adopts AASIST as the baseline. A primary advantage of AASIST is its lightweight architecture; comprising only 85,306 parameters, it operates without relying on computationally expensive score-level ensembles. This compact design facilitates the evaluation of dataset-driven generalization while mitigating the risk of model overfitting.

Architecturally, AASIST is an end-to-end model that integrates spectral and temporal information. Its front end employs a RawNet2-based encoder to extract representations directly from raw speech, constructing these in parallel as two heterogeneous graphs for the spectral and temporal domains. To integrate this heterogeneous information, the model incorporates a Heterogeneous Stacking Graph Attention Layer (HS-GAL) and a Max Graph Operation (MGO).

The HS-GAL uses a data-driven attention mechanism to assess the relationship importance among spectro-temporal nodes, capturing manipulation characteristics across diverse temporal and frequency regions. The MGO consists of two parallel branches with identical structures. Through an element-wise maximum operation, the outputs are compared across each feature dimension, retaining the larger value to encourage the branches to learn distinct manipulation characteristics. Finally, combining graph pooling layers with a readout scheme, AASIST aggregates complex local node features into a global representation. This feature vector is passed

to a classifier, outputting a score to distinguish bona fide from manipulated speech.

4.3 Dataset Preparation for Training

This study utilizes the official pre-trained AASIST model as a foundation for fine-tuning on the MPDS, HAD, and PartialSpoof datasets. In this setup, MPDS serves as the primary dataset, with its bona fide samples sourced directly from the AISHELL-3 Mandarin corpus, while PartialSpoof and HAD act as control groups. To ensure a consistent benchmark across these diverse data sources during fine-tuning and evaluation, all audio samples undergo standard preprocessing.

All audio is resampled to 16 kHz and encoded as single-channel 16-bit PCM. This ensures the signal representation matches the input specifications of the pre-trained AASIST weights.

To maintain uniform input dimensions for batch processing, the input length is standardized to 64,600 samples (roughly 4.04 seconds). For samples exceeding this limit, the cropping process is guided by the time annotations provided in each dataset. A segment containing the manipulated region and its adjacent context is isolated such that its length does not exceed 64,600 samples. Both naturally shorter audio and these cropped segments are subsequently zero-padded to reach the 64,600-sample requirement. This strategy ensures that the manipulated regions and their transition boundaries are fully captured.

Following preprocessing, all three datasets are partitioned into training, development, and evaluation sets. Within each individual dataset, subsets are divided to strictly maintain speaker isolation. Beyond intra-dataset isolation, the speaker distribution across different datasets, particularly between MPDS and HAD, must also be explicitly addressed. Since both datasets source their bona fide speech from the AISHELL-3 corpus, they inherently share the same pool of speakers. If this overlap is not properly controlled, the training data of an MPDS-AASIST could share speaker identities with the HAD evaluation set, causing the model to exploit speaker familiarity rather than learning generalized spoofing artifacts. To eliminate this potential source of cross-dataset leakage, an additional speaker constraint is imposed across the two datasets. To prevent cross-dataset speaker leakage, speaker identities appearing in the training set of one

dataset were excluded from the evaluation set of the other dataset. This ensures that cross-dataset evaluation reflects the model's ability to generalize to unseen speakers rather than benefiting from speaker-specific information encountered during training. The same constraint is applied in the reverse direction when a HAD-AASIST is evaluated on the MPDS evaluation set. Without such cross-dataset speaker isolation, the resulting performance metrics could be biased even in the absence of overlapping speech utterances.

In addition to managing speaker identities, a structural modification is applied to the official HAD dataset. Because its original test set contains only manipulated samples, it lacks the positive class required to calculate metrics such as the Equal Error Rate (EER). To resolve this, 8,975 bona fide samples are reallocated from the HAD training set to its test set. This re-splitting establishes an approximate $1:1$ balance between bona fide and manipulated samples, enabling valid performance metric calculations. The detailed sample distribution for each dataset is summarized in Table 4.1.

Table 4.1 Summary of Dataset Splits for MPDS, HAD, and PartialSpoof

Dataset	Split	Bona Fide	Spoofed	Total
PartialSpoof	Train	2,580	22,800	25,380
	Dev	2,548	22,296	24,844
	Eval	7,355	63,882	71,237
HAD	Train	17,579	26,539	44,118
	Dev	8,914	8,910	17,824
	Eval	8,975	9,072	18,047
MPDS	Train	3,329	32,842	36,171
	Dev	3,188	32,328	35,516
	Eval	5,891	6,057	11,948

4.4 Experimental Configuration

Experiments are implemented in PyTorch and conducted on a single NVIDIA GeForce RTX 4060 GPU (8GB VRAM). The three models MPDS-AASIST, HAD-AASIST, and PartialSpoof-AASIST are initialized with the official pre-trained AASIST weights. Each model is fine-tuned on its respective training set. A global random seed of 1234 is applied to ensure reproducibility.

Table 4.2 Summary of Experimental Setup and Hyperparameters

Configuration	**Value**
Random Seed	1234
Audio Input Length	64,600 samples
Batch Size	24
Optimizer	Adam
Learning Rate	10^{-4}
Weight Decay	10^{-4}
Max Epochs	100
Early Stopping Patience	10 Epochs

Hyperparameter configurations are detailed in Table 4.2. The training batch size is set to 24. Optimization is performed using Adam, with an initial learning rate of 10^{-4} and a weight decay of 10^{-4}.

Training is scheduled for a maximum of 100 epochs. Model selection is based on the Equal Error Rate (EER) evaluated on the development set; the checkpoint yielding the lowest EER is saved at the end of each epoch. An early stopping mechanism terminates training if the development EER or loss does not improve for 10 consecutive epochs. The best-performing checkpoint is subsequently retained for final evaluation.

4.5 Evaluation Metrics

This study adopts the Equal Error Rate (EER) as the primary evaluation metric. As a standard metric in speech anti-spoofing, EER measures a model's overall capability to distinguish bona fide from spoofed speech. EER is derived from the False Alarm Rate (FAR) and False Rejection Rate (FRR), defined as follows:

- **False Alarm Rate (FAR)**: The proportion of spoofed speech samples incorrectly classified as bona fide speech.
- **False Rejection Rate (FRR)**: The proportion of bona fide speech samples incorrectly classified as spoofed speech.

Mathematically, FAR and FRR as functions of the threshold θ can be expressed as follows:

$$FAR(\theta) = \frac{FP(\theta)}{FP(\theta) + TN(\theta)}\,.$$

$$FRR(\theta) = \frac{FN(\theta)}{FN(\theta) + TP(\theta)}\,.$$

Here, θ denotes the decision threshold applied to the model output score. In this study, bona fide speech is defined as the positive class, whereas spoofed speech is defined as the negative class. Based on the model predictions, the samples can be divided into the following four categories:

- TP: True positive, representing the number of bona fide speech samples correctly classified as bona fide speech.
- TN: True negative, representing the number of spoofed speech samples correctly classified as spoofed speech.
- FP: False positive, representing the number of spoofed speech samples incorrectly classified as bona fide speech.
- FN: False negative, representing the number of bona fide speech samples incorrectly classified as spoofed speech.

As the decision threshold θ changes, TP, TN, FP, and FN also change, consequently affecting FAR and FRR. When FAR and FRR are equal or closest to each other, the corresponding error rate is defined as the EER:

$$EER = FAR(\theta^*) = FRR(\theta^*)\,.$$

Here, θ^* denotes the threshold at which FAR and FRR are equal or closest to each other. A lower EER indicates that the model is better able to distinguish bona fide speech from spoofed speech. Because EER jointly accounts for both types of error, it is suitable for comparing the overall spoof detection performance of different models or under different dataset conditions.

TAIPEI TECH

Chapter 5 Results and Analysis

This chapter analyzes the detection performance of speech anti-spoofing models fine-tuned on different datasets, evaluating the practical utility of MPDS. The experiments are structured around three evaluations: in-domain, cross-dataset, and edit-type. First, the detection performance of each model is compared within its respective source domain. Second, cross-dataset generalization is analyzed across different languages, data distributions, and unseen spoofing attacks. Finally, the study investigates how different training datasets affect model performance across the three partial editing types: deletion, insertion, and substitution.

To evaluate the MPDS dataset against existing benchmarks, this study utilizes HAD and PartialSpoof to train three comparative models: MPDS-AASIST, HAD-AASIST, and PartialSpoof-AASIST. To isolate the effect of the training data, all three models share identical pretrained weights, AASIST architectures, audio preprocessing pipelines, training hyperparameters, and model selection procedures. The fine-tuning dataset serves as the sole independent variable across the models.

These controls reduce potential confounding factors and allow the effects of different training datasets to be examined under a consistent experimental setting.

5.1 In-domain and Cross-dataset Evaluation

To evaluate the spoof detection performance of models trained on different datasets and their generalization ability when confronted with cross-lingual data and unknown attacks, two evaluation scenarios are designed: in-domain evaluation and cross-dataset evaluation.

In the in-domain evaluation, MPDS-AASIST, HAD-AASIST, and PartialSpoof-AASIST are evaluated on the test sets of their respective training-source datasets. This evaluation examines the effect of each training dataset on the model's spoof detection capability and provides a reference baseline for the subsequent cross-dataset experiments.

In the cross-dataset evaluation, each model is evaluated on target test sets that do not originate from its training dataset. This evaluation analyzes model generalization when the data

distribution, language, or attack type changes. By comparing model performance under in-domain and cross-dataset conditions, it is possible to further investigate whether the characteristics learned from different training datasets can be effectively transferred to unseen data and attack scenarios.

Table 5.1 In-domain and Cross-dataset Evaluation Results

Train	Eval	EER (%)
MPDS	MPDS (In-domain)	1.21
	HAD	13.76
	PartialSpoof	29.21
HAD	MPDS	24.88
	HAD (In-domain)	0.94
	PartialSpoof	32.85
PartialSpoof	MPDS	37.91
	HAD	33.78
	PartialSpoof (In-domain)	1.43

5.1.1 In-domain Evaluation

Table 5.1 shows that the EER values of all three models on their respective training-source datasets are below 1.5%. MPDS-AASIST achieves an EER of 1.21% in the in-domain evaluation, indicating that the model effectively learns the characteristics contained in MPDS. HAD-AASIST and PartialSpoof-AASIST achieve EER values of 0.94% and 1.43%, respectively, demonstrating comparable performance. The in-domain evaluation results confirm that each dataset enables successful model training and provide a reference baseline for the subsequent analysis of cross-dataset generalization.

5.1.2 Cross-dataset Evaluation

As shown in Table 5.1, MPDS-AASIST achieves lower EER values in both cross-dataset evaluations than the corresponding cross-dataset EER values of the other two models. This result indicates that the spoof detection characteristics learned from MPDS exhibit stronger cross-dataset transferability. In particular, MPDS-AASIST achieves an EER of 13.76% on the HAD

test set, substantially outperforming the 33.78% achieved by PartialSpoof-AASIST. Because both MPDS and HAD primarily contain Mandarin speech, their shared language may reduce some differences in linguistic and acoustic distributions, resulting in a more apparent transfer advantage.

On the PartialSpoof test set, MPDS-AASIST achieves an EER of 29.21%, which is also lower than the 32.85% achieved by HAD-AASIST. Furthermore, this result is better than the performance of PartialSpoof-AASIST on the MPDS and HAD test sets. This suggests that the speech and manipulation characteristics represented in MPDS may be more transferable, allowing the model to retain a certain degree of spoof detection capability when confronted with different languages and unseen data distributions.

By comparison, the EER values of PartialSpoof-AASIST on the two Mandarin datasets increase substantially relative to its in-domain evaluation result, indicating that its learned characteristics are more difficult to transfer to Mandarin speech data. However, because differences in language, data source, spoofing method, and recording conditions occur simultaneously, the observed performance degradation cannot be attributed entirely to language. Overall, the experimental results demonstrate that MPDS improves the cross-dataset generalization ability of the model. Nevertheless, speech anti-spoofing models remain susceptible to differences in language and data distribution.

5.2 Edit-type Evaluation

The MPDS dataset constructed in this study contains three editing types: Deletion, Insertion, and Substitution. To evaluate model sensitivity to different partial manipulation techniques in greater detail, the MPDS test set is divided into three corresponding subsets. The construction of each subset follows the two rules below:

- The spoofed speech samples in each subset correspond to one editing type, while the same set of bona fide speech samples is used across all subsets to avoid evaluation bias caused by randomly sampling different bona fide speech samples.

- The ratio of bona fide speech samples to spoofed speech samples in each subset is maintained at approximately $1:1$ to prevent evaluation bias caused by differences in class proportions.

MPDS-AASIST, HAD-AASIST, and PartialSpoof-AASIST are then independently evaluated as binary classifiers on each of the three subsets.

Table 5.2 shows that MPDS-AASIST achieves EER values of 2.96%, 1.13%, and 1.07% for Deletion, Insertion, and Substitution, respectively. The model maintains strong spoof detection performance across all three editing types and does not exhibit an ability limited to recognizing only a specific editing pattern. Among the three types, Deletion produces a slightly higher EER than Insertion and Substitution. This difference may be related to the approximate $1:3:6$ ratio of Deletion, Insertion, and Substitution samples in MPDS. Because relatively fewer Deletion samples are available for training, the model has fewer relevant characteristics from which to learn. Nevertheless, its EER remains below 3%, indicating that the model can still effectively learn the partial manipulation characteristics introduced by Deletion.

By comparison, HAD-AASIST and PartialSpoof-AASIST exhibit poorer detection performance on Deletion and relatively stronger performance on Substitution. The performance variation across editing types is particularly pronounced for HAD-AASIST, suggesting that its spoof detection capability may be more strongly oriented toward specific editing patterns.

Overall, MPDS-AASIST maintains consistently low EER values across all three editing types. This result demonstrates that the diverse editing samples contained in MPDS enable the model to learn multiple partial manipulation patterns rather than relying solely on manipulation characteristics associated with a single editing type.

TAIPEI TECH

Table 5.2 Evaluation Results across Different Edit Types on MPDS Subsets

Train	MPDS Subset	EER (%)
MPDS	Deletion	2.96
	Insertion	1.13
	Substitution	1.07
HAD	Deletion	46.38
	Insertion	30.07
	Substitution	19.84
PartialSpoof	Deletion	43.71
	Insertion	37.57
	Substitution	35.34

TAIPEI TECH

Chapter 6 Conclusion and Future Work

This chapter summarizes the core findings derived from the experimental results and outlines specific directions for future research to further advance speech deepfake detection methodologies.

6.1 Conclusion

This study addresses the growing threat posed by partial deepfake speech in the field of speech deepfake detection by constructing MPDS, a Mandarin partially manipulated speech dataset. Existing research indicates that most speech anti-spoofing models are primarily designed for fully deepfake speech and exhibit substantial performance degradation when confronted with partial deepfakes that preserve a large proportion of the acoustic and environmental characteristics of bona fide recordings. Furthermore, most existing partial deepfake datasets primarily contain English speech, and early data generation methods largely rely on conventional speech-splicing techniques, which tend to introduce conspicuous artifacts at editing boundaries.

To address this research gap, this study proposes a dataset construction pipeline based on the Mandarin corpus AISHELL-3. Unlike conventional approaches that focus on a single editing operation, this study includes three editing types: deletion, insertion, and substitution. In addition, part-of-speech and semantic filtering rules are applied to ensure the grammatical plausibility of the modified text. During speech reconstruction, a context-aware speech generation model is employed to perform local speech infilling, allowing the generated segments to blend into the original audio to ensure spectral continuity.

To evaluate dataset utility, this study adopts AASIST as the baseline model and establishes an evaluation framework comprising in-domain, cross-dataset, and edit-type evaluations. The experimental results demonstrate that the MPDS-AASIST model, fine-tuned using MPDS, not only achieves an EER of 1.21% in the in-domain evaluation but also exhibits stronger cross-dataset generalization than models trained on the HAD and PartialSpoof datasets. MPDS-AASIST maintains consistent detection performance across different editing types, indicating

that the model can learn manipulation characteristics across different edit types even though deletion and insertion account for a relatively small proportion of the dataset. These results demonstrate the practical utility of MPDS in improving the ability of speech anti-spoofing models to detect partially manipulated speech.

6.2 Future Work

Although this study has achieved promising results in the construction and evaluation of Mandarin partial deepfake speech data, several directions remain worthy of further investigation:

- **Expanding dataset diversity and simulating real-world scenarios**: MPDS is currently constructed using AISHELL-3, which consists of clean speech recordings acquired in controlled environments. Future research could incorporate speech recorded under practical conditions involving background noise, reverberation, or various recording devices. The dataset could also be extended to multi-speaker conversations and to further evaluate and improve the robustness of speech anti-spoofing models in complex acoustic environments.

- **Exploring a wider variety of partial manipulation techniques**: In addition to deletion, insertion, and substitution, future attacks may involve changes in speaking style, emotion manipulation, or voice conversion applied to specific phrases. Incorporating these fine-grained generation techniques into the dataset would facilitate the development of defenses against more subtle forms of speech manipulation.

- **Investigating cross-lingual spoof detection characteristics**: This study indicates that cross-lingual and cross-dataset generalization remains challenging even when a model maintains stable performance on Mandarin speech. Future research could investigate whether universal local manipulation artifacts exist across language boundaries.

TAIPEI TECH

References

[1] Dennis H Klatt. “Review of Text-to-Speech Conversion for English”. In: *The Journal of the Acoustical Society of America* 82.3 (1987), pp. 737–793.

[2] Masanobu Abe, Satoshi Nakamura, Kiyohiro Shikano, and Hisao Kuwabara. “Voice Conversion Through Vector Quantization”. In: *Journal of the Acoustical Society of Japan (E)* 11.2 (1990), pp. 71–76.

[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*. 2021, pp. 4264–4268. doi: 10.21437/Interspeech.2021-738.

[4] 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”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 31 (2022), pp. 813–825.

[5] You Zhang, Baotong Tian, Lin Zhang, and Zhiyao Duan. “PartialEdit: Identifying Partial Deepfakes in the Era of Neural Speech Editing”. In: *Interspeech 2025*. 2025, pp. 5353–5357. doi: 10.21437/Interspeech.2025-942.

[6] 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: *Interspeech 2021*. 2021, pp. 1654–1658. doi: 10.21437/Interspeech.2021-930.

[7] Zhisheng Zheng, Puyuan Peng, Anuj Diwan, Cong Phuoc Huynh, Xiaohang Sun, Zhu Liu, Vimal Bhat, and David Harwath. “VoiceCraft-X: Unifying Multilingual, Voice-Cloning Speech Synthesis and Speech Editing”. In: *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. 2025, pp. 2737–2756.

[8] Jee-weon Jung, Hee-Soo Heo, Hemlata Tak, Hye-jin Shim, Joon Son Chung, Bong-Jin Lee, Ha-Jin Yu, and Nicholas Evans. “AASIST: Audio Anti-Spoofing Using Integrated Spectro-Temporal Graph Attention Networks”. In: *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2022, pp. 6367–6371.

[9] 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: *Interspeech 2019*. 2019, pp. 1008–1012. doi: 10.21437/Interspeech.2019-2249.

TAIPEI TECH

[10] Junichi Yamagishi, Xin Wang, Massimiliano Todisco, Md Sahidullah, Jose Patino, Andreas Nautsch, Xuechen Liu, Kong Aik Lee, Tomi Kinnunen, Nicholas Evans, and Héctor Delgado. "ASVspoof 2021: Accelerating Progress in Spoofed and Deepfake Speech Detection". In: *2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*. 2021, pp. 47–54. doi: 10.21437/ASVSPOOF.2021-8.

[11] Jiangyan Yi, Ruibo Fu, Jianhua Tao, Shuai Nie, Haoxin Ma, Chenglong Wang, Tao Wang, Zhengkun Tian, Xiaohui Zhang, Ye Bai, Cunhang Fan, Shan Liang, Shiming Wang, Shuai Zhang, Xinrui Yan, Le Xu, Zhengqi Wen, Haizhou Li, Zheng Lian, and Bin Liu. "ADD 2022: the First Audio Deep Synthesis Detection Challenge". In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2022, pp. 9216–9220.

[12] Jiangyan Yi, Jianhua Tao, Ruibo Fu, Xinrui Yan, Chenglong Wang, Tao Wang, Chu Yuan Zhang, Xiaohui Zhang, Yan Zhao, Yong Ren, Le Xu, Junzuo Zhou, Hao Gu, Zhengqi Wen, Shan Liang, Zheng Lian, Shuai Nie, and Haizhou Li. "ADD 2023: The Second Audio Deepfake Detection Challenge". In: *IJCAI 2023 Workshop on Deepfake Audio Detection and Analysis (DADA 2023)*. Vol. 3597. CEUR Workshop Proceedings. CEUR-WS.org, 2023, pp. 125–130.

[13] Menglu Li, Yasaman Ahmadiadli, and Xiao-Ping Zhang. "A Survey on Speech Deepfake Detection". In: *ACM Computing Surveys* 57.7 (2025), pp. 1–38.

[14] Jiangyan Yi, Chenglong Wang, Jianhua Tao, Xiaohui Zhang, Chu Yuan Zhang, and Yan Zhao. "Audio Deepfake Detection: A Survey". In: *arXiv preprint arXiv:2308.14970* (2023).

[15] Sandipana Dowerah, Atharva Kulkarni, Ajinkya Kulkarni, Hoan My Tran, Joonas Kalda, Artem Fedorchenko, Benoit Fauve, Damien Lolive, Tanel Alumäe, and Matthew Magimai Doss. "Speech DF Arena: A Leaderboard for Speech Deepfake Detection Models". In: *IEEE Open Journal of Signal Processing* 7 (2026), pp. 73–81. doi: 10.1109/OJSP.2026.3652496.

[16] Xin Wang, Hector Delgado, Hemlata Tak, Jee weon Jung, Hye jin Shim, Massimiliano Todisco, Ivan Kukanov, Xuechen Liu, Md Sahidullah, Tomi Kinnunen, Nicholas Evans, Kong Aik Lee, and Junichi Yamagishi. "ASVspoof 5: Crowdsourced Speech Data, Deepfakes, and Adversarial Attacks at Scale". In: *The Automatic Speaker Verification Spoofing Countermeasures Workshop (ASVspoof 2024)*. 2024, pp. 1–8. doi: 10.21437/ASVspoof.2024-1.

[17] Puyuan Peng, Po-Yao Huang, Shang-Wen Li, Abdelrahman Mohamed, and David Harwath. "VoiceCraft: Zero-Shot Speech Editing and Text-to-Speech in the Wild". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2024, pp. 12442–12462. doi: 10.18653/v1/2024.acl-long.673.

[18] Helin Wang, Meng Yu, Jiarui Hai, Chen Chen, Yuchen Hu, Rilin Chen, Najim Dehak, and Dong Yu. “SSR-Speech: Towards Stable, Safe and Robust Zero-Shot Text-Based Speech Editing and Synthesis”. In: *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2025, pp. 1–5.

[19] Wen Huang, Yanmei Gu, Zhiming Wang, Huijia Zhu, and Yanmin Qian. “SpeechFake: A Large-Scale Multilingual Speech Deepfake Dataset Incorporating Cutting-Edge Generation Methods”. In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2025, pp. 9985–9998.

[20] Haibin Wu, Yuan Tseng, and Hung-yi Lee. “CodecFake: Enhancing Anti-Spoofing Models Against Deepfake Audios from Codec-Based Speech Synthesis Systems”. In: *Interspeech 2024*. 2024, pp. 1770–1774. doi: 10.21437/Interspeech.2024-2093.

[21] Sung-Feng Huang, Heng-Cheng Kuo, Zhehuai Chen, Xuesong Yang, Chao-Han Huck Yang, Yu Tsao, Yu-Chiang Frank Wang, Hung-yi Lee, and Szu-Wei Fu. “Detecting the Undetectable: Assessing the Efficacy of Current Spoof Detection Methods Against Seamless Speech Edits”. In: *2024 IEEE Spoken Language Technology Workshop (SLT)*. IEEE. 2024, pp. 652–659.

[22] Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, and Wei-Ning Hsu. “Voicebox: Text-Guided Multilingual Universal Speech Generation at Scale”. In: *Advances in Neural Information Processing Systems*. Vol. 36. 2023, pp. 14005–14034.

[23] Vahid Noroozi, Zhehuai Chen, Somshubra Majumdar, Steve Huang, Jagadeesh Balam, and Boris Ginsburg. “Instruction Data Generation and Unsupervised Adaptation for Speech Language Models”. In: *Interspeech 2024*. 2024, pp. 4049–4053. doi: 10.21437/Interspeech.2024-1575.

[24] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. “Mixtral of Experts”. In: *arXiv preprint arXiv:2401.04088* (2024).

[25] Samuele Cornell, Jordan Darefsky, Zhiyao Duan, and Shinji Watanabe. “Generating Data with Text-to-Speech and Large-Language Models for Conversational Speech Recognition”. In: *Synthetic Data’s Transformative Role in Foundational Speech Models*. 2024, pp. 6–10. doi: 10.21437/SynData4GenAI.2024-2.

[26] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh

TAIPEI TECH

Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik

TAIPEI TECH

Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. “The Llama 3 Herd of Models”. In: *arXiv preprint arXiv:2407.21783* (2024).

[27] Hieu-Thi Luong, Haoyang Li, Lin Zhang, Kong Aik Lee, and Eng Siong Chng. “LLaMA-PartialSpoof: An LLM-Driven Fake Speech Dataset Simulating Disinformation Generation”. In: *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2025, pp. 1–5.

[28] OpenAI, : Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela

TAIPEI TECH

Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian O'Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljubeh, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan Shah,

TAIPEI TECH

Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermani, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunninghman, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan, Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyi Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. “GPT-4o System Card”. In: *arXiv preprint arXiv:2410.21276* (2024).

[29] Yao Shi, Hui Bu, Xin Xu, Shaoji Zhang, and Ming Li. “AISHELL-3: A Multi-Speaker Mandarin TTS Corpus and the Baselines”. In: *Interspeech 2021*. 2021, pp. 2756–2760. doi: 10.21437/Interspeech.2021-755.

[30] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. “Robust Speech Recognition via Large-Scale Weak Supervision”. In: *International Conference on Machine Learning (ICML)*. PMLR. 2023, pp. 28492–28518.

[31] Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. “Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi”. In: *Interspeech*. Vol. 2017. 2017, pp. 498–502.

TAIPEI TECH

[32] Han He and Jinho D Choi. “The Stem Cell Hypothesis: Dilemma Behind Multi-Task Learning with Transformer Encoders”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2021, pp. 5555–5577.

[33] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. “Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone”. In: *arXiv preprint arXiv:2404.14219* 2.6 (2024), p. 4.

[34] Yue Zhang, Bo Zhang, Haochen Jiang, Zhenghua Li, Chen Li, Fei Huang, and Min Zhang. “NaSGEC: A Multi-Domain Chinese Grammatical Error Correction Dataset from Native Speaker Texts”. In: *Findings of ACL*. 2023.

TAIPEI TECH