Everyday Conversation Speech Recognition with End-to-End Neural Networks

Xuankai Chang

CMU-LTI-24-009

June 2024

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:

Shinji Watanabe, Chair
Bhiksha Ramakrishnan
Rita Singh
Naoyuki Kanda (Microsoft)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Language and Information Technology.

Keywords: Speech Recognition, Distant Speech Recognition, Overlapping Speech Recognition, End-to-End Speech Recogniton, Neural Networks

Acknowledgement

Completing this PhD thesis has been a remarkable journey, marked by growth, challenges, and invaluable support from many individuals. I am deeply grateful to all those who have contributed to this achievement.

First and foremost, I would like to express my heartfelt gratitude to my family, especially my father and mother, for their unwavering support, encouragement, and love throughout this journey. Your belief in me has been a constant source of strength and motivation.

I owe an immense debt of gratitude to my supervisor, Prof. Shinji Watanabe, for his exceptional guidance, insightful advice, and relentless support. Your expertise and mentorship have been instrumental in shaping my research and bringing this work to fruition.

My journey began at Shanghai Jiao Tong University (SJTU), and I am deeply thankful to Prof. Kai Yu and Prof. Yanmin Qian, for their foundational support and mentorship during the early stages of my academic career.

My heartfelt appreciation goes to my collaborators, Dong Yu, Yuya Fujita, Takashi Maekaku, and Yusuke Shinohara Takuya Yoshioka, Zhuo Chen, Gaur Yashesh, Xiaofei Wang, Zhong Meng Takaaki Hori, Niko Moritz, Jonathan Le Roux, Hakan Erdogan, Scott Wisdom, and John Hershey, Hung-yi Lee, Shang-Wen Li, Karen Livescu, Aswin Subramanian, Wangyou Zhang, Chenda Li, Xie Chen, Zhehuai Chen, Nanxin Chen, Hainan Xu, Zili Huang Jing Shi, Pengcheng Guo, Zhong-Qiu Wang, Soumi Maiti, Samuele Cornell, Jee-weon Jung, Hye-jin Shim, Xinjian Li, Roshan Sharma, Jiatong Shi, Brian Yan, Yifan Peng, Jessica Huynh, Muqiao Yang, Siddhant Arora, Yen-Ju Lu, Yoshiki Masuyama, Jinchuan Tian, William Chen, Shih-Lun Wu, Kwanghee Choi, Li-wei Chen, Minsu Kim, Yihan Wu, Dan Berrebbi, Peter Wu, Eason Lu, Zhaoheng Ni, Matthew Maciejewski, Matthew Wiesner, Tianzi Wang, Yiming Wang, Desh Raj, Dongji Gao, Ke Li, Yiwen Shao, Shu-wen Yang, Cong Han, Efthymios Tzinis. Your insights and expertise were crucial in expanding my research horizons and practical experience.

This thesis is a culmination of the collective support and contributions of many individuals, and I am profoundly grateful to each and every one of you. Thank you.

Abstract

Automatic speech recognition (ASR) is an essential technology which facilitates effective human-computer interaction. With the rapid progress in deep learning techniques, end-to-end (E2E) neural network-based ASR has brought significant advancements with remarkable performance. The success of ASR models have inspired various applications such as virtual assistants and automatic transcription services. Despite these achievements, recognizing conversational speech remains a challenging task, especially in the presence of environmental noise, room reverberations and speech overlaps.

This thesis aims to address the challenges of recognizing everyday conversation speech in ASR systems using E2E neural networks. The proposed research will explore techniques and methodologies to enhance the performance of ASR in challenging real-word conversational scenarios. We divide the problem into several sub-problems focusing on speech overlaps, noise, and reverberations, where each of these factors will be individually analyzed and addressed. In addition, we conduct diverse investigations on E2E neural network architectures to leverage the benefits of joint training to handle these challenges. Specifically, we build E2E ASR models by integrating ad-hoc modules, including speech enhancement, feature extraction and speech recognition.

We begin with the fundamental task of speech recognition using single-channel input containing a single speaker. Environmental noises and room reverberations significantly degrade speech recognition performance in such scenarios. To address this challenge, we propose a novel model architecture, integrating speech enhancement, self-supervised learning, and ASR models into a single neural network with an efficient training strategy. This integration has led to notable performance improvements, demonstrating the feasibility and effectiveness of employing end-to-end (E2E) neural networks for speech recognition with complex acoustic and linguistic properties. We then extend our approach to accept multi-channel speech input with a single speaker. Inspired by recent advancements in large speech foundation models, we expand the capabilities of a model trained on thousands of hours of single-channel speech data to handle multi-channel input. This extension significantly enhances performance, particularly evident in real meeting transcription data. Furthermore, we address the challenge of speech overlaps, an area that has been under-explored. Overlapping speech poses difficulties in accurately decoding and aligning individual utterances. To tackle this, we propose several end-to-end (E2E) models designed specifically to recognize overlapping speech within single-channel input. Finally, we turn our attention to multi-channel speech input with speech overlaps present in the signal. We introduce a model capable of processing multi-channel input from multiple speakers, leveraging spatial information for improved performance. We also integrates various approaches proposed earlier, further enhancing its effectiveness in challenging scenarios.

Contents

List of Figures

List of Tables

Notation

Nomenclature

$\hat{\mathbf{s}}$	Estimated clean source waveform $\in \mathbb{R}^T$
$\mathbf{n}$	noise waveform $\in \mathbb{R}^T$
$\mathbf{O}$	Feature embedding sequence $\in \mathbb{R}^{* \times D}$, where $*$ is for sequence length
$\mathbf{S}$	Multi-channel clean source waveform $\in \mathbb{R}^{C \times T}$
$\mathbf{s}$	Single-channel clean source waveform $\in \mathbb{R}^T$
$\mathbf{X}$	Multi-chanel input noisy waveform $\in \mathbb{R}^{C \times T}$
$\mathbf{x}$	Single-chanel input noisy waveform $\in \mathbb{R}^T$
$\mathbf{Y}$	Output transcription sequence, each token $y_l \in \mathcal{V}$
$\mathcal{V}$	Vocabulary or alphabet
C	Number of channels in the input signal $\in [1, 2, \cdots]$
K	Number of speakers in the speech $\in [1, 2, \cdots]$
T	Temporal length of signal

Chapter 1

Introduction

Human-spoken language represents one of the most natural and effective means of communication, serving as a convenient interface for interacting with machines. This has spurred significant interest in the development of Automatic Speech Recognition (ASR) systems, driven by the numerous benefits of accurate ASR technology. Firstly, ASR plays a critical role in transcribing and analyzing large volumes of spoken data, including telephone calls, meetings, interviews, and chats. This capability enables efficient data retrieval, analysis, and decision-making across various domains. Secondly, ASR enhances accessibility by providing an alternative mode of interaction for individuals who prefer spoken communication over written text. This empowers users with disabilities or those seeking hands-free interaction with technology. Thirdly, ASR facilitates intuitive and seamless interaction between humans and machines, allowing users to communicate with devices using natural spoken language rather than traditional input methods like keyboards or touchscreens. This natural interface enhances user experience and simplifies human-computer interaction.

As ASR technologies continue to advance, new applications and services are emerging in voice-controlled systems, virtual assistants, and hands-free applications across diverse domains. Improved ASR capabilities pave the way for more sophisticated voice-driven technologies that enhance productivity, accessibility, and convenience for users.

Despite its transformative potential, current Automatic Speech Recognition (ASR) systems face substantial challenges when applied to everyday conversational speech due to inherent variabilities. These include diverse speech patterns (pace, rhythm, intonation), speaker characteristics (accent, speaking style, vocal qualities), transcription complexities (context, fillers, disfluencies), and challenging acoustic conditions (background noise, reverberations, overlapping speech). Among these, the complexity of acoustic conditions poses a particularly formidable obstacle that can significantly impact ASR system performance.

Efforts have been focused on mitigating these challenges, including addressing environmental noise, room reverberations, and speech overlaps. Environmental noise degrades speech signal quality, leading to reduced recognition accuracy, while room reverberations introduce additional distortions. Moreover, the presence of overlapping speech presents a unique challenge, requiring ASR systems to accurately distinguish and transcribe multiple speakers' voices concurrently.

Successfully addressing these challenges is crucial for enhancing the accuracy and robustness of ASR in real-world conversational settings. Innovative approaches and advancements in signal processing, machine learning, and neural network architectures are key to overcoming these obstacles and improving ASR performance under diverse and challenging conditions.

This thesis aims to enhance the performance of ASR in everyday conversational scenarios where the intelligibility of speech signals may be significantly degraded by speech overlaps, environmental noise, and room reverberations. To achieve this goal, we propose to develop E2E neural network models specifically tailored to handle these complex acoustic environments.

1.1 Background

The development of Automatic Speech Recognition (ASR) models has undergone significant evolution over the years. Prior to the advent of deep learning, the dominant ASR models relied on Gaussian mixture models (GMMs) and hidden Markov models (HMMs) (Young, 1996). The foundational concepts of ASR, dating back to the work of Dr. Jelinek and colleagues at IBM over half a century ago, are rooted in statistical modeling involving three key components: 1) acoustic modeling (AM) to model the likelihood of input features: $P(\text{Acoustics}|\text{Phoneme})$, where Acoustics is the acoustic feature and Phoneme is the phoneme sequence; 2) lexicon modeling: $P(\text{Phoneme}|\text{Word})$; 3) language modeling for the word sequence: $P(\text{Word})$. The ASR process involves computing the most likely word sequence given the input acoustic feature, which can be expressed as:

$$\begin{aligned}\hat{\text{Word}} &= \operatorname*{argmax}_{\text{Word}} P(\text{Word}|\text{Acoustics}) && (1.1)\\ &= \operatorname*{argmax}_{\text{Word}} \sum_{\text{Phoneme}} \left(P(\text{Acoustics}|\text{Phoneme})P(\text{Phoneme}|\text{Word})P(\text{Word})\right), && (1.2)\end{aligned}$$

where HMMs were used for the AM to model $P(\text{Acoustics}|\text{Phoneme})$. The likelihood of acoustic features used in the HMM is provided by the GMM. With the rapid advancements in deep learning techniques, GMMs have been replaced by deep neural networks (DNN) for acoustic modeling in ASR (Hinton et al., 2012; Qian et al., 2016; Chiu et al., 2018). DNN-based AMs have demonstrated substantial improvements in recognition accuracy and efficiency compared to traditional approaches based on GMM-HMM, ushering in a new era of robust and data-driven speech recognition systems. This shift towards neural network-based ASR also intrigued interests in end-to-end (E2E) speech recognition, discarding the intermediate phoneme states and lexicon models.

E2E-ASR directly maps input acoustic features to transcription sequences, further simplifying the pipeline. Multiple E2E-ASR approaches have been proposed, including the connectionist temporal classification (CTC) (Graves et al., 2006), recurrent neural network Transducer (RNN-T) (Graves, 2012) and attention-based encoder-decoder (AED) (Chorowski et al., 2014).

While ASR, particularly End-to-End (E2E) models, has achieved significant success, enhancing robustness and generalization to handle complex acoustic conditions remains a key area of research interest (Kinoshita et al., 2013; Vincent et al., 2017a; Watanabe et al., 2020). Numerous approaches have been proposed to improve robustness against noise, such as data augmentation (Ko et al., 2017; Park et al., 2019; Zhang et al., 2020b; Cornell et al., 2023) techniques and the development of novel model architectures (Ochiai et al., 2017a; Heymann et al., 2019). Additionally, addressing overlapping speech signals has led to the proposal of specialized models tailored for this purpose (Seki et al., 2018; Kanda et al., 2020b,a). These ongoing research efforts aim to enhance ASR performance under challenging real-world conditions, ensuring more accurate and reliable speech recognition systems.

As is well known, deep neural networks are quite data-hungary. The success of prominent products from large technology companies relies heavily on training their ASR models with thousands of hours of speech data, underscoring the importance of large-scale training. In recent years, numerous large-scale speech models have emerged, categorized into two types: unsupervised training and supervised training. The primary distinction lies in the supervision signal used during training. Supervised training relies on labeled data for supervision, whereas unsupervised training leverages the inherent structure of the input data itself. Notable examples of unsupervised learning methods include Wav2Vec 2.0 (Wang et al., 2021b), HuBERT (Hsu et al., 2021b), WavLM (Chen et al., 2021b), and BEST-RQ (Chiu et al., 2022). On the other hand, prominent supervised learning methods include SpeechStew (Chan et al., 2021), Whisper (Radford et al., 2023), and OWSM (Peng et al., 2023). These advancements in both unsupervised and supervised training techniques have significantly contributed to the development of robust and effective speech foundation models, driving progress in ASR technology.

In this thesis, our goal is to advance the state of the art in recognizing everyday conversational speech by employing innovative techniques and models. We aim to address the challenges posed by complex acoustic environments, overlapping speech, and varying speaker characteristics. By leveraging the end-to-end models and cutting-edge techniques, we strive to enhance the robustness, accuracy, and generalization capabilities of automatic speech recognition systems.

1.2 Problem formulation

There have been numerous existing studies dedicated to addressing the challenges posed by complex acoustic environments in automatic speech recognition (ASR) (Du et al., 2016; Vincent et al., 2017b; Barker et al., 2018; Chen et al., 2018a). These environments encompass various factors such as speech overlaps, environmental noise, and room reverberations, which significantly impact the performance of ASR systems. Researchers have recognized the importance of improving ASR accuracy and robustness in real-world conversational scenarios, where speech signals often suffer from degradation due to these complex acoustic conditions.

One area of research focuses on tackling environmental noise (Du et al., 2016; Menne et al., 2016). Noise signals are common in everyday life. We can represent the waveform of a noisy signal in mathematical form, $\mathbf{x} = [x_1, x_2, \ldots, x_T] \in \mathbb{R}^T$, where T is the length of the signal. For each data point at a discrete time t observed by a microphone, it can be denoted as

$$x_t = s_t + n_t, \tag{1.3}$$

where $s_t \in \mathbb{R}$ and $n_t \in \mathbb{R}$ are clean speech source and noisy signals at time t, respectively. The recording device may contain more than one microphone. When we collect the signals with C microphones, the notation remains the same, except that the signal at each timestep t contains C data points: $\mathbf{x}_t, \mathbf{s}_t, \mathbf{n}_t \in \mathbb{R}^C$.

Room reverberations caused by sound reflections in enclosed spaces also pose a considerable challenge to ASR systems (Delcroix et al., 2015). These reverberations introduce additional distortions to the speech signals $\mathbf{s}_t$, further degrading recognition accuracy. In this case, a reverberant signal can be represented as

$$\mathbf{x}_t = \mathbf{h}_t * \mathbf{s}_t + \mathbf{n}_t, \tag{1.4}$$

where $*$ is a convolution operator. $\mathbf{h}_t$ is the room impulse response (RIR), corresponding to the propagation of speech caused by the reflections from surfaces in the room.

Another significant challenge in ASR is the presence of speech overlaps (Yu et al., 2017b,c; Kanda et al., 2020b), where multiple speakers' voices are simultaneously present in the recorded audio. If the number of speakers to be K, the noisy signals can be denoted as

$$\mathbf{x}_t = \sum_{k=1}^{K} \mathbf{h}_t^k * \mathbf{s}_t^k + \mathbf{n}_t, \tag{1.5}$$

where $\mathbf{s}_t^k$ is the clean speech of $k-$th speaker at time t. Recognizing and transcribing individual utterances in such scenarios is a challenging task, as the overlapping speech leads to a mixture of multiple speakers' voices.

To address the challenges of complex acoustic environments, end-to-end (E2E) neural networks have garnered significant attention in the ASR community. E2E models offer a holistic approach by directly mapping acoustic features to transcriptions, eliminating the need for intermediate processing stages. In E2E-ASR models, a single neural network directly maps the input speech signals to the target transcriptions. In this case, the task is to recognize the transcription sequences, $\{\mathbf{Y}^k, 1 \leq k \leq K\}$, for all speakers of interest. Given the speech signals $\mathbf{x}$, the E2E-ASR model learns to generate $\{\mathbf{Y}^1, \mathbf{Y}^2, \ldots, \mathbf{Y}^K\} = \text{E2E-ASR}(\mathbf{x})$. For each speaker k, $\mathbf{Y}^k = [y_1^k, y_2^k, \ldots, y_{L_k}^k]$ has length L_k, and $y_i^k \in \mathcal{V}$, where $\mathcal{V}$ represents the vocabulary.

1.3 Approach

It is challenging to attempt to solve all complex acoustic challenges simultaneously due to their inherent complexities. Therefore, in this thesis, we adopt a systematic approach by breaking down the overall problem into manageable sub-problems. Starting from conventional sub-problem for ASR, we aim to propose effective solutions that collectively contribute to the improvement of ASR performance in real-world conversational scenarios.

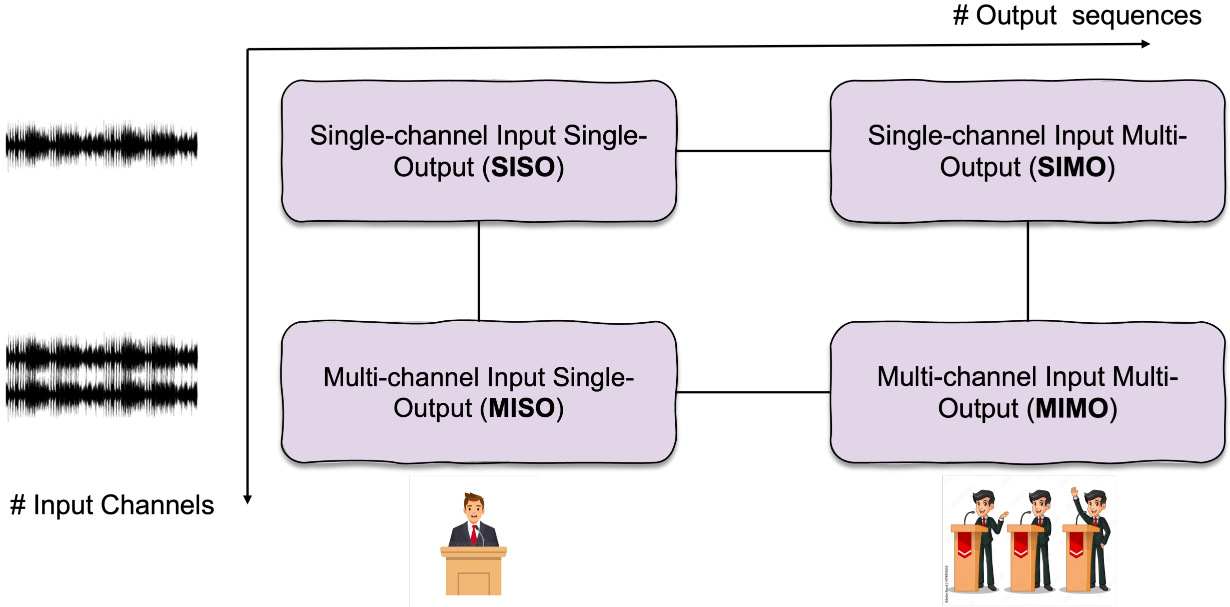

# Output sequences
Single-channel Input Single-Output (SISO)
Single-channel Input Multi-Output (SIMO)
Multi-channel Input Single-Output (MISO)
Multi-channel Input Multi-Output (MIMO)
# Input Channels

Figure 1.1: Illustration depicting the sub-problems in everyday conversation speech recognition, characterized by the number of input channels and output sequences.

We design sub-problems according to characteristics, namely the number of input channels and that of output sentences, shown in Fig. 1.1. Firstly, we consider the difference of single-channel and multi-channel scenarios. In the multi-channel case, the geometry difference between multiple microphones can provide spatial information, which is useful in speech denoising and separation. This spatial information can be leveraged to enhance the performance of ASR systems in challenging acoustic environments. Secondly, we consider the presence of speech overlaps, where multiple speakers' voices overlap in the recorded audio. Recognizing and transcribing individual utterances accurately in such scenarios is a significant challenge. To this end, we describe the sub-problems in the following.

Single-channel Input Single-speaker Output (SISO) The objective of SISO is to recognize the speech of a single speaker from a single-channel input, representing a common research focus. This scenario poses challenges due to environmental noise and room reverberations, which have been extensively studied. However, recent advancements in self-supervised learning (SSL), exemplified by Wav2Vec 2.0 (Baevski et al., 2020), HuBERT (Hsu et al., 2021b), and WavLM (Chen et al., 2021b), introduce a new paradigm and potential to elevate performance. By integrating speech enhancement, SSL, and ASR into a unified model, we achieved highly promising results.

Multi-channel Input Single-speaker Output (MISO) In the MISO sub-problem, the goal is to recognize the speech of a single speaker from a multi-channel input. While there is no overlapping speech involved, the availability of multiple channels offers spatial information that can be leveraged to enhance the performance of ASR systems. Similar to SISO, we explore the E2E model that incorporates the strong speech enhancement and SSL models. Through extensive experimentation, we aim to demonstrate the effectiveness of our proposed techniques for MISO-ASR.

Single-channel Input Multi-speaker Output (SIMO) In the SIMO case, the task is to recognize speech from multiple speakers using only a single-channel input. Unlike SISO, SIMO involves overlapping speech composed of homogeneous signals that may become misaligned during recognition. This task is notably challenging, demanding the separation and transcription of individual speakers' utterances from a blend of voices. The absence of spatial information in the single-channel input further complicates accurate distinction and decoding of overlapping speech. To tackle this challenge, we propose several new model architectures specifically designed for SIMO-ASR. These approaches aim to exploit the inherent characteristics of the overlapping speech and leverage advanced signal processing techniques to improve separation and recognition performance.

Multi-channel Input Multi-speaker Output (MIMO) In the MIMO scenario, the objective is to recognize overlapping speech from multiple speakers using a multi-channel input. Unlike the SIMO case, where only a single-channel input is available, the presence of multiple channels offers valuable spatial information that can facilitate speech separation and recognition. However, modeling presents challenges due to the complexity of multi-channel signals. Our proposed architecture comprises several modules designed to collectively address speech separation, feature extraction, and speech recognition tasks, all jointly trained from scratch to optimize performance.

1.4 Corpora

In order to verify the proposed methods, we conducted experiments using several diverse speech corpora. This section provides an overview of the major speech corpora employed throughout this thesis. These corpora are essential for training, validating, and testing the proposed models, ensuring their robustness and effectiveness across various acoustic environments and speech scenarios. Detailed information about each corpus is provided below, aligned with the divisions of the ASR tasks as described in the above sections.

1.4.1 Corpora for SISO-ASR

In the Single-channel Input Single-speaker Output (SISO) scenario, we utilized the CHiME-4 corpus (Vincent et al., 2017b), which was employed in the 4th Computational Hearing in Multisource Environments (CHiME) challenge. This dataset comprises both real and simulated noisy recordings of speech derived from the Wall Street Journal (WSJ0) corpus, at 16 kHz sampling rate. The recordings span four challenging noisy environments: bus, cafe, pedestrian area, and street. The original CHiME-4 corpus contains audio recordings captured by six microphones arranged in a specific configuration to simulate real-world multi-source environments. To evaluate the ASR performance in the single-channel case, we extracted each individual channel separately, treating each as a standalone input audio stream. This approach allowed us to focus on the SISO scenario, assessing the models' ability to handle noisy and reverberant conditions without the benefit of multi-channel information. There are 1,600 real and 7,138 simulated utterances for training, 1,640 real and 1,640 simulated utterances for development, and 1,320 real and 1,320 simulated utterances for test. The CHiME-4 corpus is crucial for testing the robustness of ASR systems in noisy environments. By using this dataset, we aimed to ensure that our proposed models could effectively mitigate the adverse effects of environmental noise and improve speech recognition accuracy under challenging conditions.

1.4.2 Corpora for MISO-ASR

In the Multi-channel Input Single-speaker Output (MISO) scenario, we conducted all experiments using real-world English meeting recordings from the AMI meeting corpus (Carletta, 2006). The AMI corpus is an extensive dataset that includes recordings captured by both close-talking and far-field microphones. The former is made using individual headset microphones (IHM) worn by each participant. While our focus is on the far-field scenario, where an 8-channel microphone array, commonly referred to as multiple distant microphones (MDM), was employed. Conventionally, the 1st channel of the MDM is selected to create an individual monaural condition known as a single distant microphone (SDM). The AMI corpus provides approximately 100 hours of meeting recordings, with human-annotated transcriptions. The AMI corpus also provides the segmentation information at the utterance level to construct individual training samples. The diverse acoustic conditions and the presence of spontaneous conversational speech in these recordings make the AMI corpus an ideal choice for evaluating the robustness of ASR systems in real-world settings. Despite the CHiME-4 data also containing multi-channel recordings, it has been extensively studied in our previous work and falls outside the scope of this thesis. The AMI corpus, with its focus on meeting scenarios and its rich annotation, offers a more suitable and challenging dataset for advancing our MISO ASR research.

1.4.3 Corpora for SIMO-ASR

To evaluate our proposed methods in the Single-channel Input Multi-speaker Output (SIMO) case, we used artificially generated single-channel multi-speaker mixed signals, where utterances from different speakers overlap. Three benchmark corpora commonly used in SIMO-related studies were employed:

1. **WSJ0-Mix.** This dataset is simulated based on the Wall Street Journal (WSJ) corpus developed by NIST, specifically using source utterances from the WSJ0 section[1]. WSJ0-Mix includes two primary categories: the 2-speaker scenario and the 3-speaker scenario. In the 2-speaker scenario, we use the common benchmark called WSJ0-2mix dataset introduced by (Hershey et al., 2016a) with a sampling rate of 16 KHz. The training and validation sets are generated by randomly selecting two utterances from different speakers from the WSJ0 si_tr_s partition, containing around 30 h and 10 h speech mixture, respectively. To mix the utterances, various signal-to-noise ratios (SNRs) are uniformly chosen from [0, 10] dB. For the test set, the mixture is similarly generated using utterances from the WSJ0 validation set si_dt_05 and evaluation set si_et_05, resulting in 5 h speech mixtures. For the 3-speaker case, similar methods are adopted except the number of speakers is three.

2. **WSJ-Mix.** Similar to the WSJ0-Mix corpus, WSJ-Mix is also derived from the Wall Street Journal (WSJ) speech corpus. Introduced by Seki et al. (Seki et al., 2018), this dataset uses the full WSJ corpus, which includes WSJ0 and WSJ1 . The generation process mirrors that of WSJ0-Mix, using the tool released by MERL[2], but with source utterances chosen from

[1] WSJ0 is also known as LDC93S6A

[2] http://www.merl.com/demos/deep-clustering/create-speaker-mixtures.zip

the full WSJ corpus (comprised of WSJ0 and WSJ1[3]). We used WSJ_SI284, to generate the training data, Dev93 for development and Eval92 for evaluation. The durations for the training, development and evaluation sets of the mixed data are 98.5 hr, 1.3 hr, and 0.8 hr respectively. Note that WSJ_SI284, Dev93 and Eval92 are the training, validation and evaluation sets for the WSJ, respectively.

3. **LibriMix.** Our methods are additionally tested on LibriMix, a recent open-source dataset for multi-speaker speech processing. The LibriMix data is created by mixing the source utterances randomly chosen from different speakers in LibriSpeech (Panayotov et al., 2015) and the noise samples from WHAM! (Wichern et al., 2019). The SNRs of the mixtures are normally distributed with a mean of 0 dB and a standard deviation of 4.1 dB. LibriMix is composed of 2-speaker or 3-speaker mixtures, with or without noise conditions. For fast evaluation, we conducted our experiments on the train-100 subset from Libri2Mix, which contains around 100 h of 2-speaker mixture speech.

By utilizing these diverse and challenging datasets, we ensured that our proposed SIMO-ASR methods were rigorously tested and validated across various overlapping speech scenarios, enhancing their robustness and applicability in real-world conditions. However, the major drawback of these corpora is that they are not real spontaneous conversational overlapping speech signals, which will be left in future studies.

1.4.4 Corpora for MIMO-ASR

To evaluate the effectiveness of our proposed end-to-end model for the Multi-channel Input Multi-speaker Output (MIMO) scenario, we conducted experiments on several benchmark datasets. These datasets were specifically designed for multi-speaker, multi-channel speech recognition and separation tasks. The detailed information about each corpus is provided below:

- **Spatialized WSJ0-2Mix.** According to the name, it is obvious that this corpus is an extension of the single-channel WSJ0-2Mix used in the SIMO case. The specialization process is described in (Wang et al., 2018), using a room impulse response (RIR) generator[4], where the characteristics of each two-speaker mixture are randomly generated including room dimensions, speaker locations, and microphone geometry[5]. Room impulse responses were simulated and convolved with dry source signals from WSJ0-2mix (Hershey et al., 2016a). The signal-to-distortion ratio (SDR) (Vincent et al., 2006) with respect to the input mixture

[3] WSJ1 is also known as LDC94S13A

[4] Available online at `https://github.com/ehabets/RIR-Generator`

[5] The spatialization toolkit is available at `http://www.merl.com/demos/deep-clustering/spatialize_wsj0-mix.zip`

is 0.07 dB in spatialized WSJ0-2mix. After spatializing, the training, validation, and test sets of both datasets contain 20,000, 5,000, and 3,000 mixtures, respectively.

- **Spatialized WSJ-2Mix.** Similar to the spatialized WSJ0-2Mix, we simulated the specialization process for the single-channel multi-speaker dataset, WSJ-2Mix as described in the SIMO case. We use the same random RIR generator and the specialization process as the WSJ0-2Mix.

- **WHAMR!.** WHAMR! (Maciejewski et al., 2020) is one of the most challenging datasets for speech separation, as it contains two-channel real-recorded environmental noise. For WHAMR!, the SDR with respect to the input mixture is -4.61 dB.

For all datasets, we used the 16 kHz version in our experiments.

1.5 Thesis Statement

In everyday conversational settings, automatic speech recognition (ASR) models face formidable challenges posed by degraded signal quality. Achieving precise recognition of such degraded speech signals is crucial for attaining human-level intelligence. Leveraging the efficiency and efficacy of end-to-end neural networks, we can markedly enhance ASR recognition accuracy, thus overcoming the hurdles inherent in conversational speech recognition.

1.6 Thesis Organization

We outline the structure of the thesis and provide an overview of the main topics covered in each chapter.

Chapter 1 provides a comprehensive overview of the thesis. In Chapter 2, we delve into the methodological background of the problem, laying a solid foundation for the approaches explored in this research. We present a detailed overview of the various approaches employed, discussing their theoretical underpinnings, implementation strategies, and the rationale behind their selection. Additionally, we highlight the general findings and insights gained from these methodologies, setting the stage for the in-depth discussions and analyses in the later chapters. This background is essential for understanding the innovative techniques and solutions proposed in this thesis.

Part I of this thesis focuses on Single-Channel Input, Single-Output (SISO) ASR models. While the SISO scenario does not involve speech overlaps, it presents challenges due to environmental noise and room reverberations. In Chapter 3, we propose innovative approaches leveraging self-supervised learning (SSL) to extract features efficiently, enhancing performance through speech enhancement processing. We also discuss effective training strategies to optimize model performance.

Transitioning to the Part II of the thesis, we extend the model to process multi-channel speech signals. Drawing inspiration from the success of large speech foundation models, Chapter 4 introduces a multi-channel extension of the foundation model. This extension demonstrates significant improvements in recognition performance using real meeting recordings from the AMI meeting corpus.

In the Part III, we focus on Single-Channel Input, Multiple-Output (SIMO) ASR models. Chapter 5 introduces End-to-End (E2E) ASR for multi-speaker overlapping speech, demonstrating feasibility and efficiency in recognizing overlapping speech with known speaker counts. Chapter 6 delves into conditional-chain models to handle varying speaker counts in speech overlaps, while Chapter 7 proposes a flexible model for transcribing partially overlapping speech encountered in real-world scenarios.

In the Part IV, we focus on the MIMO-ASR models, where the model handles multi-channel input speech signals with multi-speaker overlaps. Chapter 8 and 9 introduce a novel E2E model that incorporates a masking-based neural beamformer with multiple speech sources and a ASR model. The neural beamformer aims to enhance the target speaker's speech while suppressing interference from other speakers and background noise. The enhanced speech is then fed into an ASR model for transcription. The whole model is jointly trained from scratch solely on the ASR criterion. We also propose to use transformer as the backbone in the masking estimation network. To improve the computation efficiency and reduce the quadratic memory cost in self-attention, local self-attention is used. In Chapter 10, we incorporate some of the techniques mentioned in earlier sections, bolstering robustness in handling overlapping speech.

Finally, Chapter 11 summarizes the main conclusions of this thesis and proposes directions for future research in the area.

Chapter 2

Methodological Foundations and Design Principles for ASR Models

Summary

Conversational speech recognition aims to accurately transcribe spoken interactions in real-life scenarios, where conversations often occur in diverse and unpredictable acoustic environments. A crucial aspect is the ability to understand and process speech captured from a distance, commonly known as distant automatic speech recognition (DASR). Unlike traditional close-talking microphone setups, DASR deals with speech recorded by microphones placed at a distance from the speaker, which introduces significant challenges such as background noise, reverberation, and multiple overlapping voices.

Modern DASR models can generally be categorized into three different types:

- **Modular-based.** This approach involves building the model by concatenating several separate components, each tackling a sub-task towards the final target. A typical modular-based model consists of components for speech separation and enhancement (SE), feature extraction (FE), and speech recognition (ASR).

- **Non-modular-based.** Contrary to modular-based models, non-modular-based models are monolithic and directly recognize the input speech. These models usually consist of a large neural network that takes raw speech signals or features as input.

- **E2E modular-based.** E2E modular-based models share similar designs with modular-based models but are optimized jointly to reduce the discrepancy between different modules, aiming to achieve optimal performance.

All the methods proposed in this thesis fall into either the non-modular-based or E2E modular-based categories. Understanding these concepts is crucial for grasping the methodology and models discussed in the following sections. Therefore, the purpose of this chapter is to introduce these methodologies and provide the necessary background on these approaches.

> Xuankai Chang, Shinji Watanabe, Marc Delcroix, Tsubasa Ochiai, Wangyou Zhang, Yanmin Qian. "Modular-based End-to-End Distant Speech Processing: a case study of far-field ASR", submitted to Signal Processing Magzine 2024.

2.1 Introduction

Human listening abilities enable us to capture and understand various sources of information from complex sound scenes. This allows us, e.g., to follow a conversation at a cocktail party or notice various sounds occurring in our surroundings. The goal of speech and audio processing research has been to design technologies that could approach such human listening abilities. In this chapter, we focus on the example of distant automatic speech recognition (DASR) (Haeb-Umbach et al., 2021), which consists of transcribing the speech signals captured by microphones that are relatively far from the speakers so that it captures ambient noise, multiple speakers' voices, and room reverberation. DASR is one of the most important downstream applications in speech and audio signal processing (Haeb-Umbach et al., 2021).

The classical approach to tackle complex problems such as DASR has been to divide it into simpler sub-tasks, and design specific modules for each sub-task. We can then build a complex *modular system* by combining the individual "simple" modules. For example, the DASR problem usually builds on a model of the microphone signal consisting of speech corrupted by noise and interference speakers, etc. The speech signal carries the speech content uttered by the speaker, which can be modeled as a phoneme or word sequence. Consequently, the DASR problem can be divided into (1) a speech separation and enhancement (SSE) front-end that extracts speech from the interference speakers and noise, (2) a feature extraction (FE) module that generates an informative representation of the speech signal, and (3) an ASR back-end that converts the feature sequence into a word sequence. Each of the modules can be designed using either (1) a model-based approach, which leverages mathematical designs based on, e.g., physical consideration of the problem, (2) a data-driven approach such as deep learning, or (3) a combination of these approaches, which we call here mixed approach. Note that each of these modules can themselves also be composed of even more specific sub-modules.

There are several nice properties of such a modular approach. First, we can exploit information-theoretic models or physical knowledge about the sub-problems or from domain-specific data to design and optimize effective models independently for each sub-task. It is easy to interpret and evaluate each module because the output at each module has its own definition. Modular approaches are also advantageous in terms of flexibility. After deployment, each module can flexibly be modified without affecting other components of the system. An illustrative example of such modular design is showcased in (Haeb-Umbach et al., 2019), where it played a key role in developing a smart home assistant. At that time, modular systems were a better solution, given the limited exploration of end-to-end models. However, modular systems composed of many modules are complex and often cumbersome to optimize. Indeed, integrating multiple components, each with its own set of parameters and learning criteria, leads to discrepancy between them and suboptimal results (Yoshioka et al., 2015a).

Recently, progress in deep learning and the availability of a larger amount of data has allowed us to take an extremely data-driven approach, consisting of building a *non-modular system* to solve a complex task end-to-end (E2E) directly. A simple example consists of replacing the modular DASR system described above with a single neural network that accepts the microphone speech signal as input and outputs the transcription without any explicit SSE process. Such non-modular approaches have become increasingly popular because of their simple system design and possibility for E2E optimization, i.e., optimizing the whole system for the final task objective. However, such a black-box approach lacks the interpretability, possibility of introducing expert knowledge, and flexibility benefits of modular systems.

E2E Modular systems have emerged as a solution to combine the advantages of modular systems while allowing joint optimization of all modules. Such E2E modular systems are often realized by combining model-based and data-driven methods for SSE, FE, and ASR. E2E optimization is possible when each module of a modular system is *differentiable*. Therefore, we can represent the whole pipeline of a modular system with a single computational graph. The learnable parameters of the modules can then be optimized by back-propagation with a downstream objective, e.g., the ASR loss. Initial ideas for such DASR systems emerged in the early 2000s(Seltzer et al., 2004), but complexities and discrepancies between modules, such as the use of different optimization schemes, hindered practical implementation. However, the recent development of deep learning allowed us to re-formulate a modular DASR system as a unified neural network. This paradigm shift enabled joint optimization of the SSE front-end, FE, and ASR back-end (Narayanan and Wang, 2014; Chang et al., 2019c).

Problem formulation

This chapter discusses the design of modular, non-modular, and modular E2E systems by using the DASR problem as an example. The DASR consists of converting speech recorded at a microphone device into a word sequence for each active speaker. The microphone device can be an array composed of C microphones or a single microphone, i.e., $C = 1$. The microphone signal, denoted as $\mathbf{x}$, is based on an audio signal processing model and can be expressed as:

$$\mathbf{X} = \sum_{k=1}^{K} \mathbf{S}_k + \mathbf{N} \in \mathbb{R}^{C \times T}. \tag{2.1}$$

Here, $\mathbf{S}_k \in \mathbb{R}^{C \times T}$ represents the waveform of the speech signal of the k-th speaker, K is the number of active speakers, and $\mathbf{N} \in \mathbb{R}^{C \times T}$ denotes the background noise. The variable T represents the signal duration (number of samples). For simplicity in the discussions, we ignore the room reverberation in our notations.

Depending on the application, the goal of DASR can be to recognize a single speaker or all speakers talking in the recording. We introduce here the latter more general case. Let $\mathbf{Y} = \{Y^k\}_{k=1,\ldots,K}$, be the set of all transcriptions of the speakers in the recording, where $Y^k = [y_1^k, \ldots, y_{L_k}^k]$, is the sequence of tokens $y_l^k \in \mathcal{V}$ associated with the k-th speaker with a total sequence length of L_k. $\mathcal{V}$ represents the set of all possible tokens, which depending on the systems can be words, characters or other intermediate units. We can formalize the DASR problem as $\hat{\mathbf{Y}} = \mathrm{G}_\theta^{\mathrm{DASR}}(\mathbf{X})$, where $\mathrm{G}_\theta^{\mathrm{DASR}}(\cdot)$ represents the function of a DASR system with parameters θ, and $\hat{Y}$ is the set of predicted token sequences.

Table 2.1 shows a conceptual comparison of modular, non-modular and E2E modular systems, which we will use to guide our discussion. A modular system (first row of Table 2.1) can be expressed as the composition of modules' functions as $\mathrm{G}_\theta^{\mathrm{DASR}} = \mathrm{G}_\alpha^{\mathrm{ASR}} \circ \mathrm{G}_\beta^{\mathrm{FE}} \circ \mathrm{G}_\gamma^{\mathrm{SSE}}$, where $\mathrm{G}_\alpha^{\mathrm{ASR}}$, $\mathrm{G}_\beta^{\mathrm{FE}}$ and $\mathrm{G}_\gamma^{\mathrm{SSE}}$ represent the functions of the ASR, FE and SSE modules with parameters, α, β and γ, respectively. $\circ$ represents the function composition. Each of the modules can be designed using either a model-based approach, a data-driven approach such as deep learning, or a mixed approach. In contrast, a non-modular system (Second row of Table 2.1) uses a single module, such as a single neural network for $\mathrm{G}_\theta^{\mathrm{DASR}}$, and is optimized E2E. An E2E modular system (third row of Table 2.1) is modular, but its parameters are optimized E2E.

In this chapter, we illustrate our discussion with promising examples of approaches to design modular and non-modular ones. We emphasize the design of the SSE, FE, and ASR modules and how to optimize them jointly within an E2E modular system.

Although the discussion focuses on DASR, the combination of an SSE front-end with a back-end system and their joint optimization is also relevant to other problems. For example, replacing the ASR back-end with a speech translation or summarization module is a direction to realize meeting translation or summarization systems. Besides, acoustic event detection systems can include a sound separation front-end to allow better sound recognition(Turpault et al., 2020), which is a pipeline similar to modular/non-modular DASR systems in audio signal processing.

Table 2.1: Conceptual comparison of modular, non-modular and E2E modular schemes in terms functional representation, optimization problem and training data. $\mathcal{L}$ and D represent the training losses and training data for the different modules, respectively.

	Function	Optimization	Training data
Modular	$\mathrm{G}_{\alpha}^{\mathrm{ASR}} \circ \mathrm{G}_{\beta}^{\mathrm{FE}} \circ \mathrm{G}_{\gamma}^{\mathrm{SSE}}$	$\hat{\alpha} = \underset{\alpha}{\operatorname{argmin}} \sum_{\{\mathbf{Y},\mathbf{X}\} \in \mathcal{D}^{\mathrm{ASR}}} \mathcal{L}^{\mathrm{ASR}}(\mathbf{X}, \mathrm{G}_{\alpha}^{\mathrm{ASR}}(\mathbf{X}))$	Microphone signals $\mathbf{X}$
(Section 2.2)		$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{\{\mathbf{X}\} \in \mathcal{D}^{\mathrm{FE}}} \mathcal{L}^{\mathrm{FE}}(\mathbf{X}, \mathrm{G}_{\beta}^{\mathrm{FE}}(\mathbf{X}))$	Clean signals $\mathbf{S}$
		$\hat{\gamma} = \underset{\gamma}{\operatorname{argmin}} \sum_{\{\mathbf{S},\mathbf{X}\} \in \mathcal{D}^{\mathrm{SSE}}} \mathcal{L}^{\mathrm{SSE}}(\mathbf{S}, \mathrm{G}_{\gamma}^{\mathrm{SSE}}(\mathbf{X}))$	Transcriptions $\mathbf{Y}$
Non-modular (Section 2.3)	$\mathrm{G}_{\theta}^{\mathrm{DASR}}$	$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{\{\mathbf{Y},\mathbf{X}\} \in \mathcal{D}^{\mathrm{DASR}}} \mathcal{L}^{\mathrm{DASR}}(\mathbf{Y}, \mathrm{G}_{\theta}^{\mathrm{DASR}}(\mathbf{X}))$	Microphone signals $\mathbf{X}$ Transcriptions $\mathbf{Y}$
E2E Modular (Section 2.4)	$\mathrm{G}_{\alpha}^{\mathrm{ASR}} \circ \mathrm{G}_{\beta}^{\mathrm{FE}} \circ \mathrm{G}_{\gamma}^{\mathrm{SSE}}$	$\hat{\theta} = \underset{\theta=\{\alpha,\beta,\gamma\}}{\operatorname{argmin}} \sum_{\{\mathbf{Y},\mathbf{X}\} \in \mathcal{D}^{\mathrm{DASR}}} \mathcal{L}^{\mathrm{ASR}}(\mathbf{Y}, (\mathrm{G}_{\alpha}^{\mathrm{ASR}} \circ \mathrm{G}_{\beta}^{\mathrm{FE}} \circ \mathrm{G}_{\gamma}^{\mathrm{SSE}})(\mathbf{X}))$	Microphone signals $\mathbf{X}$ Transcriptions $\mathbf{Y}$

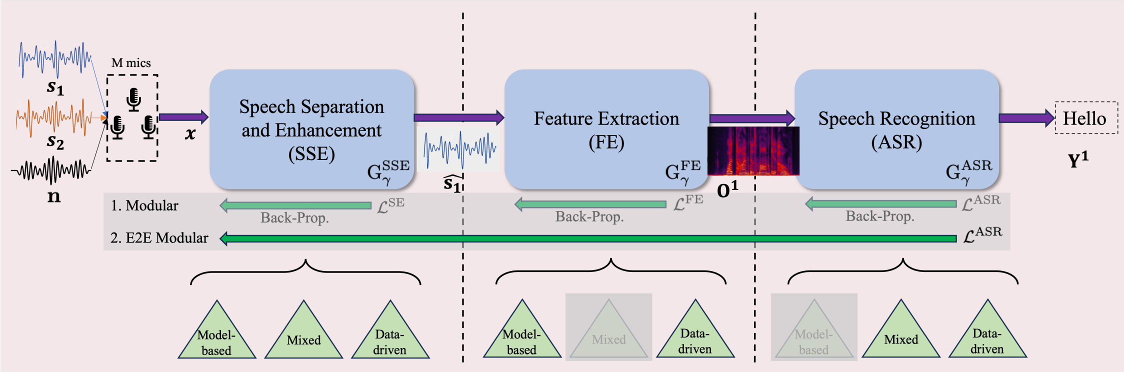

s_1
s_2
n
M mics
x
Speech Separation and Enhancement (SSE)
$\mathrm{G}_{\gamma}^{\mathrm{SSE}}$
$\widehat{s_1}$
Feature Extraction (FE)
$\mathrm{G}_{\gamma}^{\mathrm{FE}}$
$\mathbf{O}^1$
Speech Recognition (ASR)
$\mathrm{G}_{\gamma}^{\mathrm{ASR}}$
Hello
$\mathbf{Y}^1$
1. Modular
Back-Prop.
$\mathcal{L}^{\mathrm{SE}}$
Back-Prop.
$\mathcal{L}^{\mathrm{FE}}$
Back-Prop.
$\mathcal{L}^{\mathrm{ASR}}$
2. E2E Modular
$\mathcal{L}^{\mathrm{ASR}}$
Model-based
Mixed
Data-driven
Model-based
Mixed
Data-driven
Model-based
Mixed
Data-driven

Figure 2.1: The pipeline of the modular-based distant ASR systems. It consists of three components: SSE, FE and ASR. Each component is configurable with various methods. In an end-to-end modular-based system, the final ASR loss can backpropagate through all modules when the whole pipeline is differentiable. In this illustration, the output is presented for a single speaker; however, the SSE module can generate outputs for multiple speakers, with the same FE and ASR processes applied accordingly.

2.2 Modular-based distant ASR with Model-based and Data-driven approaches

We first introduce the modular system, which is based on the expert knowledge that DASR could be decomposed into individual components with different functions. It consists of a cascade of modules that can be designed independently, allowing to use loss functions and training data specific to each problem. For example, as shown in the middle of Table 2.1 and in Fig 2.1, we can build a DASR system by combining SSE, FE, and ASR modules independently optimized as follows,

$$\hat{\gamma} = \underset{\gamma}{\operatorname{argmin}} \sum_{\{\mathbf{S},\mathbf{X}\} \in \mathcal{D}^{\mathrm{SSE}}} \mathcal{L}^{\mathrm{SSE}}(\mathbf{S}, \mathrm{G}_{\gamma}^{\mathrm{SSE}}(\mathbf{X})), \tag{2.2}$$

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{\{\mathbf{X}\} \in \mathcal{D}^{\mathrm{FE}}} \mathcal{L}^{\mathrm{FE}}(\mathbf{X}, \mathrm{G}_{\beta}^{\mathrm{FE}}(\mathbf{X})), \tag{2.3}$$

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmin}} \sum_{\{\mathbf{Y},\mathbf{X}\} \in \mathcal{D}^{\mathrm{ASR}}} \mathcal{L}^{\mathrm{ASR}}(\mathbf{Y}, \mathrm{G}_{\alpha}^{\mathrm{ASR}}(\mathbf{X})). \tag{2.4}$$

Eqs (2.2)-(2.4) emphasizes that different datasets $\mathcal{D}^{\mathrm{SSE}}$, $\mathcal{D}^{\mathrm{FE}}$ and $\mathcal{D}^{\mathrm{ASR}}$, and also different losses, $\mathcal{L}^{\mathrm{SSE}}$, $\mathcal{L}^{\mathrm{FE}}$ and $\mathcal{L}^{\mathrm{ASR}}$ are used to design the SSE, FE and ASR modules. Here, S represents the clean speech reference used to optimize the SSE module. Note that the loss computation may include a transformation of the reference, such as applying the short-time Fourier transform (STFT) for SSE losses in the spectral domain or performing clustering for self-supervised learning (SSL)-based FE. Here, some model-based approaches (e.g., simple feature extraction such as FBank) do not involve an optimization problem as shown in Eqs (2.2)-(2.4). Other model-based approaches involve optimization of the parameter of a physical model, but the optimization is performed at the inference stage only using the actual observation, i.e., $\mathcal{D} = \{\mathbf{X}\}$, instead of at the training stage using training data resources.

Clearly, with such a modular approach, the output of each module is well-defined, making the whole system more interpretable and controllable than non-modular approaches introduced in Section 2.3.

We present here some representative approaches for the SSE, FE, and ASR modules, which can be combined to form a modular DASR system. We can arbitrarily modify the combination of modules to realize a DASR system with the desired properties, e.g., performing DASR of single or multiple speakers, using single or multiple microphones, etc.

2.2.1 Speech separation and enhancement front-end

The purpose of the SSE front-end is to estimate clean speech signals free of acoustic interferences from the observed microphone signals $\mathbf{X}$. There are two factors to consider when designing an SSE front-end: (1) the type of acoustic interference and (2) the availability of a microphone array.

When only one speaker is speaking, there are no interference speakers, and the signal model in Eq. (2.1) can be rewritten as $\mathbf{x} = \mathbf{s} + \mathbf{n} \in \mathbb{R}^T$. Thus, the problem reduces to noise reduction and eventually dereverberation. In this paper, we do not consider the dereverberation problem as it has been explained in a previous article(Yoshioka et al., 2012). The objective of SSE is to estimate the clean speech as $\hat{\mathbf{s}} = \mathrm{G}_{\gamma}^{\text{Denoising}}(\mathbf{x})$, where $\mathrm{G}_{\gamma}^{\text{Denoising}}$ represents the denoising function.

When multiple speakers are speaking, we need to perform speech separation to isolate the voices of the different speakers. The output consists of the speech signals of all active speakers as $\{\hat{\mathbf{s}}_k\}_{k=1,\dots,K} = \mathrm{G}_{\gamma}^{\text{Separation}}(\mathbf{x})$, where $\mathrm{G}_{\gamma}^{\text{Separation}}$ represents the separation function.

Another aspect to consider is whether the recordings are performed with a single microphone ($C = 1$) or a microphone array ($C \geq 2$). When using a microphone array, we can further leverage the benefit of the model-based approaches via multi-channel processing approaches such as beamforming, which exploit spatial information, leading to improved enhancement performance and fewer processing distortions.

Below, we provide representative examples for single- and multi-channel denoising and separation by categorizing them into three types: 1) model-based, 2) data-driven, and 3) mixed approaches. In general, distant microphone recordings may contain background noise and interfering speakers. Consequently, we can create an SSE module by combining several sub-modules to handle the desired recording conditions, e.g., $\mathrm{G}^{\text{SSE}} = \mathrm{G}^{\text{Separation}} \circ \mathrm{G}^{\text{Denoising}}$.

Model-based approaches We define the model-based SSE approach as $\mathrm{G}_{\gamma}^{\text{SSE}}$ where the function G^{SSE} is based on some physical model and the parameter γ is not learned on a training dataset in an E2E manner (but can be adaptively estimated for each input sample). Most conventional signal processing approaches belong to this category. Here, we only introduce several commonly-used model-based approaches due to the space limitation.

• **Single-channel denoising**: In single-microphone conditions, spectral subtraction is one of the first denoising methods in the literature. It operates in the frequency domain by converting the input speech $\mathbf{x}$ into a complex-valued spectrum $\mathbf{X} \in \mathbb{C}^{T \times F}$ via short-time Fourier transformer (STFT). The core idea is to estimate the corresponding noise spectrum and subtract it from the noisy speech spectrum to obtain the estimated clean spectrum. Popular noise estimation algorithms involve utilizing the minimum statistics such as the improved minimal controlled recursive averaging (IMCRA) algorithm (Cohen, 2003), which conducts rough voice activity detection (VAD) followed by recursive update of the estimated noise spectrum.

- **Multichannel denoising**: In multi-microphone conditions, the spatial information between different microphones can be utilized to achieve SSE. Beamforming (Van Veen and Buckley, 1988), one of the mostly commonly used approaches, operates on the physical model, treating multi-channel signals as time-delayed versions of the same source with attenuation. With known direction-of-arrival (DOA) of the target speech and microphone array geometry, a relative transfer function (RTF) vector is constructed, reflecting relative time delays at each microphone for a desired directional response. Such a response can maximize the signal gain in the desired direction. The fixed beamforming filter can then be derived by approximating the desired directional response. Note that the fixed beamforming filter always remains the same when processing different signals. In contrast, adaptive beamforming approaches dynamically estimate the corresponding filter for each input signal. The minimum variance distortionless response (MVDR) beamforming, for example, designs its filter by solving a constrained optimization problem which minimizes the energy of the filtered noise while keeping the desired signal intact. This leads to an adaptive filter that is estimated based on the input signal. This adaptive beamforming approach offers benefits such as low distortions in enhanced speech and compatibility with downstream tasks like ASR in real-world applications.

- **Multi-talker separation**: In addition, many works have been focusing on tackling speech separation in multi-speaker scenarios. When multiple microphones are available, blind source separation (BSS) techniques (Choi et al., 2005) have been developed to iteratively estimate the optimal unmixing matrix that can separate signals from different speakers via linear filtering. With multiple microphones, blind source separation (BSS) techniques (Choi et al., 2005) iteratively estimate an optimal unmixing matrix through linear filtering to separate signals from different speakers. BSS methods assume certain conditions, like the non-Gaussianity of source speech in independent component analysis (ICA). Unlike BSS approaches, which prefer more microphones to achieve high performance, computational auditory scene analysis (CASA) (Wang and Brown, 2006) takes inspiration from the human auditory system to build monaural or binaural SSE models with well-designed modules. Time-frequency masking is one of the most well-known CASA approaches that groups time-frequency bins in the noisy speech spectrum according to certain cues (e.g., sound location, pitch, spectral features, etc.). While model-based approaches are well-formulated on some theoretical bases, they usually face limitations due to explicit assumptions that may not hold in realistic conditions, leading to drastic performance degradation. Meanwhile, these approaches often do not fully exploit the information in the collected data since the parameters are mostly derived in a handcrafted manner (either as a closed-form solution or solved iteratively) based on the actual observed data. Given the fact that a large amount of data can be collected or simulated to cover a wide range of conditions, it is often favorable to take full advantage of these data to build capable and robust SSE systems.

Data-driven approaches With the rapid development of deep learning, data-driven approaches have gained attention for their strong capability in learning from data. To keep consistency with previous methods defined in Table 2.1, we define data-driven approaches as $\mathrm{G}_{\gamma}^{\mathrm{SSE}}$ where G is designed fully based on deep neural networks (DNN) and the parameter γ is learnable from data. As seen in Eq. (2.2), training or tuning an SSE module usually requires access to the microphone signal $\mathbf{X}$ and the corresponding clean speech $\mathbf{S}$. Recording simultaneous clean and noisy signals is challenging, leading to the use of simulated data where mixtures are artificially generated from isolated clean speech and noise signals, following the signal model in Eq. (2.1). In the past decade, data-driven SSE approaches have advanced greatly, surpassing traditional model-based approaches in most benchmarks.

• **Model**: Data-driven SSE approaches follow a common design paradigm with three main components: an encoder, a predictor, and a decoder. The encoder transforms the input speech into a feature $\mathbf{H} = \mathrm{Encoder}(\mathbf{x})$. The predictor generates representations for each speaker k and falls into two categories: mapping-based and masking-based. In mapping-based, the representation is the enhanced feature $\hat{\mathbf{H}}_k$, while in masking-based, it's a mask $\hat{\mathbf{M}}_k$ used for element-wise multiplication to obtain the enhanced feature $\hat{\mathbf{H}}_k = \hat{\mathbf{M}}_k \odot \mathbf{H}$. Note that this differs from the aforementioned CASA's time-frequency masking in that the mask estimation is purely based on the neural network instead of auditory cues. The enhanced feature is finally converted to the corresponding waveform $\hat{\mathbf{s}}_k = \mathrm{Decoder}(\hat{\mathbf{H}}_k)$. If the encoder and decoder are based on some frequency-domain transform (e.g., STFT) and its inverse transform, it is called a frequency-domain approach (Wang and Chen, 2018a). Otherwise, it is called a time-domain approach (Luo and Mesgarani, 2019a), where the encoder and decoder are learnable neural networks. For frequency-domain approaches, the loss function in Eq. (2.2) can be computed based on different output levels (mask, spectrum, and waveform):

$$\mathcal{L}^{\mathrm{SSE}}(\cdot,\cdot) \subseteq \left\{\mathcal{L}^{\mathrm{mask}}(\mathbf{M}_k, \hat{\mathbf{M}}_k), \mathcal{L}^{\mathrm{spectrum}}(\mathbf{X}_k, \hat{\mathbf{X}}_k), \mathcal{L}^{\mathrm{waveform}}(\mathbf{s}_k, \hat{\mathbf{s}}_k)\right\}. \tag{2.5}$$

Note that the mask-based loss, $\mathcal{L}^{\mathrm{mask}}$, is only used in models with masking-based predictors. For all losses, the L1 or L2 distance is usually used. For the waveform-based loss, $\mathcal{L}^{\mathrm{waveform}}$, some metric-based loss functions can be alternatively used, e.g., scale-invariant signal-to-noise ratio (SNR) (Le Roux et al., 2019a). For time-domain approaches, the model is often only trained with $\mathcal{L}^{\mathrm{waveform}}$, and sometimes also with $\mathcal{L}^{\mathrm{spectrum}}$.

• **Handling multi-talker situations**: The previously mentioned approaches are applicable to both single- and multi-speaker scenarios. However, in multi-speaker situations, there exists a permutation problem, leading to $K!$ possible ways to assign the order of the clean speech $\mathbf{s}_k$ to the corresponding separation output $\hat{\mathbf{s}}_{k'}$. This challenge has been addressed through two prominent frameworks in speech separation: deep clustering (DC) (Hershey et al., 2016b) and Permutation

Invariant Training (PIT) (Kolbæk et al., 2017). DC implicitly tackles the permutation problem by formulating the training procedure as clustering of time-frequency representations from different speakers. It involves projecting each time-frequency (T-F) bin into a high-dimensional embedding. The training objective, denoted as $\mathcal{L}^{\mathrm{SSE}} := \mathcal{L}^{\mathrm{DC}}$, aims to bring T-F embeddings corresponding to the same speaker cluster together while keeping them far apart otherwise. In contrast, PIT explicitly addresses the permutation problem by enumerating all possible permutations and consistently selecting the optimal permutation $\hat{\pi}$ for model training. PIT has gained popularity in the speech separation community due to its flexibility in model design and its explicit consideration of the permutation problem.

Note that all approaches above can be used for both single- and multi-channel processing by simply configuring the encoder to take different channels as input.

Mixed approaches Mixed approaches combine model-based and data-driven techniques. The SSE function, $\mathrm{G}^{\mathrm{SSE}}_{\gamma}$, is designed using a physical model as discussed in the above discussion on model-based SSE approaches but has parameters γ, which are optimized using a large amount of data. The physical model constrains the solution, which can help obtain more robust solutions than purely data-driven approaches. The use of a large amount of data allows improved SSE models with optimized parameters γ, which are hard to be explored with purely model-based approaches.

Mixed approaches have been extensively investigated with microphone array processing, where numerous relatively simple yet powerful and principled physical models exist. One example of such a mixed approach is the mask-based beamformer for multi-channel noise reduction(Haeb-Umbach et al., 2021). It exploits a neural network to compute time-frequency masks, as introduced in the masking-based approaches. These masks are then used to estimate the spatial covariance matrices of the speech and noise, which are necessary to compute the spatial filters of the beamformer. The final enhancement is performed with linear spatial filtering. We can build on the strong theoretical foundation of beamforming theory to design enhancement systems with desired physical properties, such as the distortionless constraint of MVDR. Moreover, we can exploit a large amount of data to learn a powerful mask estimator neural network, which can provide reliable estimates of the spatial covariance matrices.

Another line of research casts SSE as an analysis-resynthesis approach with the integration of deep neural networks. For example, (Jiang and Yu, 2023) takes inspiration from the conventional source-filter model in speech modeling and re-synthesizes the clean speech from the excitation and vocal tract components estimated from the single-channel noisy speech via neural networks.

We provided examples of mixed approaches for single- and multi-channel noise reduction, but similar ideas have also been applied to speech separation (Yoshioka et al., 2018a). Mixed approaches have been very successful and are often used in the development of SSE systems dealing with challenging recordings (Watanabe et al., 2020).

2.2.2 Feature extraction

FE plays a pivotal role in numerous signal processing and machine learning tasks, including ASR. Directly utilizing raw audio waveforms in ASR models would necessitate learning intricate patterns and representations from scratch, incurring computational expenses and potentially compromising effectiveness. Through FE, we map speech signals, $\mathbf{x}$, into a more suitable representation, $\mathbf{O}$, aligning it with the inherent structure of speech: $\mathbf{O} = \mathrm{G}_{\beta}^{\mathrm{FE}}(\mathbf{x})$. This transformation enhances the efficiency and accuracy of processing within ASR systems. Note that in this and the next section about ASR, we omit the speaker index k for simplicity without loss of generality.

Similar to the SSE component detailed in Section 2.2.1, we can categorize FE methods into model-based and data-driven classes based on the physical model and the parameter β.

Model-based approaches Model-based methods are traditionally dominant in FE, employing well-established techniques like Mel-Frequency Cepstral Coefficients (MFCC), Filterbank (FBank), and Perceptual Linear Prediction (PLP). These methods operate on the premise of predefined signal processing steps, emphasizing a structured approach to feature extraction. Historically, these model-based approaches have provided robust representations for speech signals, contributing significantly to the success of ASR systems. However, their effectiveness often relies on domain expertise and assumptions about the underlying characteristics of the speech data. Usually, the hyperparameters used in models are determined empirically by expert knowledge and tuned based on performance on the target task.

Data-driven approaches While model-based approaches have been foundational in speech processing, the surge of data-driven methods has gained prominence in the FE process, harnessing the capabilities of deep learning and SSL techniques. Instead of depending on handcrafted features, these methods directly learn feature representations from raw audio data through downstream or pretext tasks and optimized as shown in Eq. (2.3). In such cases, the parameter β of the FE component corresponds to partial or whole parameters of the deep neural network. $\mathcal{L}^{\mathrm{FE}}$ is the loss function, which also includes manipulating $\mathbf{y}$ to get the reference signal. Earlier studies attempted to directly learn hidden representations for speech recognition, exemplified by bottleneck features(Hermansky et al., 2000). More recently, SSL approaches (Mohamed et al., 2022a), including contrastive learning, autoencoders, and masked prediction, have showcased success in extracting meaningful representations from speech signals. The features derived from data-driven approaches can uncover patterns not readily apparent in handcrafted feature engineering. Additionally, they exhibit robustness to variations in real environments, such as changes in speaker or acoustic conditions. Consequently, data-driven approaches often demonstrate superior generalization capabilities.

Substituting the straightforward model-based FE with data-driven approaches yields enhanced performance. Nonetheless, these advantages are accompanied by the downside of employing significantly larger models, incurring computational and memory expenses. This is made possible due to the progress in computing resources and deep learning techniques, allowing the utilization of extensive amounts of labeled or unlabeled data and intricate model architectures.

2.2.3 Automatic speech recognition

The final module is the ASR, mapping acoustic features of enhanced speech signals to transcriptions, denoted as $\hat{\mathbf{Y}} = \mathrm{G}_{\alpha}^{\mathrm{ASR}}(\mathbf{O})$, with the parameter α learned through Eq. (2.4). The foundations of modern large vocabulary continuous speech recognition were laid by Dr. Jelinik and his colleagues at IBM about half a century ago. ASR can be recognized as a statistical process determined by three models: 1) acoustic modeling (AM) to model the likelihood of input features: $P(\mathbf{O}|V)$, where V is the phoneme sequence; 2) lexicon modeling: $P(V|\mathbf{Y})$; 3) language modeling for the word sequence $P(\mathbf{Y})$. The final prediction can be expressed as the following computation:

$$\hat{\mathbf{Y}} = \underset{\mathbf{Y}}{\operatorname{argmax}}\, P(\mathbf{Y}|\mathbf{O}) = \underset{\mathbf{Y}}{\operatorname{argmax}} \sum_{V} P(\mathbf{O}|V)P(V|\mathbf{Y})P(\mathbf{Y}). \tag{2.6}$$

Stemming from this, Gaussian Mixture Model (GMM)-Hidden Markov Model (HMM) and Deep Neural Network (DNN)-Hidden Markov Model (HMM) (Hinton et al., 2012) were very successful, except that multiple models are required. To simplify, recent years have seen the rise of E2E models (Li et al., 2022), including Connectionist Temporal Classification (CTC), Recurrent Neural Network Transducer (RNN-T), and attention-based encoder-decoder. These methods fall into two main categories: data-driven approaches and mixed approaches combining elements of both model-based and data-driven techniques.

Mixed approaches **GMM-HMM and DNN-HMM** follow the same paradigm in Eq. (2.6). They exemplify the mixed approach. Specifically, HMM is integrated into the AM to establish sequence alignment, making them adept at capturing temporal dynamics in speech signals. When training a DNN-HMM, obtaining a fixed pronunciation alignment between the input and target is challenging. To address this, **CTC** was proposed, incorporating a temporal modeling component to handle varying input and output sequences. This is achieved by considering all possible alignment sequences, $\mathbf{a} \in \mathcal{A}$, where each $\mathbf{a}$ is an expansion of the target sequence to match the length of the input. It computes sequence probability as $P(\mathbf{Y}|\mathbf{O}) = \sum_{\mathbf{a}\in\mathcal{A}} P(\mathbf{a}|\mathbf{O})$. **RNN-T**, introduced as an enhanced ASR framework compared to CTC, includes two additional sub-networks: a joiner and a predictor. The joiner integrates encoded acoustic features and the predictor's output to generate new tokens autoregressively. In these approaches, it is assumed that the output sequence is monotonically aligned with input.

All these methods in *mixed approaches* apply specific physical characteristics, such as statistical or temporal modeling, with data-driven elements. This hybrid nature allows them to benefit from the structured representations of traditional models while leveraging the expressive power of neural networks.

Data-driven approaches Within the domain of data-driven approaches, a notable and extensively explored avenue is represented by E2E-ASR models featuring attention-based encoder-decoder. These models aim to streamline the ASR process by directly mapping input audio sequences to transcriptions. The architecture is conceptually divided into three core components: the encoder, the decoder, and the attention module. The encoder captures high-level features from the input audio sequences, transforming raw acoustic information into a meaningful representation. The decoder generates textual output based on the encoded features, mapping it into a coherent and accurate textual representation. The attention module models the alignment between encoder outputs and decoder outputs, allowing dynamic focus on specific segments during decoding, effectively adapting to varying temporal complexities.

This section provided an overview of the fundamental modules in DASR, including SSE, FE and ASR, as depicted in Fig. 2.1. These modules represent essential stages in the DASR pipeline, addressing tasks such as signal cleaning, feature extraction, and transcription. While additional modules, such as speaker diarization, can be integrated into the pipeline, their detailed exploration is deferred here due to space constraints. It is noteworthy that similar counterparts such as modular-based systems can be identified in other fields like audio processing.

2.3 Data-driven based distant ASR - Non-modular models

While the modular-based system introduced in the previous section is successful in many applications, it requires expert knowledge to design and optimize each module. In comparison, another line of research focuses on non-modular DASR approaches that are generally data-driven and only require limited expert knowledge. As shown in the middle row of Table 2.1, a non-modular DASR system can be formulated as a single function $\mathrm{G}_{\theta}^{\mathrm{DASR}}$ such as a DNN. Such a system is usually optimized in an E2E manner, i.e., the parameters θ are updated to minimize the total ASR loss on all training data:

$$\hat{\theta} = \underset{\theta}{\mathrm{argmin}} \sum_{\{\mathbf{Y},\mathbf{X}\}\in\mathcal{D}^{\mathrm{ASR}}} \mathcal{L}^{\mathrm{DASR}}\left(\mathbf{Y}, \mathrm{G}_{\theta}^{\mathrm{DASR}}(\mathbf{X})\right) , \tag{2.7}$$

where $\hat{\theta}$ is the optimized parameter, $\mathcal{L}^{\text{DASR}}$ is a loss function specifically designed for the non-modular DASR system, and $\mathcal{D}^{\text{ASR}}$ defines the training data of the DASR system as in Table 2.1. The core idea is to integrate the functions of different modules into a single DASR model so that it can directly handle acoustic distortions captured at the microphone while performing speech recognition. It is thus based on the design of conventional clean-speech ASR systems, but modifies the training data, network structure or/and training loss to fit the DASR problem. We will discuss some representative examples below.

Note that we have not regarded simple FE (e.g. FBank) as a separate module, following the usual convention. FE is a widely used technique applied in various tasks across different domains, including ASR. It is often assumed to be part of the standard processing pipeline. We only explicitly consider FE as a separate module when it involves non-pure model-based methods.

2.3.1 Representative example of non-modular DASR systems

First, we focus on the single-speaker DASR sub-task, assuming only one speaker per sample. The non-modular DASR system in this scenario usually adopts the same architectures and loss functions of standard ASR models, e.g., CTC, RNN-T, and attention-based encoder-decoder (Li et al., 2022), as discussed in Section 2.2. Meanwhile, several training strategies enhance the noise robustness.

• **Multi-condition training or multi-style training**: One of the most commonly-used strategies is known as *multi-condition training* or *multi-style training* (Haeb-Umbach et al., 2021). This method combines data from various conditions/styles to train the DASR system. Typically, a large amount of clean and simulated noisy speech data are used together with a small amount of real-recorded noisy data, i.e., $\mathcal{D}^{\text{ASR}} = \{\mathcal{D}^{\text{ASR}}_{\text{clean}}, \mathcal{D}^{\text{ASR}}_{\text{simulated}}, \mathcal{D}^{\text{ASR}}_{\text{real}}\}$. Clean data helps model convergence, while real data enhances generalization in realistic conditions.

• **Domain adaptation**: Another popular line of research is called *domain adaptation*, aiming to transfer a model trained on the source domain (e.g., simulated noisy speech $\mathcal{D}^{\text{ASR}}_{\text{simulated}}$) to a target domain (e.g., real-recorded speech $\mathcal{D}^{\text{ASR}}_{\text{real}}$) with a limited amount of labeled or unlabeled target domain data. Domain adversarial training (DAT) (Shinohara, 2016a) is a typical example, which introduces an additional classifier (named discriminator G^{cls}) in the DASR architecture. It shares speech features from the original ASR model and is trained to classify whether the input speech belongs to the source domain $\mathcal{D}^{\text{ASR}}_{\text{simulated}}$ or the target domain $\mathcal{D}^{\text{ASR}}_{\text{real}}$. The DASR model is trained to fool the discriminator, resulting in domain-invariant ASR features and improved noise robustness across domains.

• **Handling multi-talker situations**: Similar to the multi-speaker SSE, addressing the challenge of multi-speaker DASR involves complexities in defining the optimal formulation. The presentation format of transcriptions, $\{Y^k\}_{k=1,\dots,K}$ and $\{\hat{Y}_k\}_{k=1,\dots,K}$, lacks standardization, with no predefined alignment between predictions and ground truth, leading to intractable possibilities. To tackle multi-speaker DASR challenges, various approaches have been proposed. One strategy involves treating transcriptions for different speakers as separate entities, aligning individual prediction sequences with corresponding ground truth sequences. Two common alignment methods are: 1) using a single predefined order; 2) considering all possible orders following PIT, as introduced in Section 2.2.1. Alternatively, an approach rearranges transcriptions into a meta-sequence during model training based on heuristic clues. This can be achieved by concatenating all utterances in a First In, First Out (FIFO) order based on onset time (Kanda et al., 2020b). These approaches effectively transform the multi-speaker DASR problem from an ill-defined task into a well-structured one.

Finally, it is worth noting that both single- and multi-channel DASR systems can be designed easily by adapting the input convolutional encoder to take different channels as input or by combining multi-channel features (e.g., inter-channel phase difference) into a single feature via feature fusion.

2.3.2 Advantages and disadvantages

As can be seen in Section 2.3.1, data-driven non-modular DASR approaches enjoy a simple system design, with only one model for optimization. The architecture design is also simple in the sense that the only goal is to maximize the DASR performance. The system development is relatively easy since limited expert knowledge is needed compared to modular systems. On the other hand, such an E2E design results in a black-box model that lacks interpretability. It is thus difficult to analyze the causes of performance degradation and partially optimize the system to alleviate such issues. In addition, data collection for training such a system is also relatively costly since we require accurate transcriptions of speech data. Finally, since the approaches are purely data-driven, the generalizability largely depends on the data, and it is difficult to leverage expert knowledge to mitigate overfitting.

2.4 From cascade to E2E joint-optimization in modular-based systems

Described in Sec. 2.2, modular-based DASR systems involves the direct cascade integration of SSE, FE, and ASR modules within a pipeline shown in Fig. 2.1. This method, chosen for simplicity, allows swift DASR system deployment, especially in time-sensitive scenarios. The straightforward and expeditious integration facilitates the easy component replacement or updating, preserving modularity for quick system readiness. Yet, the straightforward integration of modules may exhibit drawbacks compared to non-modular models, as individual modules optimized for distinct tasks may lack perfect alignment with the final target. In contrast, non-modular models undergo optimization directly based on the final target, ASR. Moreover, each individual component could be trained on the data with different characteristics from the output of its preceding module, leading to even worse performance due to the domain mismatch.

To address the issues in modular-based DASR systems, a sophisticated strategy involving joint optimization was developed. Key aspects of the joint optimization encompass training the entire model by optimizing the final target using in-domain data, which minimizes discrepancies between different modules. In-domain parallel data, $\mathcal{D}^{\text{Joint}} : \{\mathbf{X}, \mathbf{Y}\}$, are used in the process. $\mathcal{D}^{\text{Joint}}$ may also contains the clean speech signal $\mathbf{S}$ in some cases. The major optimization process is based on the ASR loss, $\mathcal{L}^{\text{ASR}}$:

$$\hat{\theta} = \underset{\theta=\{\alpha,\beta,\gamma\}}{\operatorname{argmin}} \sum_{\{\mathbf{Y},\mathbf{X}\}\in\mathcal{D}^{\text{DASR}}} \mathcal{L}^{\text{ASR}}(\mathbf{Y}, (\mathrm{G}^{\text{ASR}}_{\alpha} \circ \mathrm{G}^{\text{FE}}_{\beta} \circ \mathrm{G}^{\text{SSE}}_{\gamma})(\mathbf{X})). \tag{2.8}$$

Gradients of all parameters, $\{\alpha, \beta, \gamma\}$, are computed based on $\mathcal{L}^{\text{ASR}}$. Note that the other losses, $\mathcal{L}^{\text{SSE}}$ and $\mathcal{L}^{\text{FE}}$, are occasionally employed to regularize the learning of the corresponding parameters.

2.4.1 Building an E2E Modular System

Model selection: Ensuring the entire system is "differentiable" is crucial for establishing a back-propagation path from $\mathcal{L}^{\text{ASR}}$ to all modules of the system. In the realm of purely data-driven approaches, typically constructed with neural networks, each SSE, FE, and ASR module is inherently differentiable, rendering the entire cascaded system also differentiable. However, more consideration is needed for model-based approaches. Model-based approaches can be classified into three types: 1) knowledge-based deterministic operation, 2) optimization with closed-form solutions, and 3) optimization with iterative optimization algorithms. Knowledge-based deterministic operations, like MFCC and Fbank, consist of sequences of vector/matrix operations akin to neural networks, making the entire operation differentiable. In contrast, most model-based approaches formulate observations mathematically, involving optimization problems based on their objectives. Approaches with closed-form solutions, such as mask-based beamformer mentioned in Section 2.2.1, derive deterministic operations through vector/matrix operations, ensuring differentiability. Others without closed-form solutions require iterative optimization algorithms, like the majorization-minimization (MM) algorithm (Sun et al., 2016). While such approaches include optimization procedures in the inference stage, making them seemingly incompatible with the joint training framework, the unfolding technique (Monga et al., 2021) considers iterative optimization as a sequence of vector/matrix operations, rendering even these approaches differentiable. In summary, many data-driven and model-based approaches, including their mixed versions, can be treated as sets of differentiable operations, making them applicable to the E2E modular framework. It is essential to note that operations resulting in selections, such as median, argmax, and max, render the entire system non-differentiable, making them unsuitable for gradient descent-based approaches, specifically within the E2E modular framework.

Optimization: Assuming all modules are differentiable, optimizing the entire system can be achieved via gradient descent. However, this process is not always straightforward, especially with complex modules. For instance, signal processing modules, like mask-based beamforming, introduce complex-valued matrix operations (e.g., matrix inversion and eigenvalue decomposition) within the computational graph, leading to potential instability in training, including invalid gradients or loss values and poor convergence. To address such challenges, various techniques, such as diagonal loading, mask flooring, and optimized implementations, have been explored to significantly enhance training stability (Zhang et al., 2022a).

Pre-training and fine-tuning: Successful trials have been conducted in training multi-channel DASR systems from scratch in both single-/multi-speaker cases (Chang et al., 2019c). However, with deeper and larger models, pretraining the module parameters proves to be more effective (Masuyama et al., 2023b). Subsequent joint optimization refines these pre-trained parameters, tailoring them to the specific task at hand for optimal integration. Fine-tuning enables the modular-based model to adapt effectively to task-specific characteristics, ensuring robust performance in diverse real-world scenarios, especially when labeled data for the target domain is limited. Leveraging pre-trained parameters enhances fine-tuning efficiency, addressing challenges associated with sparse labeled data. While training large and deep models can be unstable and resource-intensive, the initialization and fine-tuning processes serve to alleviate these issues.

2.4.2 Developing difficulties and challenges

Joint optimization, essential for tailoring the model to task-specific nuances, may pose computational challenges, particularly with large and deep architectures. Training such models in cascaded DASR systems introduces complexities, requiring careful consideration of stability, convergence, and avoiding overfitting. In resource-limited environments, practical challenges may arise in joint optimization, and mitigating these challenges requires efficient strategies such as model compression, transfer learning, or adaptation to smaller datasets.

Another critical issue is the availability and quality of training data. The success of joint optimization relies on in-domain parallel data, denoted as $\mathcal{D}^{\text{Joint}}$, which is a significant hurdle. The adaptability and generalization capabilities of the jointly optimized model are significantly influenced by the quality and diversity of this data. Addressing issues like data distribution shifts between pre-training and fine-tuning stages is essential to prevent performance degradation. The looming risk of overfitting to specific training data, especially in scenarios with small or unrepresentative labeled datasets, emphasizes the need for advanced regularization techniques and prudent model complexity management.

Despite achieving good performance on the final target task, such as ASR, the output of intermediate modules may not meet expectations. Misalignment in evaluation metrics between the final task and early stages can lead to sub-optimal representations for intermediate modules during the forward process. However, this issue may also stem from the optimization process.

2.5 Conclusion

This chapter overviews distant speech recognition as an important downstream application of model-based and data-driven audio signal processing. It categorizes distant speech recognition systems into non-modular, modular, and E2E modular systems. The modular-based systems, explored in detail, showcase sub-modules like SSE, FE, and ASR operating under diverse acoustic conditions. Representative methods for each sub-module, categorized into model-based, data-driven, and mixed approaches, are discussed. The advantages and disadvantages of different integration methods are examined, with a focus on the robust performance of E2E modular systems. The adaptability of these learning approaches is highlighted, indicating potential extensions to address challenges in DASR problems involving multiple speakers.

There are several challenges faced in designing E2E modular models and their perspectives. The most challenging issue is its complexity in the network architecture and optimization procedure in real scenarios. For example, when we apply them to more natural conversations in meeting and dinner party scenarios in CHiME-6 (Watanabe et al., 2020), we must extend our systems to further deal with speaker diarization and long-form recording processing. Such an extension results in increased computational demands and larger memory requirements due to increased model complexity and longer input sequences. Further, the current state-of-the-art modular system in these scenarios requires interactive processing in speaker diarization and SSE, further complicating the network architecture.

Another important challenge is a streaming capability. The modular system is often realized in an incremental processing manner where the following module has to wait for the preceding modules (e.g., ASR processing has to wait for SSE processing). Thus, together with the above iterative process, the modular system intrinsically has the latency issue from the incremental processing, which weakens the streaming capability. However, the E2E modular model can employ powerful optimization, leading to joint optimization of the entire system. This will bring tight integration across the modules and simplify the complicated interfaces between modules, including eliminating the iterative process.

The third challenge is the use of various types of data. For example, E2E models, both modular and non-modular, are typically data-hungry, demanding large amounts of matched pair data, which is highly costly in multi-channel multi-speaker conversation scenarios. However, modular models own effectiveness with various paired or unpaired data types. Different modules can be pre-trained using speech-only data, clean and noisy speech pair data generated by simulation, and single-channel speech and text pair data, respectively. Compared with the real multi-channel multi-speaker data, these data are relatively easy to obtain. This direction would be further explored with other data types, e.g., by integrating a large language model obtained with text-only data.

Finally, to achieve comprehensive machine listening capabilities by encompassing full auditory scenes, we need to consider diverse non-speech events and modalities alongside distant speech processing. This includes essential audio components such as sound and music events, each requiring dedicated processing modules. A promising avenue for research involves expanding E2E modular distant speech processing by integrating these sound and music event modules, which have been actively studied in audio signal processing, along with modules from other modalities like video and sensor data.

Having discussed the methodological foundations, we now turn our attention to the actual models designed to tackle various sub-problems in ASR. We will begin with the most common scenario: Single-Input Single-Output (SISO).

Part I

E2E-ASR for Single-channel-Input Single-speaker-Output (SISO)

Chapter 3

IRIS: monaural E2E ASR robust to noise

Summary

In this chapter, we delve into addressing the common challenges of Single-Input-Single-Output (SISO) Automatic Speech Recognition (ASR) through novel approaches that harness the power of self-supervised learning (SSL). SSL has emerged as a potent paradigm in various speech-related tasks, exemplified by successful models like Wav2Vec 2.0 (Baevski et al., 2020), HuBERT (Hsu et al., 2021a), and WavLM (Chen et al., 2021b). Our goal is to integrate speech enhancement, SSL, and ASR into a unified model to tackle environmental noise and room reverberations that degrade SISO-ASR performance. To achieve this, we propose innovative methods that leverage SSL for feature extraction, significantly enhancing the robustness of ASR systems to adverse acoustic conditions. The resulting model belongs to the E2E modular-based category mentioned in Sec. 2. By incorporating SSL within our system, we aim to extract discriminative speech representations from large-scale unlabeled speech corpora, leveraging the model's ability to learn from raw acoustic inputs without the need for explicit labels. Furthermore, our approach integrates speech enhancement techniques within the SSL-based ASR framework, enabling joint optimization to enhance speech quality and intelligibility. This combined approach offers a holistic solution to the challenges posed by environmental noise and reverberations, ultimately leading to substantial improvements in SISO-ASR performance. Through the exploration of these novel methodologies, we demonstrate the feasibility and effectiveness of leveraging SSL for enhancing SISO-ASR systems, showcasing advancements in speech recognition capabilities under adverse acoustic conditions. Our work contributes to the growing body of research aimed at developing robust and adaptable ASR systems capable of handling real-world speech scenarios with varying levels of acoustic complexity.

Xuankai Chang, Takashi Maekaku, Yuya Fujita, Shinji Watanabe. InterSpeech 2022

End-to-End Integration of Speech Recognition, Speech Enhancement, and Self-Supervised Learning Representation.

3.1 Introduction

In the past decade, deep learning has significantly pushed the development of automatic speech recognition (ASR) moving forward. Many interesting models and technologies have been proposed. Deep neural network-hidden Markov model (DNN-HMM) based hybrid system (Hinton et al., 2012) is one of the them. DNN-HMM hybrid ASR systems usually train a DNN to predict frame-aligned states, e.g. context-dependent phonemes. Recently, end-to-end speech recognition systems have become more and more popular. Several end-to-end ASR technologies were proposed, including connectionist temporal classification (CTC) (Graves et al., 2006), Transducer (Graves et al., 2013) and attention-based encoder-decoder (Chan et al., 2016; Kim et al., 2017b; Watanabe et al., 2018). A lot of existing speech recognition techniques exhibit strong performance in clean conditions. However, applying speech recognition in noisy environments is still challenging, especially in the monaural case. DNN-HMM hybrid ASR systems still outperform E2E ASR system on a well-known noisy speech corpus(Yang et al., 2022b), CHiME-4 corpus (Vincent et al., 2017b).

Usually, speech signals recorded in the real scenarios contain unpredicted noise. The noise is from the environment or the device imperfections, which degrades the ASR performance. The existing solutions to address the noisy speech recognition can be summarized as two categories. One is to train the ASR model robust to noise (Hannun et al., 2014a; Shinohara, 2016b; Kim et al., 2017a). The other is to use an dedicated model to improve the intelligibility of the noisy speech before sending it to the ASR model. Such preprocessing is one of the important topics in speech research, called speech enhancement (SE) or denoising (Loizou, 2007). The SE model and the ASR model can be trained separately or jointly (Ochiai et al., 2017a; Narayanan and Wang, 2014; Subramanian et al., 2019). However, it is well known that the monaural SE techniques produce distortions which deteriorates the ASR performance (Iwamoto et al., 2022; Zhang et al., 2021c).

Recently, self-supervised learning representations (SSLR) have demonstrated great potential in improving the speech recognition (Baevski et al., 2020; Zhang et al., 2020b; Hsu et al., 2021b; Chang et al., 2021). One primary drawback of current SSLR models is that the pre-training cost is too high for most of the research groups. As an alternative solution, some researchers fine-tune the pre-trained SSLR models to get their customized version (Pasad et al., 2021). In our previous study (Chang et al., 2021), we have shown that directly using the pre-trained Wav2Vec2.0 (Baevski et al., 2020) and HuBERT (Hsu et al., 2021b) for feature extraction improves the ASR performance. However, the improvement on mismatched conditions is usually limited. The result of CHiME-4

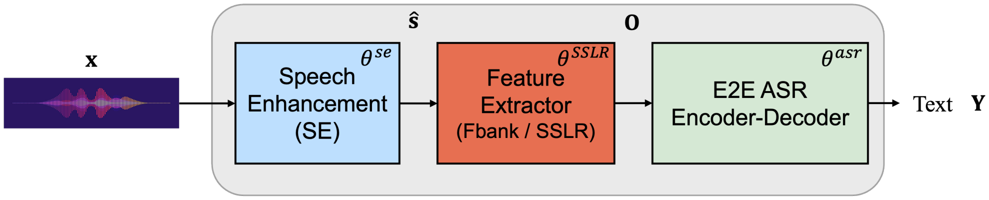


Figure 3.1: Overview of the proposed end-to-end model.

corpus in (Chang et al., 2021) shows the word error rate (WER) reduction for the multi-channel data with beamforming are much better than those for the isolated single channel. Because the audio of the latter set is relatively noisier than the former one. We believe that it is due to the mismatch between the pre-training and the target task. Wav2Vec2.0 and HuBERT models were pre-trained on the LibriLight (Kahn et al., 2020) data, a clean read English speech corpus. Later, WavLM (Chen et al., 2021b) was proposed to learn a representation model on simulated noisy / overlapped speech. In another recent work, Wang *et al.*(Wang et al., 2021b) proposed a noisy robust SSLR model based on Wav2Vec2.0, which also shows promising results on CHiME-4. But the model is not publicly available.

In this work, we propose a new model, called IRIS, for robust speech recognition, which integrates an SE module, an SSLR module and an ASR module into a single end-to-end model. We extensively investigate the benefits of the SE module and the SSLR module for robust speech recognition. Through experiments, we establish an efficient training scheme for the proposed E2E IRIS model. Finally, we show that our proposed model achieves state-of-the-art performance on the single-channel CHiME-4 ASR tasks.

3.2 E2E SISO ASR

We describe the proposed IRIS model in this section. The model includes a speech enhancement module (SE) and an self-supervised learning representation (SSLR)-based ASR (SSLR-ASR) module, shown in Figure 3.1. Each module can be trained separately. Then whole model can be fine-tuned with the objectives of speech enhancement and recognition. For the convenience of the following discussions, we denote the noisy speech input as $\mathbf{x} \in \mathbb{R}^T$.

Speech Enhancement

Most of the data collected in real scenarios contains not only speech signal but also undesired noise and reverberation. The target of speech enhancement is to keep the speech signal from data and to suppress the undesired signals. We denote the SE process as the following:

$$\hat{\mathrm{s}} = \mathrm{SE}(\mathbf{x};\, \theta^{\mathrm{se}}), \tag{3.1}$$

where $\hat{\mathrm{s}}$ is the enhanced speech and θ^{se} represents the parameters of the SE model.

A lot of powerful speech enhancement techniques have been proposed. In this work, we choose the Conv-TasNet proposed in (Luo and Mesgarani, 2019b) as the SE module. Conv-TasNet is a very successful model for end-to-end time-domain speech enhancement. Besides this, many other strong end-to-end time-domain speech enhancement models were proposed before (Luo and Mesgarani, 2018; Pandey and Wang, 2019; Luo et al., 2020). The advantage of the time-domain speech enhancement model is that we do not need to care about the phase when we generate enhanced signals. This might be helpful to reduce the distortion generated by speech enhancement models. Without loss of generality, any speech enhancement models can be used in our model.

SSLR-ASR

E2E-ASR

To recognize the speech, we use an E2E-ASR model. If we denote the input speech signal as $\hat{\mathrm{s}}$, the feature of the speech as $\mathbf{O}$ and the text as $\mathbf{Y}$, we can write the ASR process as:

$$\mathbf{O} = \mathrm{FeatureExtraction}(\hat{\mathrm{s}}), \tag{3.2}$$

$$\mathbf{Y} = \mathrm{ASR}(\mathbf{O};\, \theta^{\mathrm{asr}}), \tag{3.3}$$

where θ^{asr} represents the parameters of the ASR model. In this work, we use the joint CTC / attention-based encoder-decoder framework proposed in (Kim et al., 2017b) to build our E2E-ASR model. More details can be referred to (Kim et al., 2017b; Watanabe et al., 2018). It is worth to note that the choice of ASR is not limited to a specific architecture.

SSLR

Conventional ASR models use energy-based features such as log Mel-Filterbanks (Fbank) and mel-frequency cepstral coefficients (MFCC). In our previous work (Chang et al., 2021), we have shown that replacing the energy-based features with SSLRs can improving the performance of E2E-ASR. In this way, the Eq. 3.2 would be rewritten as:

$$\mathbf{O} = \mathrm{SSLR}(;\, \theta^{\mathrm{sslr}}), \tag{3.4}$$

where θ^{sslr} represents the parameters of the SSLR model.

SSLR models are learning-based speech representations. SSLR models are trained using large amount of unlabelled data. In this study, we propose to use WavLM proposed in (Chen et al., 2021b) to improve robust speech recognition. Similar to HuBERT (Hsu et al., 2021b), WavLM is trained to predict pseudo-labels of masked segments. In this way, WavLM / HuBERT learns the linguistic information from speech. The HuBERT model we used is trained on 60k hours of Libri-Light (Kahn et al., 2020) speech data. Whereas the WavLM learns to handle the noise from the speaker identification, separation, and diarization tasks by training on 60k hours of Libri-Light (Kahn et al., 2020), 10k hours of GigaSpeech (Chen et al., 2021a), and 24k hours of Vox-Populi (Wang et al., 2021a). This motivates us to use WavLM to extract features for noisy speech.

End-to-End IRIS Model

Although WavLM shows good performance on the noisy speech input in downstream tasks, such as speaker identification, separation and diarization tasks (Chen et al., 2021b), it is still a question whether it can handle various noises. We propose to use a speech enhancement model to help the WavLM. Our end-to-end model can be written as:

$$\mathbf{Y} = \mathrm{ASR}\ (\ \mathrm{SSLR}\ (\ \mathrm{SE}(\mathbf{x};\ \theta^{\mathrm{se}});\ \theta^{\mathrm{sslr}}\ );\ \theta^{\mathrm{asr}}\ ) \tag{3.5}$$

The proposed model adopts a modularized design, where the SE module enhances the input noisy speech, SSLR module extracts the feature and the ASR mudule generates the transcription. We directly use the pre-trained SSLR models from existing works, which are publicly available. Usually, SSLR models are very large, which makes it difficult to train the whole model. To address this issue, all three modules are initialized by pre-trained models with parameters $\hat{\theta}^{\mathrm{se}}$, $\hat{\theta}^{\mathrm{sslr}}$ and $\hat{\theta}^{\mathrm{asr}}$, respectively. Then the parameters of SE and ASR are fine-tuned to be get better performance.

3.3 Experiment

Dataset: CHiME-4 Challenge Corpus

We carried out all the experiments on the CHiME-4 corpus (Vincent et al., 2017b), which is previously mentioned in Sec. 1.4.1. The following is a review of the dataset details. The dataset contains real and simulated six-channel noisy recordings of speech from Wall Street Journal (WSJ0) corpus. The recordings cover four noisy scenarios including bus, cafe, pedestrian and street. There are 1,600 real and 7,138 simulated utterances for training, 1,640 real and 1,640 simulated utterances for development, and 1,320 real and 1,320 simulated utterances for test.

All the channels of CHiME-4 simulated recordings are used to train the SE model. To train the ASR model, we exclude the second channel of the CHiME-4 training set. This brings slight improvement because the second channel faces backward. Besides the noisy utterances in CHiME-4, the clean Wall Street Journal (WSJ0 + WSJ1) utterances are also used to train the E2E ASR model, based on the original ESPnet CHiME-4 recipe[1]. In fine-tuning, we use the same data as in ASR training. During evaluation, the single-channel development and test sets are used.

Configurations

We use a relatively small Conv-TasNet enhancement model to save the computation. The encoder consists of an 1-D convolution layer, with 256 output channel (N). The kernel and stride sizes are 40 and 20 respectively. The decoder has a reverse 1-D convolution layer with corresponding hyper-parameters of encoder. In the separation part, the temporal convolutional network (TCN), 4 convolutional blocks (X) are repeated twice (R). The number of channels (H) and the kernel size (P) in convolutional blocks are 512 and 3, respectively. The bottleneck has 256 channels (B). More detailed meaning of the hyper-parameters can be referred to (Luo and Mesgarani, 2019b). SI-SNR(Luo and Mesgarani, 2018) is used to computed the enhancement loss between the reference signal and the enhanced signal. The enhancement model is optimized by adam algorithm with learning rate at 1×10^{-3}.

For the ASR model, we use Transformer block to build the encoder and the decoder. The ASR model contains 12 encoder and 6 decoder Transformer layers. For each Transformer layer, the number of attention heads is 4. The dimension of the linear projection is $2,048$. The encoder uses two convolutional layers to downsample the input feature sequence and the total frame shift is 40ms. The dropout is set to be 0.1. In addition to the log Mel-Filterbank (Fbank), we use two SSLR models as feature extractor including the HuBERT-large and WavLM-large. When using the SSLR as feature extractor, the feature dimension is reduced from $1,024$ to 128 with a linear layer before input to the encoder. The ASR model is optimized by adam algorithm with peak learning rate at 1×10^{-3} and 20k steps to warm up. Specaug (Park et al., 2019) is used for both Fbank and SSLR feature during training. In decoding, we use a Transformer language model based on character level, with weight 1.0 during beam search.

In the proposed IRIS model, each of the modules is initialized by pre-training. SE and ASR parameters are from the pre-trained models described above. The IRIS model is fine-tuned with 10 epochs using both the enhancement and ASR losses. The same optimizer algorithm for ASR model training is used, with learning rate at 5×10^{-4}. During the training of both ASR and IRIS models, the parameters of the SSLR models are not updated.

Unless otherwise mentioned, model averaging is performed over the 10 checkpoints with best accuracy during decoding.

[1] https://github.com/espnet/espnet/tree/master/egs2/chime4/asr1

E2E-ASR Model with SSLRs

In this part, we show the evaluation results of ASR models on the monaural CHiME-4 corpus. The word error rates (WERs) of both simulated and real speech recordings are computed on the development and the test sets. The results are shown in Table 3.1. The results of systems 1-4 are from existing research works. Among them, system 4 is based on E2E-ASR. The rest systems are built by hybrid ASR systems. We can observe that the best performance is achieved by the hybrid ASR method. We have trained systems 5-7. In system 5, we use the conventional Fbank feature to train the E2E-ASR model, the performance of which is worse than system 3 by a large gap. In system 6 and 7, we use the HuBERT and WavLM models, which are pre-trained on large amount of unlabelled data, to extract feature. When using HuBERT to generate speech features, there is no consistent or obvious improvement across all the evaluation data. We conjecture that it is because the HuBERT is only pre-trained on the clean speech. This can be inferred from the performance of system 4 and 7. In system 4, the Wav2Vec2.0-based model was trained with noisy speech data, leading to similar performance as system 3. Likewise, system 7 using WavLM for feature extraction also achieves comparable performance with system 3. The WERs of simulated speech are 5.9% and 8.2% on dev and test sets, respectively, and those of real speech are 4.0% and 4.5%. Specially, in the test set, the WERs of real recordings is 28% better than the previous best results. From this results, we find that it is important to use noisy data to train the robust speech SSLR.

Table 3.1: Single-channel CHiME-4 ASR performance (%WER) of the E2E-ASR model and previous studies on monaural dev and test sets. In system 6 and 7, HuBERT and WavLM are pre-trained models learned on different sets of external data.

ID	System	Model	Dev. Set		Test Set	
			Simu.	Real	Simu.	Real
1	Kaldi Baseline (Chen et al., 2018a)	Hybrid	6.81	5.58	12.15	11.42
2	Du *et al.* (Du et al., 2016)	Hybrid	6.61	4.55	11.81	9.15
3	Yang *et al.* (Yang et al., 2022b)	Hybrid	**4.99**	**3.35**	8.61	6.25
4	Wav2Vec-Switch (Wang et al., 2021b)	E2E	-	3.5	-	6.6
5	E2E Transformer - Fbank	E2E	11.32	9.43	19.67	17.99
6	E2E Transformer - HuBERT	E2E	11.56	9.13	18.02	20.41
7	E2E Transformer - WavLM	E2E	5.93	4.03	**8.25**	**4.47**

Table 3.2: Monaural CHiME-4 ASR performance (%WER) of the IRIS model. Different combinations of fine-tuning SE (FT. SE) and fine-tuning ASR (FT. ASR) are evaluted.

Enhancement	Feature	FT. SE	FT. ASR	Dev. Set		Test Set	
				Simu.	Real	Simu.	Real
Conv-TasNet	Fbank	✗	✗	17.22	16.76	30.28	32.50
	Fbank	✗	✓	11.42	9.92	21.16	21.82
	Fbank	✓	✗	9.20	8.33	17.01	16.56
	Fbank	✓	✓	9.52	7.94	17.42	15.24
	WavLM	✗	✗	5.96	4.37	13.52	12.11
	WavLM	✗	✓	5.45	4.04	12.68	11.57
	WavLM	✓	✗	3.54	2.27	6.73	4.90
	WavLM	✓	✓	**3.43**	**1.98**	**6.21**	**3.64**

Table 3.3: ASR performance (%WER) comparison between the proposed IRIS model and the best existing single- and multi-channel systems.

System	Track	Dev. Set		Test Set	
		Simu.	Real	Simu.	Real
IRIS (proposed)	1ch	3.43	1.98	6.21	3.64
Yang *et al.* (Yang et al., 2022b)	1ch	4.99	3.35	8.61	6.25
Du *et al. (Du et al., 2016)*	2ch	3.46	2.33	5.74	3.91
Wang *et al. (Wang et al., 2020)*	2ch	2.17	1.99	2.53	3.19
Kaldi Baseline (Chen et al., 2018a)	6ch	1.90	2.10	2.74	2.66
Wang *et al. (Wang et al., 2020)*	6ch	1.15	1.50	1.45	1.99

IRIS Model

Next, we evaluate our proposed IRIS models. From the results in Table 3.1, we already know that WavLM is robust in the noisy condition. In this part, we further investigate if adding a speech enhancement module is beneficial to the model. As a reference, we did the similar evaluation on the E2E-ASR based on Fbank. Considering the computation cost when concatenated with the ASR model, we choose Conv-TasNet as the enhancement model and reduce the number of parameters by using a shallow architecture described in Sec. 3.3. The SI-SNRs of the pre-trained speech enhancement model are 9.55 dB and 9.71 dB on the development and test sets, respectively.

First, we directly concatenate the speech enhancement model and E2E-ASR models to perform the speech recognition. The results are shown in the Table 3.2. If the simple concatenation is used, both the performance of the Fbank-based system and that of the WavLM-based system are degraded, compared with the results of system 5 and 7 in the previous table. This indicates that speech enhancement models do not necessarily improve the ASR performance on noisy speech, because the training objectives of speech enhancement and recognition are not very well aligned. It is a well-known phonomenon in previous research (Zhang et al., 2021c).

Second, if we keep the enhancement model fixed and fine-tune the ASR model with ASR loss, the performance of a WavLM-based system is slightly improved but not reaching the same level as system 7 in the previous table. We believe the artifacts from the enhancement model is difficult to handle by the WavLM. For the Fbank-based system, the performance degradation is mitigated. However, in the other way around, if we keep the ASR model fixed and fine-tune the enhancement model with both enhancement loss and the ASR loss, we find that the performance are significantly improved in WavLM-based model and Fbank-based model, especially on the simulation sets. We assume that the major reason is because only the simulation data is used to fine-tune the enhancement module.

As the last case, we fine-tune both enhancement and ASR models with the enhancement and ASR losses. We observe further improvements on both Fbank-based and WavLM-based models. For the WavLM system, the best performance is achieved. Compared to the system 7 in the previous table, WavLM without speech enhancement, WERs on all the evaluation sets are further improved with a nonneglectable improvement. In Table 3.3, we list the best result of existing systems from Table 3.1, the result of the 1st ranking system in CHiME-4 two- and six-channel track (Du et al., 2016) and the result of our end-to-end IRIS system. Our system achieves a new state-of-the-art performance on the monaural CHiME-4 ASR task[2], outperforming the best monaural system. More interestingly, the results are comparable to the CHiME-4 challenge best 2-channel results from (Du et al., 2016).

The results indicate that the noise robust SSLR can still suffer from the degradation of noise. We can greatly alleviate the problem by introducing a speech enhancement as pre-processing. However, it is critical to fine-tune both models jointly to eliminate the mismatch. This rule can be applied to Fbank-based E2E-ASR model as well.

Analysis

It is interesting to know how the fine-tuning improves the IRIS model. We show the ASR performance with the checkpoints in the middle of fine-tuning the IRIS model in Figure. 3.2. It can be observed that the fine-tuning converges very fast. After only one epoch, the WERs can reach a very good level. With the model average over the first 10 epochs, the best performance can be observed.

[2] The pre-trained SSLR has more parameters and uses more data.

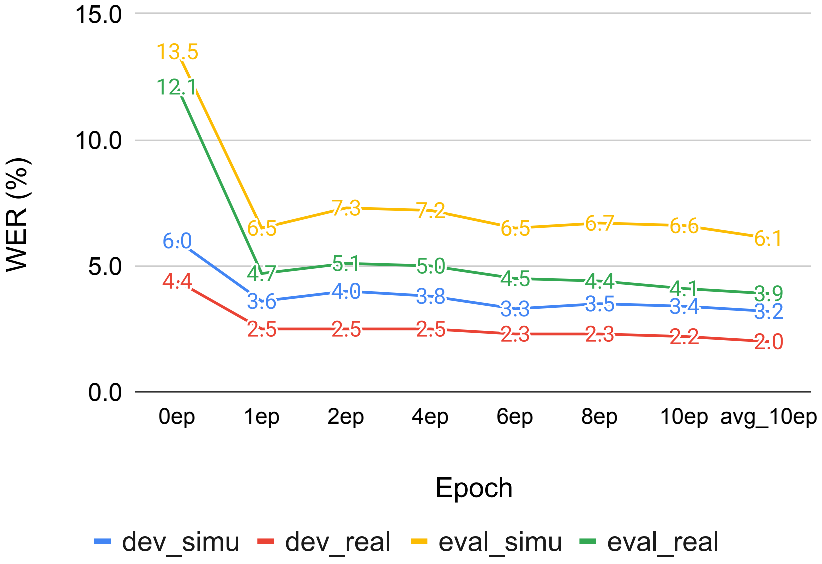


Figure 3.2: CHiME-4 ASR performance (WERs) of the IRIS model at different epochs during fine-tuning. Both SE and ASR are fine-tuned.

One difficult point is that the current IRIS model needs pre-training, taking extra efforts to prepare the individual enhancement and ASR models. In Figure 3.3, we show the training curves of the following models:

Model	Init. Param.	Update Param.
SSLR-ASR	$\hat{\theta}^{sslr}$	θ^{asr}
IRIS-random	$\hat{\theta}^{sslr}$	$\theta^{se}, \theta^{asr}$
IRIS-init-FT_SE	$\hat{\theta}^{se}, \hat{\theta}^{sslr}, \hat{\theta}^{asr}$	θ^{se}
IRIS-init-FT_ASR	$\hat{\theta}^{se}, \hat{\theta}^{sslr}, \hat{\theta}^{asr}$	θ^{asr}
IRIS-init-FT_SE+ASR	$\hat{\theta}^{se}, \hat{\theta}^{sslr}, \hat{\theta}^{asr}$	$\theta^{se}, \theta^{asr}$

We can see that training the IRIS model from random initialization could not converge to a good point. We assume that the deep architecture of the SSLR models might disturb the gradient back-propagation from ASR to the enhancement. More training tricks are required. However, if we initialize the parameters of each module, the training reaches a good level after the 1st epoch.

Another significant challenge arises from the large model size during joint optimization. The SSLR model we utilized, WavLM-Large, contains 316.62 million parameters, which constitute the bulk of the total parameters in our system. Fine-tuning the entire model, including the SE, SSLR, and ASR modules, is computationally demanding and makes it difficult to fit even a single

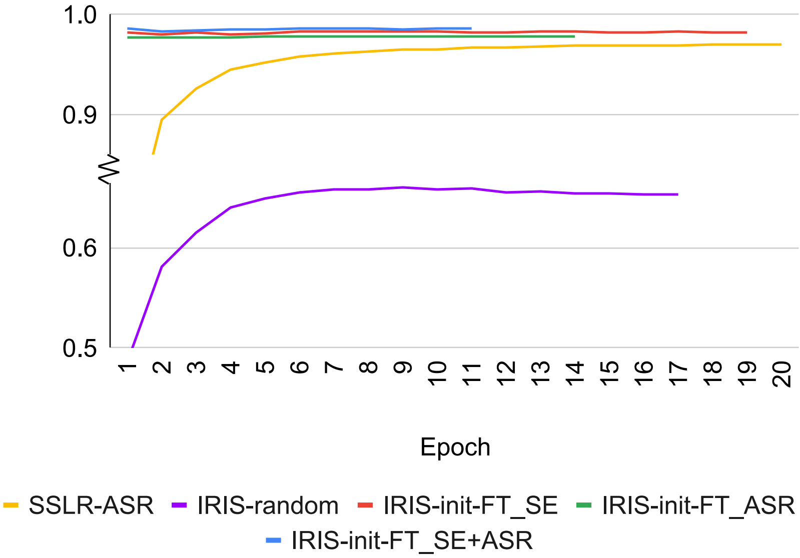


Figure 3.3: Accuracies on the development set for training and fine-tuning different models.

utterance into GPU memory. This constraint is the primary reason we did not jointly update the SSLR parameters. However, we believe that fine-tuning the SSLR parameters for feature extraction could lead to further improvements.

One potential solution to this problem is the use of adapters. By incorporating adapters, we can introduce a small number of trainable parameters while allowing the SSLR model to adapt to the downstream task. This approach would enable us to leverage the benefits of fine-tuning without overwhelming the computational resources.

3.4 Conclusions

We propose a new end-to-end model, IRIS, for robust speech recognition in this chapter. The model contains three modules including an SE module, an SSLR module and an ASR module. For the implementation, we use Conv-TasNet as SE module, WavLM as SSLR module and a joint CTC/attention-based encoder-decoder as ASR module. In the evaluation on monaural CHiME-4 task, the IRIS model outperforms the current state-of-the-art system, which is based on the hybrid ASR model. It should be noted that the pre-training of SSLR model uses more data and more parameters.

Having established the effectiveness of the IRIS model for single-input single-output (SISO) scenarios, we should turn our attention to more complex speech recognition challenges. The next chapter delves into the Multi-Input Single-Output (MISO) scenario, where we explore how leveraging multiple input channels can further improve ASR performance. With the spatial information provided by the multi-channel input, we aim to address the limitations of single-channel systems and enhance robustness in diverse and noisy environments.

Part II

E2E-ASR for Multi-channel-Input Single-speaker-Output (MISO)

Chapter 4

Enhancing real-world conversational speech recognition with speech foundation models

Summary

In the preceding section, we achieved significant advancements in noise-robust Automatic Speech Recognition (ASR) by introducing End-to-End (E2E) models that integrate speech enhancement, self-supervised learning (SSL) models, and speech recognition, yielding valuable insights and experience. To further push the boundaries of conversational speech recognition in real-world scenarios, our focus now shifts to multi-channel input signals, particularly within the context of real-life scenarios like the AMI meeting corpora (Carletta et al., 2005). Aligned with the prevailing trend in the machine learning community, we address this real-world challenge by leveraging large foundation models, notably Whisper (Radford et al., 2023). Originally designed for single-channel speech input, we extend the capabilities of the Whisper model to handle multi-channel speech signals, named MC-Whisper. Different from the IRIS model mentioned in Ch. 3, MC-Whisper adopts a non-modular architecture design, as described in Sec. 2. Such a design eliminates the need for explicit modular components and allowing for seamless integration of multi-channel input processing within the foundation model. Through the development of MC-Whisper, we aim to demonstrate the effectiveness of large foundation models in enhancing the robustness and adaptability of ASR systems to complex real-life contexts characterized by multi-channel speech inputs. By leveraging the scalability and representational power of foundation models like Whisper, our approach offers a promising pathway to address the challenges posed by multi-channel speech recognition, ultimately advancing the frontier of conversational speech recognition in diverse and dynamic environments.

Chang, Xuankai, Guo, Pengcheng, Fujita, Yuya, Maekaku, Takashi, and Watanabe, Shinji. Submitted to Signal Processing Letter. MC-Whisper: Extending Speech Foundation Models to Multichannel Distant Speech Recognition

4.1 Introduction

Significant advancements have been made in automatic speech recognition (ASR) (Hinton et al., 2012; Qian et al., 2016; Graves et al., 2006; Graves, 2012; Chorowski et al., 2015; Prabhavalkar et al., 2023) in recent decades, largely driven by deep learning techniques. These models, known for their data-intensive nature, achieve superior performance when trained on extensive and diverse datasets, showcasing robust generalization and knowledge transfer capabilities. Large-scale speech foundation models have gained significant attention due to their promising performance across various conditions, owing to their extensive model size and training data. These models fall into two primary categories: self-supervised training (Baevski et al., 2020; Hsu et al., 2021b; Chen et al., 2021b; Mohamed et al., 2022b) and supervised training (Radford et al., 2023; Zhang et al., 2023; Peng et al., 2023), with the former relying solely on input data without external supervised signals. Notably, most of the existing foundation models only take single-channel speech input, collected by a single-microphone device. This is mainly due to the ease of collection and cleaning of single-channel speech signals compared to multi-channel data. Recently, some studies have shown to successfully transfer the capability of foundation models to more complicated tasks. For example, Whisper (Radford et al., 2023), a representative foundation model, has been extended to handle multi-speaker overlapping speech (Li et al., 2023), commonly encountered in real-world scenarios. Nonetheless, these extensions still operate on single-channel input.

Within the domain of speech and audio processing, distant speech recognition (DASR) (Souden et al., 2009; Kumatani et al., 2012; Narayanan and Wang, 2014; Barker et al., 2017; Kinoshita et al., 2016; Heymann et al., 2017; Haeb-Umbach et al., 2021; Watanabe et al., 2020; Cornell et al., 2023) emerges as a pivotal application scenario. In DASR, the speech signal is captured by a device positioned at a considerable distance from the source, resulting in a signal infused with ambient noise and reverberations. DASR systems favor the use of multi-channel speech signals acquired through a multi-microphone device. This is because exploiting spatial information, embodied in multi-channel signals, can mitigate the background noise and reverberation levels(Van Veen and Buckley, 1988; Yoshioka and Nakatani, 2012; Barker et al., 2017; Kinoshita et al., 2016; Ochiai et al., 2017a; Erdogan et al., 2016; Heymann et al., 2016; Lu et al., 2022b). With speech separation and enhancement (SSE) techniques, the multi-channel input signal can be pre-processed before being fed into downstream models. Previous studies have revealed that a joint system of a multi-channel SSE and ASR can improve the DASR performance (Heymann et al., 2017; Wu et al., 2017; Xu et al., 2019; Masuyama et al., 2023a; Iwamoto et al., 2023).

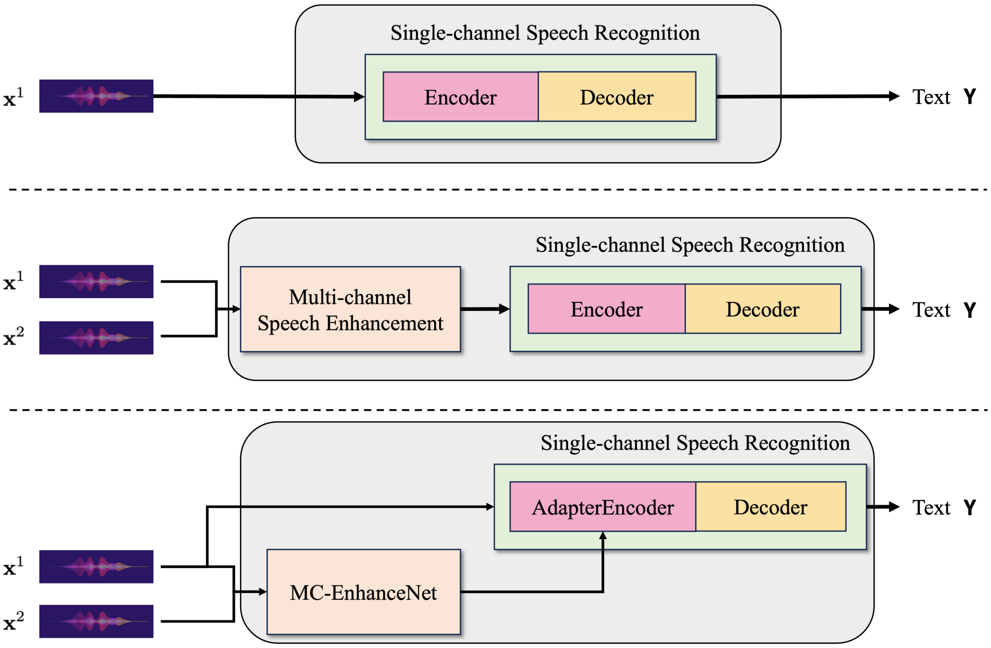


Figure 4.1: Different types of DASR models: (top) pure single-channel ASR; (middle) cascaded DASR with multi-channel speech enhancement module and single-channel ASR module; (bottom) proposed DASR with parallel multi-channel speech enhancement branch.

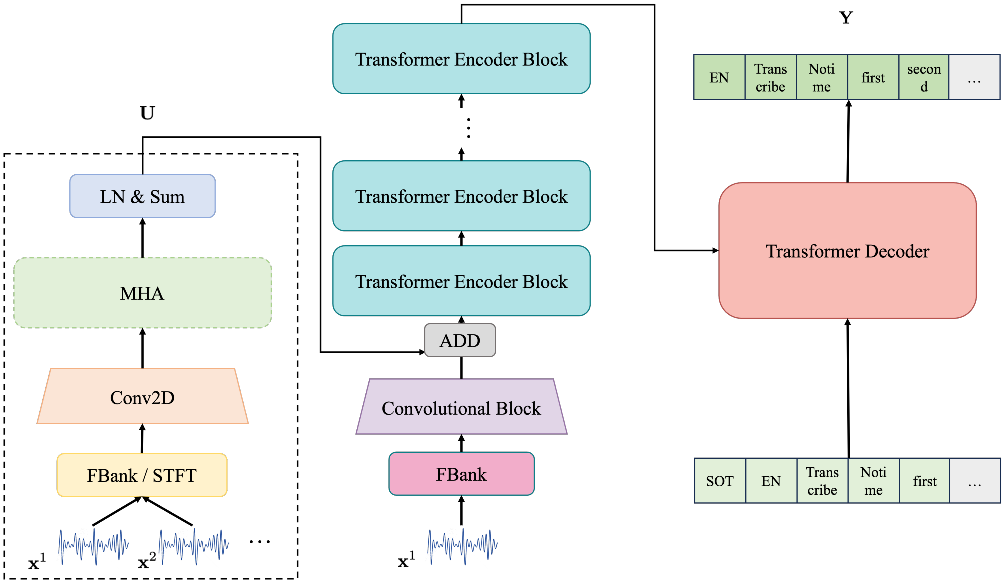


Figure 4.2: Model architecture of the proposed MC-Whisper. The dashed block on the left is the multi-channel sub-network. The encoded multi-channel embedding is injected into the encoder of the original foundation model via the ADD adapter.

Inspired by the joint model, we propose to extend the capabilities of the existing pre-trained speech foundation model, Whisper (Radford et al., 2023), to accommodate multi-channel (MC) conditions, thereby enhancing DASR performance. The incapacity of speech foundation models to handle multi-channel signals poses a potential obstacle to their accurate recognition of speech in real-world environments characterized by non-negligible noise and reverberation. However, constructing a multi-channel foundation model from the ground up proves impractical due to constraints related to data scarcity and training costs. In this study, we propose an innovative approach to overcome this limitation, called MC-Whisper, which introduces a parallel multi-channel processing sub-network into the original Whisper architecture. This sub-network processes multi-channel speech input separately and connects its output to the original Whisper encoder through specialized adapters (Huang et al., 2023). In contrast to previous joint models, which often rely heavily on the performance of the SSE frontend, this design maximizes the utilization of the foundation model's capabilities to process the multi-channel input. To train the model, we also explored the parameter-efficient fine-tuning based on the Low-Rank Adaptation (LoRA) (Hu et al., 2022), leading to improved performance while mitigating computational costs. We carried out experiments on the AMI meeting corpus (Carletta, 2006) distant-microphone recordings. Results show that the proposed methods improve the Whisper's ASR performance on AMI given the multiple distant microphone (MDM) recordings compared to both the single distant microphone (SDM) or BeamformIt (Anguera et al., 2007) processed counterparts that are conventionally used. To the best of our knowledge, this is the first effort in extending the speech foundation model, Whisper, to effectively operate under multi-channel conditions. Note that our proposed method is a general framework and can be migrated to most of the foundation models, such as OWSM (Peng et al., 2023) and HuBERT (Hsu et al., 2021b), similarly.

4.2 Proposed Model: MC-Whisper

In this section, we introduce the proposed model, MC-Whisper, as shown in Fig. 4.2. To provide context, we begin with an overview of the original Whisper model. Subsequently, we delve into the details of the multi-channel branch.

4.2.1 Background of Whisper

Whisper (Radford et al., 2023) has gained widespread recognition for its robust capabilities in speech recognition, phrase-level timestamp prediction, and speech translation. The model converts the single-channel speech input into the corresponding transcription, as illustrated in the *upper* section of Fig. 4.1. When presented with single-channel input audio $\mathbf{x}^1 \in \mathbb{R}^{1\times T}$, where T is the signal length, the model segments or pads it into 30-second chunks and converts it into the log-Mel filterbank (FBank) feature . These features undergo processing through two convolutional layers (ConvBlock) to reduce the input sequence length. Note that the superscript 1 is employed to signify that the speech comprises a single channel.

Trained on an extensive dataset comprising 680 thousand hours of speech data[1], Whisper is an end-to-end model implemented with the Transformer-based encoder-decoder architecture. The encoder maps the speech signal to hidden embeddings, and the decoder generates output in an autoregressive manner conditioned on both the encoder output and the tokens. This process is represented by the following equations:

$$\mathbf{E} = \text{ConvBlock}\left(\text{FBank}(\mathbf{x}^1)\right) \in \mathbb{R}^{T' \times D^{\text{in}}}, \tag{4.1}$$

$$y_t = \text{EncoderDecoder}\left(\mathbf{E}; \mathbf{p}, \mathbf{y}_{1:t-1}\right), \tag{4.2}$$

where $\mathbf{E}$ is the output from the convolutional layers with length T' and dimension D^{in}. $\mathbf{p}$ comprises a special token sequence containing task specifiers and tokens from the previous segment, while $\mathbf{y}_{1:t-1}$ represents the predicted tokens up to the previous time step.

It is worth noting that the pre-trained Whisper model offers multiple versions corresponding to different sizes. We focus on the medium and large versions, with 769M and 1550M parameters, respectively.

4.2.2 Multi-channel extension of Whisper

As discussed in Section 4.2.1, the original Whisper model is explicitly tailored for the processing of single-channel speech signals, as emphasized in Eq. (4.2). Nevertheless, in the context of DASR, the incorporation of multi-channel input assumes significance due to the inclusion of spatial information, leading to improving the performance of DASR (Ochiai et al., 2017a; Heymann et al., 2019; Haeb-Umbach et al., 2019; Watanabe et al., 2020).

Rather than training a multi-channel DASR model from scratch, considerable computational resources and effort can be conserved by leveraging a pre-trained ASR model, such as Whisper, known for its robust capabilities. To extend a pretrained single-channel ASR model to support multi-channel input, a common approach involves concatenating a multi-channel Speech Enhancement (SE) module and an ASR module into a unified system, forming a pipeline (Heymann et al., 2017; Xu et al., 2019). The SE module transforms the noisy multi-channel speech signal into a denoised single-channel speech signal, as illustrated in the *middle* part of Fig. 4.1. In such a design, the output of the SE module is vital to the ASR performance, overshadowing the contribution of the large foundation model.

In contrast, our approach adopts a distinct design by introducing a parallel multi-channel input

[1] 1 million hours for the latest large version

branch into the encoder, as depicted in the *bottom* of Fig. 4.1. We retain all components of the original encoder, which accepts single-channel speech as input, potentially leveraging the robustness of the original foundation model. A sub-network (MC-EnhanceNet) operates separately on the C-channel input speech signals, $\mathbf{X} = (\mathbf{x}^c \in \mathbb{R}^{1\times T} | 1 \leq c \leq C)$. Finally, the output of the MC-EnhanceNet can be seamlessly incorporated into the Whisper encoder through adapters, as illustrated in Fig. 4.2. Theoretically, the input to the MC-EnhanceNet can be any type of input to provide the spatial information.

MC-EnhanceNet: We propose two simple implementations of the MC-EnhanceNet which are independent on the number of channels. Two designs are different in the input features, using **F**Bank and **C**omplex spectrum, respectively. Consequently, the resulting models are denoted as MC-Whisper-F and MC-Whisper-C. The information from multi-channel input can be aggregated to enhance the DASR performance.

- **MC-Whisper-F**: For every channel of the input signal, $\mathbf{X}$, FBank features are extracted and stacked together along the channel axis. Subsequently, these features are transformed using a 2-layer convolutional block (Conv2D) and mapped to D-dimension embeddings. Conv2D also downsamples the length by halve, aligning with the original Whisper encoder's length. The computation is expressed as follows:

$$\mathbf{U} = \text{Conv2D}\left(\text{FBank}\left(\mathbf{X}\right)\right) \in \mathbb{R}^{T'\times D}, \tag{4.3}$$

  where $\mathbf{U}$ denotes embeddings with T' frames and size D.

- **MC-Whisper-C**: In distant speech processing, a complex spectrum is often employed to retain the phase information of the signal. We apply the Short Time Fourier Transform (STFT) on every channel of $\mathbf{X}$, and stack the real and imaginary components to form a new axis with size 2. Thus the stacked feature is denoted as $\mathbf{F}_\text{S} \in \mathbb{R}^{C\times 2\times T''\times D'}$, with T'' and D' being the number of frames and the dimension, respectively. Similar to the above FBank approach, a Conv2D module is used. Following this, multi-head attention (MHA) is applied along the channel axis to extract common information. The output is aggregated across all channels and followed by a LayerNorm (LN). The process is outlined as follows:

$$\mathbf{F}_\text{S} = \text{STFT}\left(\,\mathbf{X}\right) \qquad \in \mathbb{R}^{C\times 2\times T''\times D'}, \tag{4.4}$$

$$\mathbf{I} = \text{MHA}\left(\text{Conv2D}\left(\mathbf{F}_\text{S}\right)\,\right) \qquad \in \mathbb{R}^{C\times L\times D}, \tag{4.5}$$

$$\mathbf{U} = \text{LN}\left(\sum_{c=1}^{C}\mathbf{I}_c\right) \qquad \in \mathbb{R}^{L\times D}, \tag{4.6}$$

  where $\mathbf{I}$ represents the output from the multi-head attention, and $\mathbf{U}$ signifies the output, similar to the output in Eq. (4.3).

Adapter: To inject the multi-channel input embedding to the original Whisper encoder, we use the same ADD adapter in (Huang et al., 2023), and inject the information at the beginning of the Transformer encoder. Two linear projection layers are used to transform the injected embedding $\mathbf{U}$ and added to the original input of the encoder, $\mathbf{E}$ in Eq. (4.1). The computation can be denoted as:

$$\mathbf{E}' = \mathbf{E} + \text{ADD}(\mathbf{U}) \in \mathbb{R}^{L \times D^{\text{in}}}, \tag{4.7}$$

where $\mathbf{E}'$ is fed into the encoder, replacing $\mathbf{E}$ in Eq. (4.2).

4.2.3 Efficient training

To train the proposed model, one can optimize all parameters simultaneously with the ASR loss. However, given the relatively large size of the original Whisper model, we also investigate the usage of efficient parameter-tuning approaches. In this study, we examined LoRA (Hu et al., 2022). Note that other proper fine-tuning approaches can also be applied.

LoRA stands as a commonly utilized technique for efficiently adapting large foundation models to new datasets and tasks. Its primary concept involves injecting trainable rank decomposition matrices into each layer of the large Transformer model while keeping all pre-trained model weights frozen. Specifically, given a pre-trained weight matrix $\mathbf{W} \in \mathbb{R}^{d x k}$, where d and k represent the input and output dimensions, respectively, two new matrices $\mathbf{W}_b \in \mathbb{R}^{d \times r}$ and $\mathbf{W}_a \in \mathbb{R}^{r \times k}$ are introduced, with r representing the rank and $r \ll \min(d, k)$. The modified forward process can be formulated as:

$$\mathbf{h}^{\text{out}} = \mathbf{W}\mathbf{h}^{\text{in}} + \mathbf{W}_b\mathbf{W}_a\mathbf{h}^{\text{in}}, \tag{4.8}$$

where $\mathbf{W}_b$ and $\mathbf{W}_a$ are updated during fine-tuning, while $\mathbf{W}$ remains frozen, resulting in a significant reduction in memory footprint. $\mathbf{h}^{\text{in}}$ and $\mathbf{h}^{\text{out}}$ are input and output of the layer.

4.3 Experiments

4.3.1 Experimental Setup

All experiments were conducted using real-world English meeting recordings from the AMI meeting corpus (Carletta, 2006), which is previously mentioned in Sec. 1.4.2. The following is a review of the dataset details. The AMI corpus encompasses recordings obtained from both close-talking and far-field microphones. Our focus is on the far-field scenario, where an 8-channel microphone array, commonly referred to as multiple distant microphones (MDM), was employed. Conventionally, the 1^{st} channel of the MDM is selected to create an individual monaural condition known as a single distant microphone (SDM).

The AMI corpus provides human-annotated transcriptions. The speech recordings are segmented at the utterance-level to construct each individual training sample. Data pre-processing steps are detailed in the ESPnet (Watanabe et al., 2018) recipe[2]. The corpus comprises approximately 100 hours of meeting recordings. During training, speed perturbation augmentation is applied to augment the training data twofold, resulting in a total of approximately 232.3 hours. To conduct the single-channel experiments, we use the SDM condition of the AMI corpus. Additionally, we preprocess the MDM data, consisting of 8 channels, into a single channel using BeamformIt (MDM8+BFIt), the conventional data preprocessing method for MDM in the ESPnet recipe.

For the implementation, we use the ESPnet tools. For fast development, the medium version of Whisper trained on English data is employed, if not mentioned specifically. Details of the MC-EnhanceNet can be found in Section. 4.2.2. We only train the models for 3 epochs. A warm-up scheduler adjusts the learning rate, peaking at $1e-6$ with $25,000$ steps. During inference, the greedy decoding is used. For LoRA fine-tuning, we added LoRA adapters to all the query, key, and value projection layers, as well as the feed-forward layer, with rank $r=8$.

4.3.2 Results of MC-Whisper Medium

The experimental outcomes utilizing the Whisper-medium model with distant microphone recordings from the AMI corpus are detailed in Table 4.1. Initially, we perform DASR on the two single-channel conditions using the pretrained Whisper-medium without modification, corresponding to A1 and A2. While employing BeamformIt to enhance the data undoubtedly leads to performance improvement, the word error rate (WER) remains relatively high, hovering around 40%, making them unsuitable for real-world applications.

Subsequently, we fine-tune all parameters of Whisper-medium for the AMI distant microphone task, designated as B1-B4 in the table. Results of B1 and B2 demonstrate a notable performance boost through fine-tuning. Fine-tuning all parameters of the Whisper model on the SDM condition reduces the dev/eval WER by 48% and 45%, respectively. Similarly, notable improvement is observed in the MDM8+BFIt, achieving a 46% dev/eval WER reduction. We then explore the proposed MC-Whisper model, as indicated in Table 4.1 at rows B3/B4. When utilizing FBank features from multiple channels as input for the multi-channel sub-network, the model (B3) achieves marginally better performance compared to the system with BeamformIt preprocessing (B2). How-

[2] `https://github.com/espnet/espnet/blob/master/egs2/ami/asr1/local/data.sh`
No additional text normalization was performed to align with the Whisper.

Table 4.1: DASR performance (WER%) on the AMI corpus distant microphone recordings. Whisper medium.en checkpoints is used for all models in this table. For evaluating the single-channel based systems, the AMI-SDM and the AMI-MDM8 with BeamformIt (BFIt) is used. The Whisper model is updated via full finetuning (Full) or LoRA methods. The ratio of trainable parameters is included in the Fine-tuning column. For the proposed MC-Whisper systems, we denote the FBank-based MC-Whisper as **MC-Whisper-F**, and the complex spectrum-based MC-Whisper as **MC-Whisper-C**.

Model ID	Method	Fine-tuning (ratio)	Audio device	WER (dev/eval)
A1	Whisper	-	SDM	41.1 / 44.4
A2	Whisper	-	MDM8 + BFIt	37.5 / 41.0
B1	Whisper	Full (100%)	SDM	21.5 / 24.4
B2	Whisper	Full (100%)	MDM8 + BFIt	20.4 / 22.0
B3	MC-Whisper-F	Full (100%)	MDM8	20.4 / 21.8
B4	MC-Whisper-C	Full (100%)	MDM8	**19.8 / 21.0**
C1	Whisper	LoRA (0.6%)	SDM	22.4 / 26.2
C2	Whisper	LoRA (0.6%)	MDM8 + BFIt	20.8 / 23.5
C3	MC-Whisper-F	LoRA (0.6%)	MDM8	20.7 / 22.6
C4	MC-Whisper-C	LoRA (0.7%)	MDM8	**20.4 / 22.1**

Table 4.2: DASR performance (WER%) on the AMI corpus distant microphone recordings. Different pre-trained Whisper model are used: medium.en (M) and large (L). For evaluating the single-channel based systems, the AMI-SDM and the AMI-MDM8 with BeamformIt (BFIt) are used. The ratio of trainable parameters is included (adapter and MC-EnhanceNet). For the proposed MC-Whisper systems, we denote the complex spectrum-based MC-Whisper as **MC-Whisper-C**.

Model ID	Method	Fine-tuning (ratio)	Audio device	WER (dev/eval)
A1	Whisper (M)	-	SDM	41.1 / 44.4
A2	Whisper (M)	-	MDM8 + BFIt	37.5 / 41.0
C2	Whisper (M)	LoRA (0.6%)	MDM8 + BFIt	20.8 / 23.5
C4	MC-Whisper-C (M)	LoRA (0.7%)	MDM8	20.4 / 22.1
L1	Whisper (L)	-	SDM	38.3 / 40.1
L2	Whisper (L)	-	MDM8 + BFIt	35.7 / 38.0
L3	Whisper (L)	LoRA(0.5%)	MDM8 + BFIt	20.3 / 21.1
L4	MC-Whisper-C (L)	LoRA (0.5%)	MDM8	**19.4 / 20.5**

ever, it is known that FBank features lose valuable spatial information, such as the phase. To address this, we employ complex spectrum-based input, which includes rich phase information. The proposed method (B4) yields the best performance, resulting in improvements of 3% and 5% on the dev/eval sets compared to B2, respectively.

Optimizing the entire model is computationally expensive. Thus we adopt the LoRA parameter-efficient fine-tuning method, as described in Section 4.2.3. Results at rows C1-C4 also demonstrates significant performance improvement, albeit slightly less effective than fine-tuning all parameters. Notably, only around 0.6% of parameters are trainable, considerably reducing computational costs. Similar to the trend observed in B1-B4, using BeamformIt leads to substantial improvement, underscoring the significance of spatial information in far-field scenarios. The proposed methods, especially the complex spectrum-based design, demonstrate further improvement.

4.3.3 Results of MC-Whisper Large

To demonstrate the generalization ability of the proposed method, we further conducted experiments using Whisper-large and compared its performance with the medium version under similar settings. The results are summarized in Table. 4.2.

As baselines, we evaluate performance under the single-channel scenario (L1-L3). The large model consistently outperforms the medium one across all conditions. Furthermore, we observe fine-tuning significantly enhances performance, even with a minimal fraction of learnable parameters. Given the substantial memory and computation requirements of the large model, we exclusively employed LoRA fine-tuning.

Finally, we integrate the complex spectrum-based multi-channel sub-network into the Whisper-large. The model reduces the dev/eval WER by 4% and 7%, respectively, compared with L3. With the inclusion of multi-channel input, performance sees improvement, suggesting that the original capacity of Whisper-large can be augmented through the proposed extension.

4.4 Conclusion

In this section, we proposed a novel multi-channel extension for pre-trained large speech foundation models, enhancing their ability to process far-field speech. Rather than concatenating ad-hoc multi-channel pre-processing modules or altering the original model input, we introduced a parallel multi-channel sub-network, which may help preserve the robustness of the original model. Experimental results on the distant microphone AMI corpus demonstrate that our proposed method is effective for large foundation models. Looking ahead, further exploration of sophisticated sub-networks for other types of information and fusion methods holds promise for increasing downstream applications and improving performance in the future. In addition, exploring the long-form audio speech recognition in the multi-channel scenario is an important direction.

We now have successfully built a system to handle multi-channel inputs, we will shift our focus to another challenging aspect of speech recognition: multi-speaker overlapping speech. The next chapter delves into the Single-Input Multi-Output (SIMO) scenario, where we tackle the complexities of separating and transcribing speech from multiple speakers using a single-channel input. This involves developing advanced techniques to manage overlapping speech and enhance the robustness of ASR systems in scenarios where multiple speakers are simultaneously active.

Part III

E2E-ASR for Single-channel-Input Multi-speaker-Output (SIMO)

Chapter 5

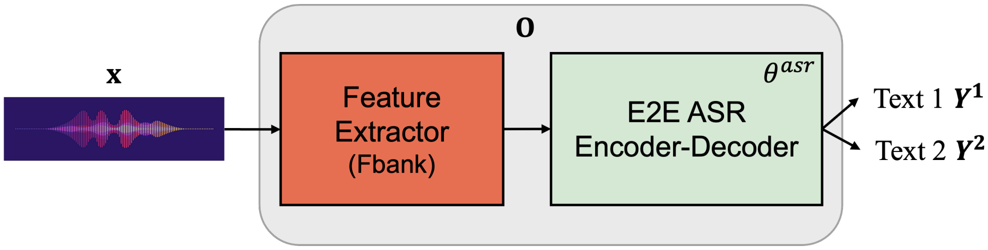
E2E-PIT-ASR: monaural E2E ASR for speech with overlaps

Summary

We start Part III of this thesis by delving into the Single-Input-Multiple-Output (SIMO) ASR task. This challenging scenario involves recognizing and transcribing overlapping speech from multiple speakers using a single-channel input, necessitating the separation and transcription of individual speakers' utterances from a mixture of voices without spatial cues. The overall pipeline can be shown in Fig. 5.1.

Within this chapter, our primary focus is on transcribing overlapped speech from multiple speakers using end-to-end (E2E) ASR models. These models are based on the non-modular architecture design, as described in Sec. 2. To simplify the problem, we introduce two key assumptions: firstly, we posit that the overlap persists from the beginning until near the end of the speech segment; secondly, we presuppose a fixed number of overlapping speakers. These assumptions, while helpful for model development and experimentation, represent idealized scenarios rarely encountered in real-world settings. The experiments conducted in this chapter are based on simulated data to explore fundamental concepts and methodologies. Nonetheless, we acknowledge the necessity of addressing and mitigating these assumptions to enhance the model's applicability to real-world scenarios. Subsequent chapters, specifically Chapter 6 and Chapter 7, are dedicated to addressing these challenges by introducing novel approaches that accommodate variable speaker counts and realistic speech overlap scenarios.

Through these endeavors, we aim to advance the understanding and capability of E2E ASR models in handling overlapping speech, paving the way for more robust and versatile speech recognition systems capable of addressing the complexities of real-world conversational environments.



O
x
Feature Extractor (Fbank)
θ^{asr}
E2E ASR Encoder-Decoder
Text 1 $\boldsymbol{Y}^1$
Text 2 $\boldsymbol{Y}^2$

Figure 5.1: Overview of the end-to-end SIMO model.

Xuankai Chang, Yanmin Qian, Kai Yu, Shinji Watanabe. ICASSP 2019. End-to-end monaural multi-speaker ASR system without pretraining.

5.1 Introduction

In the deep learning era, single-speaker automatic speech recognition systems have achieved a lot of progress. Deep neural networks (DNN) and hidden markov model (HMM) based hybrid systems have attained surprisingly good performance (Hinton et al., 2012; Sainath et al., 2013; Xiong et al., 2017). Recently, there has been a growing interest in developing end-to-end models for speech recognition (Kim et al., 2017b; Watanabe et al., 2017, 2018; Chen et al., 2018b), in which the various modules of the hybrid systems, such as the acoustic model (AM) and language model (LM), are folded into a single neural network model. Two major approaches of end-to-end speech recognition systems are connectionist temporal classification (Graves and Jaitly, 2014; Miao et al., 2015) and attention-based encoder-decoder (Chorowski et al., 2014; Chan et al., 2016). The performance of deep learning based conventional speech recognition systems has been reported to be comparable with, or even surpassing, human performance (Xiong et al., 2017). However, it is still extremely difficult to solve the cocktail party problem (Carletta et al., 2005; Cooke et al., 2010; Barker et al., 2018; Qian et al., 2018b), which refers to the task of separating and recognizing the speech from a specific speaker when it is interfered by noise and speech from other speakers.

To address the monaural multi-speaker speech separation and recognition problem, there has been a lot of research in single-channel multi-speaker speech separation and recognition, which aims to separate the overlapping speech and recognize the resulting separated speech individually, given a single-channel multiple-speaker mixtured speech. In (Hershey et al., 2016a; Isik et al., 2016), a method called deep clustering (DPCL) was proposed for speech separation. DPCL separates the mixed speech by training a neural network to project each time-frequency (T-F) unit into a high-dimensional embedding space, in which pairs of T-F units are close to each other if they have the same dominating speaker and farther away otherwise. In addition to segmentation using k-means clustering, a permutation-free mask objective was proposed to refine the output (Isik et al., 2016). In (Yu et al., 2017c; Kolbæk et al., 2017), a speech separation method called permutation invariant training (PIT) was proposed to train a compact deep neural network with the objective that minimizes the average minimum square error of the best output-target assignment at the utterance level. PIT was later extended to train a speech recognition model for multi-speaker speech mixture by directly optimizing with the ASR objective (Yu et al., 2017a; Chen et al., 2017; Chang et al., 2018b; Qian et al., 2018a). In (Settle et al., 2018; Seki et al., 2018), a joint CTC/attention-based encoder-decoder network for end-to-end speech recognition (Kim et al., 2017b; Watanabe et al., 2017) was applied to multi-speaker speech recognition. First, an encoder separates the mixed speech into hidden vector sequences for every speaker. Then an attention-based decoder is used to generate the label sequence for each speaker. To avoid label permutation problem, a CTC objective is used in permutation-free manner right after the encoder to determine the order of the label sequences. However, the model needs to first be pre-trained on single-speaker speech so that decent performance can be achieved.

In this chapter, we explore several new methods to refine the end-to-end speech recognition model for multi-speaker speech. Firstly, we revise the model in (Seki et al., 2018) so that pretraining on single-speaker speech is not required without loss of performance. Secondly, we propose to use speaker parallel attention modules. In previous work, the separated speech streams were treated equally in the decoder, regardless of the energy and speaker characteristics. We bring in multiple attention modules (Vaswani et al., 2017) for each speaker to enhance the speaker tracing ability and to alleviate the burden of the encoder similar to (Chang et al., 2018b). Another method is to use scheduled sampling (Bengio et al., 2015) to randomly choose the token from either the ground truth or the model prediction as the history information, which reduces the gap between training and inference in the sequence prediction tasks. This would be extremely helpful in our setup, since the separation is not always perfect and we often observe mixed label results. Schedule sampling can help to recover such errors during inference.

5.2 E2E SIMO ASR with PIT

In this section, we first describe the end-to-end ASR system for multi-speaker speech that has been used in (Seki et al., 2018). Then we introduce two techniques to improve the training process and performance of the end-to-end ASR multi-speaker system, namely the speaker parallel attention and scheduled sampling (Bengio et al., 2015).

End-to-End Multi-speaker ASR

In (Kim et al., 2017b; Watanabe et al., 2017; Hori et al., 2017), an end-to-end speech recognition model was proposed to take advantage of both the Connectionist Temporal Classification (CTC) and attention-based encoder-decoder, in aim of using the CTC to enhance the alignment ability of the model. An end-to-end model for multi-speaker speech recognition was brought up in (Seki et al., 2018), extending the joint CTC/attention-based encoder-decoder network to be applied on multi-speaker speech mixtures and to allow the permutation-free training in the objective function to address the permutation problem. The model is shown in Fig.5.2, in which the modules *Attention 1* and *Attention 2* share parameters. The input speech mixture is first explicitly separated into multiple sequences of vectors in the encoder, each representing a speaker source. These sequences are fed into the decoder to compute the conditional probabilities.

The encoder of the model can be divided into three stages, namely the $\text{Encoder}_{\text{Mix}}$, $\text{Encoder}_{\text{SD}}$ and $\text{Encoder}_{\text{Rec}}$. Let $\mathbf{x}$ denote an input speech mixture from K speakers. The first stage, $\text{Encoder}_{\text{Mix}}$, is the mixture encoder, which encodes the input speech mixture $\mathbf{x}$ as an intermediate representation $\mathbf{H}$. Then, the representation $\mathbf{H}$ is processed by K speaker-different (SD) encoders, $\text{Encoder}_{\text{SD}}$, with the outputs being referred to as feature sequences $\mathbf{H}^k, k = 1, \cdots, K$. $\text{Encoder}_{\text{Rec}}$, the last stage, transforms the features sequences to high-level representations $\mathbf{G}^k$, $k = 1, \cdots, K$. The encoder is computed as

$$\mathbf{H} = \text{Encoder}_{\text{Mix}}(\mathbf{x}) \tag{5.1}$$

$$\mathbf{H}^k = \text{Encoder}^k_{\text{SD}}(\mathbf{H}), k = 1, \cdots, K \tag{5.2}$$

$$\mathbf{G}^k = \text{Encoder}_{\text{Rec}}(\mathbf{H}^k), k = 1, \cdots, K \tag{5.3}$$

In the single-speaker joint CTC/attention-based encoder-decoder network, the CTC objective function is used to train the attention model encoder as an auxiliary task right after the encoder (Kim et al., 2017b; Watanabe et al., 2017; Hori et al., 2017). While in the multi-speaker framework, the CTC objective function is also used to perform the permutition-free training as in Eq.5.4, which is referred to as permutation invariant training in (Qian et al., 2018b; Yu et al., 2017c,a; Chen et al., 2017; Chang et al., 2018b; Qian et al., 2018a; Chang et al., 2018a; Tan et al., 2018).

$$\hat{\pi} = \arg\min_{\pi \in \mathcal{P}} \sum_k \text{Loss}_{\text{ctc}}(\hat{\mathbf{Y}}^k, \mathbf{Y}^{\pi(k)}), \tag{5.4}$$

where $\hat{\mathbf{Y}}^k$ is the predicted sequence variable computed from the encoder output $\mathbf{G}^k$, $\pi(k)$ is the k-th element in a permutation π of $\{1, \cdots, K\}$, and $\mathbf{Y}$ is the reference labels for K speakers. Later, the permutation $\hat{\pi}$ with minimum CTC loss is used for the reference labels in the attention-based decoder in order to reduce the computational cost. Note that PIT can be computationally expensive, as it involves traversing all possible permutations and calculating the loss function for each one. To mitigate this issue and avoid combinatorial explosion, the efficient Hungarian algorithm can be employed. This algorithm streamlines the process by optimizing the assignment problem, significantly reducing the computational burden associated with PIT.

After obtaining the representations $\mathbf{G}^k, k = 1, \cdots, K$ from the encoder, an attention-based decoder network is used to decode these streams and output label sequence $\hat{\mathbf{Y}}^k$ for each representation stream according to the permutation determined by the CTC objective function. For each pair of representation and reference label index $(k, \hat{\pi}(k))$, the decoding process is described as the following equations:

$$p_{\text{att}}(\hat{\mathbf{Y}}^{k,\hat{\pi}(k)}|\mathbf{x}) = \prod_n p_{\text{att}}(\hat{\mathbf{y}}_n^{k,\hat{\pi}(k)}|\mathbf{x}, \hat{y}_{1:n-1}^{k,\hat{\pi}(k)}) \tag{5.5}$$

$$\mathbf{c}_n^{k,\hat{\pi}(k)} = \text{Attention}(\mathbf{a}_{n-1}^{k,\hat{\pi}(k)}, \mathbf{e}_{n-1}^{k,\hat{\pi}(k)}, \mathbf{G}^k) \tag{5.6}$$

$$\mathbf{e}_n^{k,\hat{\pi}(k)} = \text{Update}(\mathbf{e}_{n-1}^{k,\hat{\pi}(k)}, \mathbf{c}_{n-1}^{k,\hat{\pi}(k)}, \hat{y}_{n-1}^{\hat{\pi}(k)}) \tag{5.7}$$

$$\hat{y}_n^{k,\hat{\pi}(k)} \sim \text{Decoder}(\mathbf{c}_n^{k,\hat{\pi}(k)}, \hat{y}_{n-1}^{\hat{\pi}(k)}) \tag{5.8}$$

where $\mathbf{c}_n^{k,\hat{\pi}(k)}$ denotes the context vector, $\mathbf{e}_n^{k,\hat{\pi}(k)}$ is the hidden state of the decoder, and $y_n^{\hat{\pi}(k)}$ is the n-th element in the reference label sequence. During training, the reference label $y_{n-1}^{\hat{\pi}(k)}$ from $\mathbf{Y}$ is used as the history in the manner of teacher-forcing, instead of $y_{n-1}^{\hat{\pi}(k)}$ in Eq.5.7 and Eq.5.8. And, Eq.5.5 means the probability of the target label sequence $\hat{\mathbf{Y}} = \{y_1, \cdots, y_N\}$ that the attention-based encoder-decoder predicted, in which the probability of $\hat{y}_n$ at n-th time step is dependent on the previous sequence $\hat{y}_{1:n-1}$.

The final loss function is defined as

$$\mathcal{L}_{\text{mtl}} = \lambda \mathcal{L}_{\text{ctc}} + (1 - \lambda)\mathcal{L}_{\text{att}}, \tag{5.9}$$

$$\mathcal{L}_{\text{ctc}} = \sum_k \text{Loss}_{\text{ctc}}(\hat{\mathbf{Y}}^k, \mathbf{Y}^{\hat{\pi}(k)}), \tag{5.10}$$

$$\mathcal{L}_{\text{att}} = \sum_k \text{Loss}_{\text{att}}(\hat{\mathbf{Y}}^{k,\hat{\pi}(k)}, \mathbf{Y}^{\hat{\pi}(k)}), \tag{5.11}$$

where λ is the interpolation factor, and $0 \leq \lambda \leq 1$.

Speaker parallel attention modules

Due to the differences in the characteristics of speakers and energy, the encoder usually has to compensate for those differences while separating the speech. The motivation of speaker paral-

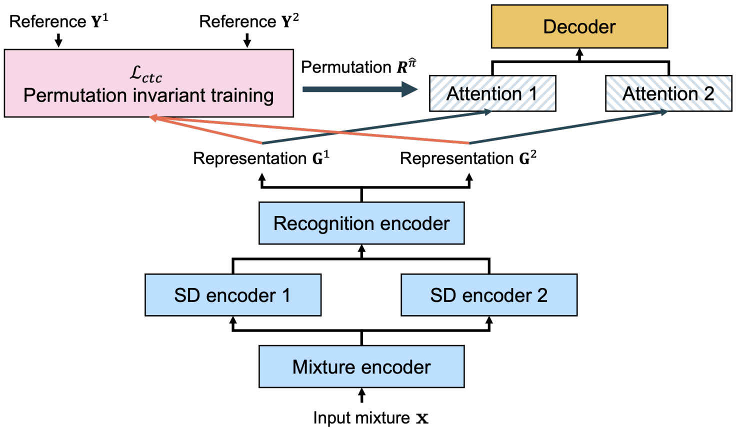

Reference $\mathbf{Y}^1$
Reference $\mathbf{Y}^2$
$\mathcal{L}_{ctc}$
Permutation invariant training
Permutation $\boldsymbol{R}^{\hat{\pi}}$
Decoder
Attention 1
Attention 2
Representation $\mathbf{G}^1$
Representation $\mathbf{G}^2$
Recognition encoder
SD encoder 1
SD encoder 2
Mixture encoder
Input mixture $\mathbf{x}$

Figure 5.2: End-to-End Multi-speaker Speech Recognition Model in the 2-Speaker Case

lel attention module that we proposed is to alleviate the burden for the encoder and to make the attention-decoder learn to filter the separated speech as well while keeping the model compact. In light of (Chang et al., 2018b), we proposed to use independent attention modules called speaker parallel attention. Fig.5.2 illustrates the architecture of the model, in which *Attention 1* and *Attention 2* are not sharing. The computation process in Eq.5.6 should be rewritten in a stream-specific way, in particular for the k-th stream, as:

$$\mathbf{c}_n^{k,\hat{\pi}(k)}, \mathbf{a}_n^{k,\hat{\pi}(k)} = \text{Attention}^s(\mathbf{a}_{n-1}^{k,\hat{\pi}(k)}, \mathbf{c}_{n-1}^{k,\hat{\pi}(k)}, \mathbf{G}^k) \tag{5.12}$$

Scheduled sampling

We generally trained the decoder network in a teacher-forcing fashion, which means the reference label token r_n, not the predicted token y_n, is used to predict the next token in the sequence during training. However, during inference, we are only accessible to the predicted token y_n from the model itself. This difference may lead to performance degradation, especially in the multi-speaker speech recognition task susceptible to the label permutation problem. We alleviate this problem by using the scheduled sampling technique (Bengio et al., 2015). During training, whether the history information is chosen from the ground truth label or the prediction is done randomly with a probability of p from the the prediction and $(1-p)$ from ground truth. Thus Eq.5.7 and Eq.5.8 should be changed as:

$$\mathbf{e}_n^{k,\hat{\pi}(k)} = \text{Update}(\mathbf{e}_{n-1}^{k,\hat{\pi}(k)}, \mathbf{c}_{n-1}^{k,\hat{\pi}(k)}, h), \tag{5.13}$$

$$\hat{y}_n^{k,\hat{\pi}(k)} \sim \text{Decoder}(\mathbf{c}_n^{k,\hat{\pi}(k)}, h), \tag{5.14}$$

where

$$b \sim Bernoulli(p), \tag{5.15}$$

$$h = \begin{cases} y_{n-1}^{\hat{\pi}(k)}, & if\ b = 0 \\ \hat{y}_{n-1}^{\hat{\pi}(k)}, & if\ b = 1 \end{cases} \tag{5.16}$$

5.3 Experiment

Experimental setup

To evaluate our method, we used the artificially generated single-channel two-speaker mixed signals, called WSJ-2Mix (Seki et al., 2018). As described in Sec. 1.4.3, WSJ-2Mix is derived from the Wall Street Journal (WSJ) speech corpus, using the tool released by MERL[1]. We used the WSJ SI284 to generate the training data, Dev93 for development and Eval92 for evaluation. The durations for the training, development and evaluation sets of the mixed data are 98.5 hr, 1.3 hr, and 0.8 hr respectively. In section 5.3, we also compared our model with previous works on the wsj0-2mix dataset, which is a standard speech separation and recognition benchmark (Hershey et al., 2016a; Isik et al., 2016; Settle et al., 2018).

The input feature is 80-dimensional log Mel filterbank coefficients with pitch features and their delta and delta delta features extracted using the Kaldi (Povey et al., 2011). Zero mean and unit variance are used to normalize the input features. All the joint CTC/attention-based encoder-decoder networks for end-to-end speech recognition were built based on the ESPnet (Watanabe et al., 2018) framework. The networks were initialized randomly from uniform distribution in the range -0.1 to 0.1. We used the AdaDelta algorithm with $\rho = 0.95$ and $\epsilon = 1e - 8$. During training, we set the interpolation factor λ in Eq.5.9 to be 0.2. We revise the deep neural network, replacing the original encoder layers with shallower but wider layers (Zeyer et al., 2018), so that the performance can be good enough without pre-training on single-speaker speech.

To make the model comparable, we set all the neural network models to have the same depth and similar size. We use the VGG-motivated CNN layers and bidirectional long-short term memory recurrent neural networks with projection (BLSTMP) as the encoder. The total depth of the encoder is 5, namely two CNN blocks and three layer BLSTMP layers. For all models, the decoder network has 1 layer of unidirectional long-short term memory network (LSTM) with 300 cells.

During decoding, we combined both the joint CTC/attention score and the pretrained word-level recurrent neural network language model (RNNLM) score, which had 1-layer LSTM with 1000 cells and was trained on the transcriptions from WSJ SI284, in a shallow fusion manner. We

[1] http://www.merl.com/demos/deep-clustering/create-speaker-mixtures.zip

set the beam width to be 30. The interpolation factor λ we used during decoding was 0.3, and the weight for RNNLM was 1.0.

Performance of baseline systems

In this section, we describe the performance of the baseline E2E ASR systems on multi-speaker mixed speech. The first baseline system is the joint CTC/attention-based encoder-decoder network for single-speaker speech trained on WSJ corpus, whose performance is 0.9% in terms of CER and 1.9% in terms of WER on the eval92_5k test set with the closed vocabulary. In the encoder, there are 3 layers of BLSTMP following the CNN and each BLSTMP layer has 1024 memory cells in each direction. The second baseline system is the joint CTC/attention-based encoder-decoder network for multi-speaker speech. The 2-layer CNN is used as the Encoder$_{\text{Mix}}$. The depth of the following BLSTMP layers is also 3 including 1 layer of BLSTMP as the Encoder$_{\text{SD}}$ and 2 layers of BLSTMP as the Encoder$_{\text{Rec}}$. The attention-decoder in the multi-speaker system is shared among representations $\mathbf{G}^s$, which is of the same architecture with single-speaker system. The results are shown in Table 5.1.

Model	dev CER	eval CER
single-speaker	79.13	76.52
multi-speaker (Seki et al., 2018)	n/a	13.7
multi-speaker	15.14	12.20
Model	dev WER	eval WER
single-speaker	113.47	112.21
multi-speaker	24.90	20.43

Table 5.1: Performance (Avg. CER & WER) (%) on 2-speaker mixed WSJ corpus. Comparison between End-to-End single-speaker and multi-speaker joint CTC/attention-based encoder-decoder systems

In the case of single-speaker, the CER and WER is measured by comparing the output against the reference labels of both speakers. From the table, we can see that the speech recognition system designed for multi-speaker can improve the performance for the overlapped speech significantly, leading to more than 80.0% relative reduction on both average CER and WER. As a comparison, we also include the CER result from (Seki et al., 2018) in the table, and it shows that the newly constructed end-to-end multi-speaker system without pretraining in this work can achieve better performance.

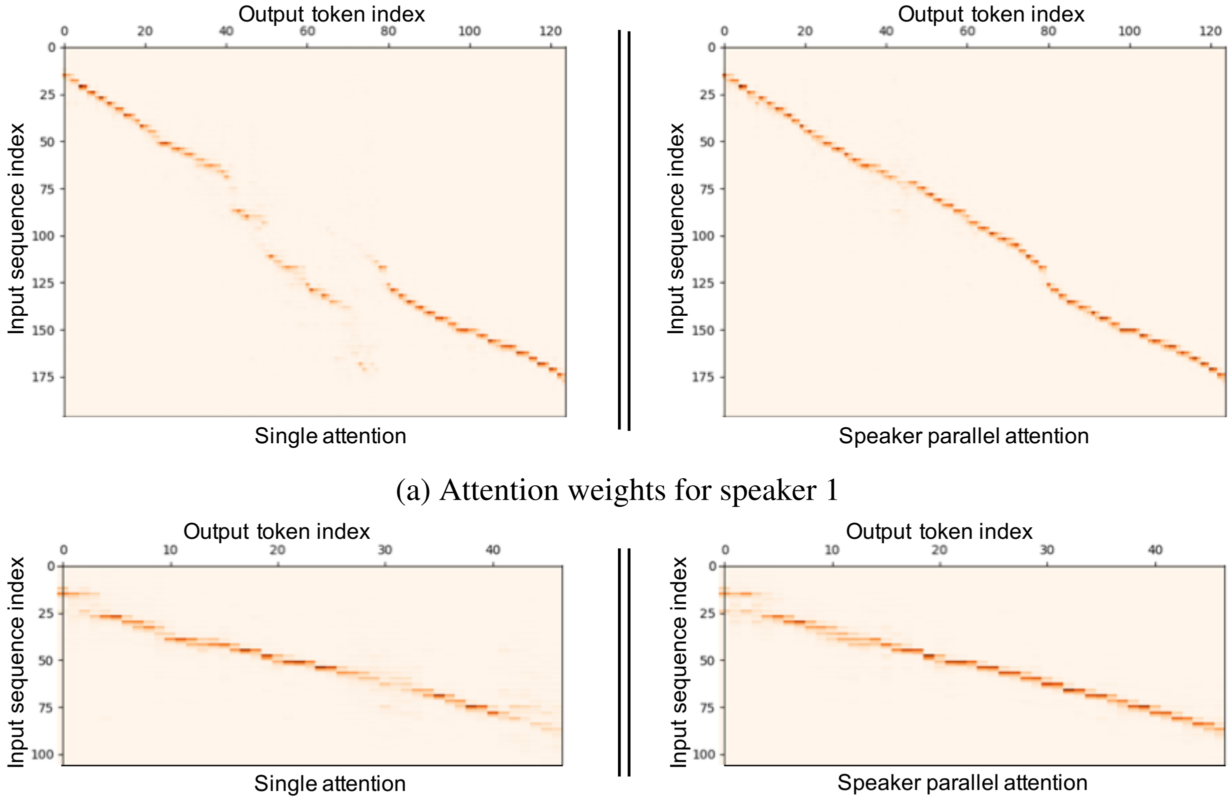
Performance of speaker parallel attention with scheduled sampling

In this section we report the results of the evaluation of our proposed methods. The first method is the speaker parallel attention, introducing independent attention modules for each speaker source instead of using a shared attention module. The rest of the network is kept the same as the baseline multi-speaker model, containing a 2-layer CNN Encoder$_{\text{Mix}}$, 1-layer BLSTMP Encoder$_{\text{SD}}$, a 2-layer BLSTMP Encoder$_{\text{Rec}}$, and a shared 1-layer LSTM as the decoder network. The performance is illustrated in the Table 5.2. The speaker parallel attention module reduces the average CER by 9% and average WER by 8% relatively. From the results we can tell that the CER is high, so the gap is large between the training and inference using the teacher-forcing fashion. Thus we adopted the scheduled sampling method with probability $p = 0.2$ in Eq. 5.15, which lead to a further improvement in performance. Finally, the system using both speaker parallel attention and scheduled sampling can obtain relative $\sim 10.0\%$ reduction on both CER and WER on the evaluation set.

Model	dev CER	eval CER
multi-speaker (baseline)	15.14	12.20
+ speaker parallel attention	14.80	11.11
++ scheduled sampling	**14.78**	**10.93**
Model	dev WER	eval WER
multi-speaker (baseline)	24.90	20.43
+ speaker parallel attention	24.88	18.76
++ scheduled sampling	**24.52**	**18.44**

Table 5.2: Performance (Avg. CER & WER) (%) on 2-speaker mixed WSJ corpus. Comparison between End-to-End multi-speaker joint CTC/attention-based encoder-decoder systems

We show the visualization of the attention weights sequences for two overlapped speakers, generated by the baseline single-attention multi-speaker end-to-end model and the proposed speaker-parallel-attention multi-speaker end-to-end model individually. The horizontal axis represents the output token sequence and the vertical axis represents the input sequence to the attention module. The left parts of Figures.5.3 (a) and (b) show the attention weights for speaker 1 and speaker 2 generated by the previous single-attention model. The right parts show the attention weights generated by the proposed speaker-parallel-attention model. We can observe that the right parts are more smooth and clear, and the attention weights are more concentrated. This observation conforms with the characteristics of alignments between output sequence and input sequence for speech recognition, and further shows the superiority of the proposed speaker parallel attentions.



(a) Attention weights for speaker 1

(b) Attention weights for speaker 2

Figure 5.3: Visualization of the attention weights sequences for two overlapped speakers. The left part is from the previous single-attention multi-speaker end-to-end model and the right part is from the proposed speaker-parallel-attention multi-speaker end-to-end model.

Comparison with previous work

We then compared our work with other related work. We trained and tested our model on wsj0-2mix dataset that was first used in (Hershey et al., 2016a). Table 5.3 shows the WER results of hybrid systems including PIT-ASR (Qian et al., 2018a), DPCL-based speech separation with Kaldi-based ASR (Isik et al., 2016), and the end-to-end systems constructed in (Seki et al., 2018) and ours in this paper. These were evaluated under the same evaluation data and metric as in (Isik et al., 2016) based on the wsj0-2mix. Noted that the model in (Seki et al., 2018) was trained on a different, larger training dataset than that used in other experiments. From Table. 5.3, we can observe that our new system constructed by the proposed methods in this paper is significantly better than the others.

Model	Avg. WER
DPCL+ASR (Isik et al., 2016)	30.8
PIT-ASR (Qian et al., 2018a)	28.2
End-to-end ASR (Char/Word-LM) (Seki et al., 2018)	28.2
Proposed End-to-end ASR with SPA (Word LM)	**25.4**

Table 5.3: WER (%) on 2-speaker mixed **WSJ0** corpus. The comparison is done between our proposed end-to-end ASR with speaker parallel attention (SPA) and previous works including DPCL+ASR, PIT-ASR and end-to-end ASR systems.

5.4 Conclusion

In this chapter, we have introduced an end-to-end multi-speaker speech recognition system under the joint CTC/attentin-based encoder-decoder framework. More specifically, a new neural network architecture enabled us to train the model from random initialization. And we adopted the speaker parallel attention module and scheduled sampling to improve performance over the previous end-to-end multi-speaker speech recognition system. The experiments on the 2-speaker mixed speech recognition show that the proposed new strategy can obtain a relative $\sim 10.0\%$ improvement on CER and WER reduction.

However, we operated under two assumptions: 1) the number of speakers in the overlapping speech is known, and 2) the speech segments are mostly overlapped. While these assumptions simplify the task and allow us to focus on complex ASR challenges, they are artificial and uncommon in real-world scenarios. To advance the robustness and applicability of our models, it is crucial to explore methods that remove these constraints.

In the next chapter, we begin by discarding the first assumption. Our focus will be on developing techniques to handle overlapping speech without prior knowledge of the number of speakers involved. This shift addresses a significant limitation of our current approach and moves us closer to more practical and versatile ASR solutions.

Chapter 6

Conditional-chain: monaural multi-speaker E2E ASR for various number of speakers

Summary

In the previous chapter, we introduced an end-to-end (E2E) Automatic Speech Recognition (ASR) model designed to transcribe speech signals with multiple overlapping speakers. However, it's essential to acknowledge that this model operated under specific constraints. Firstly, it assumed a fixed number of speakers in the output, which may not accurately reflect the variability encountered in real-world scenarios. Secondly, the model assumed that speech overlap persists from the beginning of the utterance, simplifying the problem but overlooking the complexities of varying speech overlaps.

In this chapter, we present the *conditional chain model*, a novel approach tailored to address scenarios with varying speaker counts. This model transcribes the speech of one speaker in each iteration, dynamically adjusting to accommodate multiple speakers until all utterances are transcribed. By leveraging previously estimated speaker features, the model maintains awareness of generated transcriptions to prevent redundancy effectively. To enhance prediction efficiency and support parallel inference, we adopt a non-autoregressive ASR approach based on connectionist temporal classification (CTC) (Graves et al., 2006). This strategy allows for simultaneous processing of multiple speaker utterances, significantly improving model performance and scalability in handling overlapping speech scenarios. Through these advancements, we aim to broaden the applicability and robustness of E2E ASR models, enabling more effective transcription of complex, real-world speech with varying speaker counts and overlapping segments. Similar to the previous chapter, the conditional chain model is based on the non-modular architecture design, as described in Sec. 2.

- Jing Shi*, Xuankai Chang*, Pengcheng Guo*, Shinji Watanabe, Yusuke Fujita, Jiaming Xu, Bo Xu, Lei Xie. 2020 NeurIPS. Sequence to multi-sequence learning via conditional chain mapping for mixture signals.
- Pengcheng Guo, Xuankai Chang, Shinji Watanabe, Lei Xie. 2021 Interspeech. Multi-speaker ASR combining non-autoregressive conformer CTC and conditional speaker chain.

6.1 Introduction

End-to-end architectures have demonstrated their effectiveness and became the dominant models across various sequence to sequence tasks, like neural machine translation (NMT) (Bahdanau et al., 2015; Vaswani et al., 2017) and automatic speech recognition (ASR) (Chan et al., 2016; Dong et al., 2018; Karita et al., 2019a; Gulati et al., 2020; Guo et al., 2021a). However, most of these models follow an autoregressive (AR) strategy, which predicts a target token conditioned on both previously generated tokens and the source input sequence. The incremental process makes it hard to compute parallel and results in a large latency during the inference. In contrary to AR models, non-autoregressive (NAR) models have drawn immense interest recently, aiming to get rid of the temporal dependency and perform parallel inference.

NAR models were first proposed in NMT and have achieved competitive performance with conventional AR models (Gu et al., 2018; Libovickỳ and Helcl, 2018; Lee et al., 2018; Stern et al., 2019; Gu et al., 2019; Ghazvininejad et al., 2019, 2020; Saharia et al., 2020). The idea of NAR models is to predict the whole target sequence within a *constant* number of iterations which is not strict with the sequence length. In (Gu et al., 2018), Gu *et al.* introduced a fertility module to predict the number of times each encoder output should be repeated and regraded the repeated encoder outputs as decoder input to perform parallel inference. In (Lee et al., 2018), Lee *et al.* proposed a deterministic NAR model by iteratively refine the outputs from corrupted predictions. In addition, there were lots of studies based on the insert or edit sequence generation (Stern et al., 2019; Gu et al., 2019), connectionist temporal classification (CTC) (Libovickỳ and Helcl, 2018), and masked language model objective (Ghazvininejad et al., 2019, 2020; Saharia et al., 2020).

Inspired by the success of NAR models in NMT, several NAR methods were also proposed to reach the performance of AR models on ASR (Chen et al., 2019; Higuchi et al., 2020; Chan et al., 2020; Tian et al., 2020; Higuchi et al., 2021; Chi et al., 2021; Fan et al., 2020). Since CTC learns a frame-wise latent alignment between the input speech and output tokens and predicts the target sequence based on a strong conditional independence assumption (Graves et al., 2006), it can be viewed as an early-stage realization of NAR ASR models. In (Chan et al., 2020), Imputer was proposed to iteratively generate a new CTC alignment based on mask prediction. Besides, Mask-

CTC (Higuchi et al., 2020, 2021) and Align-Refine (Chi et al., 2021) aimed to refine a token-level CTC output or latent alignments with the mask prediction. In (Tian et al., 2020), Tian *et al.* proposed to use the estimated CTC spikes to predict the length of target sequence and adopt the encoder states as the input of decoder. However, most of aforementioned methods mainly focus on sequence to sequence tasks, like NMT and single-speaker ASR, and it is hard to directly extended to sequence to multi-sequence tasks, like multi-speaker ASR.

Multi-speaker ASR aims to predict the corresponding transcription for each speaker from multiple speakers overlapping speech. Although lots of AR models were explored for multi-speaker ASR, such as permutation invariant training (PIT) (Qian et al., 2018a) or deep clustering (DPCL) (Menne et al., 2019) based hybrid system and recurrent neural network (RNN) or Transformer based end-to-end model models (Seki et al., 2018; Chang et al., 2019a, 2020; Kanda et al., 2020a; von Neumann et al., 2020b), few attempts have been made to realize NAR training. In this study, we revisit the proposed conditional chain based methods (von Neumann et al., 2019; Shi et al., 2020b,a; Fujita et al., 2020) and extend it to NAR multi-speaker ASR. By doing this, the output of each speaker is predicted one-by-one by making use of both the mixed input as well as previously-estimated conditional speaker features. In each prediction step, a CTC-based NAR encoder network is used to perform parallel computation. Since the performance of CTC may suffer a severe degradation due to the conditional independence assumption, we also explore adopting an advanced Conformer encoder (Gulati et al., 2020) architecture to capture both local and global acoustic dependencies and an additional intermediate loss (Lee and Watanabe, 2021) as a regularization function. Finally, while the original conditional chain model takes the token-level CTC alignments as the “hard" conditional speaker features, we propose to use “soft" conditions which are latent feature representations extracted after the last encoder layer. We evaluate the effectiveness of our model on two multi-speaker ASR benchmarks, WSJ0-Mix and LibriMix. Both results outperform other NAR models with a minor increment of latency and even achieve comparable results with the AR models.

6.2 E2E SIMO ASR with conditional chain

End-to-end models proposed in previous chapter 5 mainly focus on an AR strategy, which will be cumbered with a complex computation and large latency problems. Although an encoder-only CTC framework can be regarded as a NAR model, the system may be susceptible to performance degradation due to the conditional independence assumption. In this study, we revisit our proposed conditional speaker chain based method for NAR multi-speaker ASR. The improved model consists of a conditional speaker chain module and Conformer CTC encoders. While the conditional speaker chain explicitly models the relevance between outputs of different iterations, the

Conformer CTC aims to conduct NAR computation in each single step. The total inference steps are restricted to the number of mixed speakers. In addition, we also explore incorporating an intermediate CTC loss as a regularization function to further improve the system performance.

Conformer Encoder

Conformer encoder (Gulati et al., 2020) is a stacked multi-block architecture, which includes a multihead self-attention (MHSA) module, a convolution (CONV) module, and a pair of position-wise feed-forward (FNN) module in the Macaron-Net style. While the MHSA learns the global context, the CONV module efficiently captures the local correlations synchronously. Since the Conformer encoder has shown consistent improvement over a wide range of end-to-end speech processing applications (Guo et al., 2021a), we expect it to compensate for the modeling capacity of CTC and improve the system performance.

Intermediate CTC Loss

In (Lee and Watanabe, 2021), researchers proposed a simple but efficient auxiliary loss function for CTC based ASR models, named intermediate CTC loss. The main idea of intermediate CTC loss is to choose an intermediate layer within the encoder network and induce a sub-network by skipping all higher layers after the selected layer. By computing the additional CTC loss w.r.t the output of intermediate layer, the sub-network relies more on the lower layers instead of the higher layers, which can regularize the model training. Choosing the m-th layer from a L-layer encoder network, its output can be defined as $\mathbf{H}_m^k$. Thus, the final loss of our model becomes:

$$\begin{aligned}\mathcal{L} = \sum_{k=1}^{K} \big((1-\lambda)\mathcal{L}_{\text{CTC}}(\mathbf{G}^k, \mathbf{Y}^{\pi(k)}) + \\ \lambda\mathcal{L}_{\text{InterCTC}}(\mathbf{H}_m^k, \mathbf{Y}^{\pi(k)})\big)\,,\end{aligned} \tag{6.1}$$

where λ refers to the weight of intermediate loss and $k \in [1, \dots, K]$ is the speaker index. $\pi(k)$ represents the corresponding value at position k of the permutation. In this work, we set λ equals to 0.1 and choose a middle layer of the Encoder$_{\text{Rec}}$ as the intermediate layer ($m = L/2$).

Conditional Chain Model

Fig. 6.1 shows an overview of our model. Different from the AR models described in chapter 5, we replace the SD encoders with a conditional speaker chain module (CondChain) and predict the output of each speaker one-by-one. With a hidden mixture representation $\mathbf{H}$ computed in Eq. (5.1),

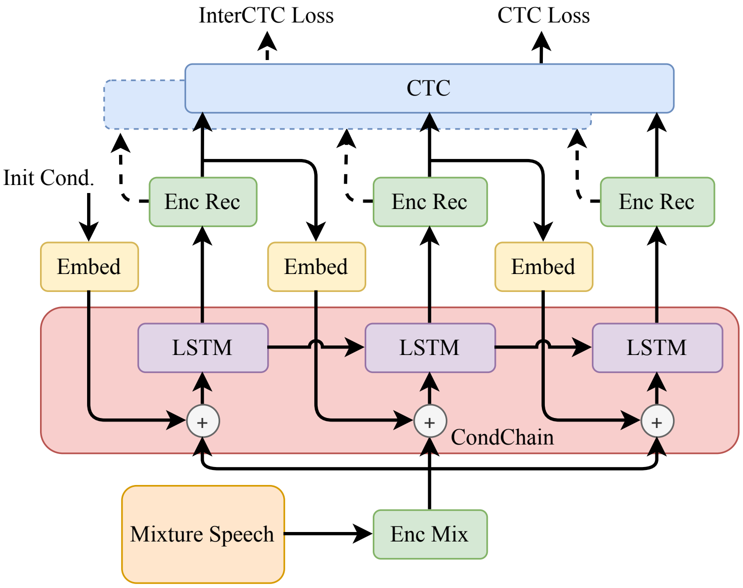


Figure 6.1: A overview of proposed conditional speaker chain based Conformer CTC model for multi-speaker ASR. This figure shows a training procedure of a 3-speaker mixed waveform. The parameters of blocks with the same name are shared.

the CondChain module extracts each speaker's speech representation by taking advantage of both the mixture representation $\mathbf{H}$ and the previously-estimated high-level embedding $\mathbf{G}^{k-1}$:

$$\mathbf{H}^{k} = \mathrm{CondChain}(\mathbf{H}, \mathrm{Embed}(\mathbf{G}^{k-1})),\ k = 1, \ldots, K, \tag{6.2}$$

where $\mathbf{G}^{k-1}$, obtaining from the $\text{Encoder}_{\text{Rec}}$ output for previous speaker, can be viewed as the speaker condition. The Embed module is a multi-layer fully connected layer aiming to project the linguistic sequence $\mathbf{G}^{k-1}$ into the acoustic sub-space. In the first step, an all-zero vector will be initialized as the speaker condition. Besides, the long short-term memory (LSTM) layer also helps to provide all historic speaker conditions by the flowing states. With this design, we can successfully perform a NAR computation in each step and the total inference steps is a constant number equaling to the number of mixed speakers. Moreover, compared with other multi-speaker ASR methods, which have to fix the number of mixed speakers in the training data, our model can handle variable mixed data and further improve the performance. Algorithm 1 outlines the training procedure of our proposed model.

```
Algorithm 1: Training procedure of our model
Initialize the model parameters $\boldsymbol{\theta}$ and a all-zero condition $\mathbf{G}^0$ for the first step ;
Given hyper parameters: learning rate $\alpha$, InterCTC loss weight $\lambda$ ;
Loading pre-trained model or not ;
while $Epoch < TotalEpoch$ do
    Given the input mixture speech waveform $\mathbf{x} = \{x_1, \dots, x_T\}$ and the corresponding
      transcriptions $\mathbf{Y} = \{\mathbf{Y}^1, \dots, \mathbf{Y}^K\}$ of $K$ different speakers ;
    Forward the Encoder$_{\text{mix}}$ with $\mathbf{x}$ and obtain the mixture hidden representations of $\mathbf{H}$
      using Eq. (5.1);
    for $(k = 1; k < K; k{+}{+})$ do
        Concatenate the $\mathbf{H}$ with previously-estimated condition $\mathbf{G}^{k-1}$ and forward the
          LSTM layer as in Eq. (6.2);
        Forward the Encoder$_{\text{rec}}$ with the output of LSTM layer;
        The output of Encoder$_{\text{rec}}$ is used to compute the Loss$_{\text{CTC}}$ as well as determine the
          best permutation of transcriptions as in Eq. (5.4);
        The output of the intermediate layer in Encoder$_{\text{rec}}$ is used to compute the
          Loss$_{\text{InterCTC}}$ with above best transcription permutations;
        $\mathbf{G}^i$ will also be regarded as the condition for the prediction of the next speaker;
    end
    Update the model using Eq. (6.1);
    Epoch = Epoch + 1;
end
return $\boldsymbol{\theta}$
```

6.3 Experiments

Setup

The proposed models are evaluated on two commonly used simulated multi-speaker speech datasets which have been described in Sec. 1.4.3.

WSJ0-Mix. The dataset can be divided into two categories, namely the 2-speaker scenario and 3-speaker scenario. In the 2-speaker scenario, we use the common benchmark called WSJ0-2mix dataset introduced by (Hershey et al., 2016a) with a sampling rate of 16 KHz. The training and validation sets are generated by randomly selecting two utterances from different speakers from the WSJ0 si_tr_s partition, containing around 30 h and 10 h speech mixture, respectively. To mix the utterances, various signal-to-noise ratios (SNRs) are uniformly chosen from [0, 10] dB. For the test set, the mixture is similarly generated using utterances from the WSJ0 validation set si_dt_05 and evaluation set si_et_05, resulting in 5 h speech mixtures. For the 3-speaker experiments, similar methods are adopted except the number of speakers is three.

LibriMix. Our methods are additionally tested on LibriMix, a recent open-source dataset for multi-speaker speech processing. The LibriMix data is created by mixing the source utterances randomly chosen from different speakers in LibriSpeech (Panayotov et al., 2015) and the noise samples from WHAM! (Wichern et al., 2019). The SNRs of the mixtures are normally distributed with a mean of 0 dB and a standard deviation of 4.1 dB. LibriMix is composed of 2-speaker or 3-speaker mixtures, with or without noise conditions. For fast evaluation, we conducted our experiments on the train-100 subset from Libri2Mix, which contains around 100 h of 2-speaker mixture speech.

All the proposed models are implemented with ESPnet (Watanabe et al., 2018). We followed the ESPnet recipe to set the hyper-parameters of the model. For all Transformer- and Conformer-based models, EncoderMix is comprised of two CNN blocks and EncoderRec contains 8 Transformer or Conformer layers, depending on the model choices. For non-conditional chain models, the EncoderSD is a 4-layer Transformer or Conformer network, while the CondChain is a 1-layer LSTM network with 1024 hidden units. The common parameters of the Transformer and Conformer layers are: $d^{\text{head}} = 4, d^{\text{att}} = 256, d^{\text{ff}} = 2048$ for the number of heads, dimension of attention module, and dimension of feed-forward layer, respectively.

Results on WSJ0-Mix

In this part, we present the performance on the WSJ0-Mix corpus, which is shown in Table 6.1. To evaluate the effectiveness, we compare our conditional speaker chain based Conformer CTC model with a variety of systems including the hybrid systems, PIT-based end-to-end AR and NAR models, and conditional speaker chain based Transformer models. Since all PIT-based models are unable to deal with variable numbers of speakers, only the results of 2-speaker scenario are presented. To make a fair comparison with NAR methods, the end-to-end AR models are decoded only with greedy search.

For the PIT-based AR models, PIT-Conformer (5) shows the best performance, achieving a word error rate (WER) of 22.4% on the WSJ0-2mix test set. When comparing the NAR models, PIT-Transformer-CTC (6), which is only trained with CTC loss, suffers a dramatic performance degradation (50.3%). There is no doubt that a pure CTC based encoder network can hardly model different speaker's speech simultaneously. When applying the conditional speaker chain based method, both model (7) and model (8) are better than PIT model. By combining the single and multi-speaker mixture speech, model (8) shows a significant improvement, whose WER is 29.5% on the WSJ0-2mix test set. For our conditional Conformer-CTC model (9), we explore two types of conditional features, including the “hard" CTC alignments and “soft" latent features after $\text{Encoder}_{\text{Rec}}$. Both approaches are better than above models with only a $\sim$0.07 seconds increase of latency and applying the “soft" features achieves a WER of 24.4%. By incorporating the interme-

Table 6.1: Word error rates (WERs) and real time factor (RTF) for multi-speaker speech recognition on WSJ0-Mix dataset. The RTF results are obtained by averaging the results of 5 decoding processes on CPUs.

Models	**Training Data**	**WER (%)**		**RTF**
		WSJ0-2mix	**WSJ0-3mix**	
Hybrid model (w/ beam search)				
(1) PIT-DNN-HMM (Qian et al., 2018a)	WSJ0-2mix	28.2	-	-
(2) DPCL + DNN-HMM (Menne et al., 2019)	WSJ0-2mix	16.5	-	-
E2E Autoregressive Model (w/ greedy search)				
(3) PIT-RNN (Chang et al., 2019a)†	WSJ0-2mix	51.4	-	1.4293
(4) PIT-Transformer (Shi et al., 2020a)†	WSJ0-2mix	37.0	-	1.4695
(5) PIT-Conformer	WSJ0-2mix	22.4	-	1.3970
E2E Non-autoregressive Model (w/ greedy search)				
(6) PIT-Transformer-CTC	WSJ0-2mix	50.3	-	0.1091
(7) Conditional-Transformer-CTC (Shi et al., 2020a)†	WSJ0-2mix	41.0	-	0.1293
(8) Conditional-Transformer-CTC (Shi et al., 2020a)†	WSJ0-1&2&3mix	29.4	53.3	-
(9) Conditional-Conformer-CTC	WSJ0-2mix	25.3	-	0.1824
+ hidden feature conditions	WSJ0-2mix	24.4	-	0.1758
+ InterCTC loss	WSJ0-2mix	22.3	-	0.1854
(10) Conditional-Conformer-CTC	WSJ0-1&2&3mix	23.4	39.1	0.1771 / 0.2096
+ hidden feature conditions	WSJ0-1&2&3mix	22.2	38.6	0.1741 / 0.2241
+ InterCTC loss	WSJ0-1&2&3mix	**19.9**	**34.3**	0.1732 / 0.2088

†: The results are obtained by the same implementation in (Shi et al., 2020a) but w/o beam search and LM rescoring. When using both beam search and LM rescoring, the results are 14.9% / 37.9% of model (8) and 12.4% / 26.6% of model (10).

Table 6.2: Word error rates (WERs) for multi-speaker speech recognition on LibriMix dataset.

Models	**Dev**	**Test**
E2E Autoregressive Model (w/ greedy search)		
(1) PIT-Transformer	34.8	36.0
E2E Non-autoregressive Model (w/ greedy search)		
(2) PIT-Transformer-CTC	45.2	45.9
(3) Conditional-Transformer-CTC	32.7	33.3
(4) Conditional-Conformer-CTC + both	**24.5**	**24.9**

Table 6.3: Correlation between the hypothesis (Hyp.) generation order and the source signal (Src.) length order on WSJ0-2mix.

Src. **Hyp.**	long	short
1st output	2749	251
2nd output	251	2749

diate loss, we can obtain a superior WER of 22.3%, reaching a strong AR PIT-Conformer model (5). However, after combining latent feature conditions and the intermediate CTC loss, we don't get a further improvement. Finally, we also train our model on the data of variable numbers of speakers and obtain the best WERs of 19.9% and 34.3%, which are even better than model (5) with only 1/7 latency.

We further investigate the correlation between the hypothesis generation order and the source signal length (from long to short), as shown in Table 6.3. We find that only 251/3000 utterances do not follow the order on 2-speaker scenario and the average Spearman's Coefficient is 0.833.

Results on LibriMix

The results on LibriMix are summarized in Table 6.2. From the table, we can see a quite similar trend as the WSJ0-Mix results in the previous subsection. Our Conditional-Conformer-CTC with both latent features conditions and intermediate CTC loss obtains the best WERs of 24.5% and 24.9% on dev and test sets, respectively, which yields up to 25% relative improvement compared with the Conditional-Transformer-CTC model.

6.4 Conclusions

In this chapter, we revisit our proposed conditional speaker chain based multi-speaker ASR by enhancing the NAR ability. Our improved model mainly includes a conditional speaker chain (CondChain) module and Conformer CTC based encoders. To boost the performance of a pure Conformer CTC encoder, we also investigate two approaches, which are using the “soft" latent features from the encoder output as speaker conditions and including an additional intermediate CTC loss. We evaluate the effectiveness of our model on two multi-speaker benchmarks, WSJ0-Mix and LibriMix. Our model shows consistent improvement over other models with only a slight increment of RTF and even better than a strong AR model in some cases.

While the conditional chain model successfully removes the requirement of knowing the number of speakers in the overlapping speech, it still relies on the assumption that the input speech is mostly overlapped, eliminating the need to detect when the overlap starts. In the next chapter, we will address this limitation by investigating methods to handle scenarios where the overlap onset is not predetermined, further enhancing the flexibility and applicability of our multi-speaker ASR system.

Chapter 7

GTCe: monaural multi-speaker E2E ASR towards real speech overlaps

Summary

In the previous chapter (Ch. 5), we introduced an E2E ASR model tailored for recognizing speech signals featuring multiple overlapping speakers. However, the model operated under specific constraints to simplify the task, which may not fully reflect the complexities of real-world scenarios. Firstly, it assumed a fixed number of speakers in the output, limiting its flexibility in accommodating varying speaker counts. Secondly, it presumed that speech overlap begins from the beginning of the utterance, which may not align with sparse speech overlapping patterns encountered in practice. In Ch. 6, we addressed the first constraint by introducing a more flexible model capable of handling varying speaker counts in overlapping speech scenarios. However, sequentially generating transcriptions for each speaker proved computationally intensive and did not fully capture the nuances of speech overlaps.

In this chapter, we present a novel non-modular approach designed to handle multi-speaker overlapping speech with a single output sequence. Our model assumes that tokens (e.g., subwords) from multiple speakers are sparsely distributed and ordered by activation time, reflecting real-world speech overlap patterns. To achieve this, we propose employing an extended Graph-based Temporal Classification (GTC-e) loss, which enables us to train two distinct predictions—one for speakers and one for ASR outputs—aligned at the frame level. This innovative approach represents a significant step towards improving the flexibility and accuracy of E2E ASR models in transcribing complex, multi-speaker speech with overlapping segments.

Xuankai Chang, Niko Moritz, Takaaki Hori, Shinji Watanabe, Jonathan Le Roux. ICASSP 2022. Extended Graph Temporal Classification for Multi-Speaker End-to-End ASR.

7.1 Introduction

In recent years, dramatic progress has been achieved in automatic speech recognition (ASR), in particular thanks to the exploration of neural network architectures that improve the robustness and generalization ability of ASR models (Qian et al., 2016; Graves et al., 2013; Vaswani et al., 2017; Gulati et al., 2020; Guo et al., 2021a). The rise of end-to-end ASR models has simplified ASR architecture with a single neural network, with frameworks such as the connectionist temporal classification (CTC) (Graves et al., 2006), attention-based encoder-decoder model (Chan et al., 2016; Kim et al., 2017b; Watanabe et al., 2017), and the RNN-Transducer model (Graves, 2012).

Graph modeling has traditionally been used in ASR for decades. For example, in hidden Markov model (HMM) based systems, a weighted finite-state transducer (WFST) is used to combine several modules together including a pronunciation lexicon, context-dependencies, and a language model (LM) (Mohri et al., 2002; Hori et al., 2007). Recently, researchers proposed to use graph representations in the loss function for training deep neural networks (Hannun et al., 2020). In (Moritz et al., 2021), a new loss function, called graph-based temporal classification (GTC), was proposed as a generalization of CTC to handle sequence-to-sequence problems. GTC can take graph-based supervisory information as an input to describe all possible alignments between an input sequence and an output sequence, for learning the best possible alignment from the training data. As an example of application, GTC was used to boost ASR performance via semi-supervised training (Lamel et al., 2002; Huang et al., 2013) by using an N-best list of ASR hypotheses that is converted into a graph representation to train an ASR model using unlabeled data. However, in the original GTC, only posterior probabilities of the ASR labels are trained, and trainable label transitions are not considered. Extending GTC to handle label transitions would allow us to model further information regarding the labels. For example, in a multi-speaker speech recognition scenario, where some overlap between the speech signals of multiple speakers is considered, we could use the transition weights to model speaker predictions that are aligned with the ASR label predictions at frame level, such that when an ASR label is predicted we can also detect if it belongs to a specific speaker. Such a graph is illustrated in Fig. 7.1.

In the last few years, several multi-speaker end-to-end ASR models have been proposed. In (Seki et al., 2018; Chang et al., 2019a), permutation invariant training (PIT) (Hershey et al., 2016a; Isik et al., 2016; Yu et al., 2017b) was used to compute the loss by choosing the hypothesis-reference assignment with minimum loss. In (Kanda et al., 2020b), an attention-based encoder-

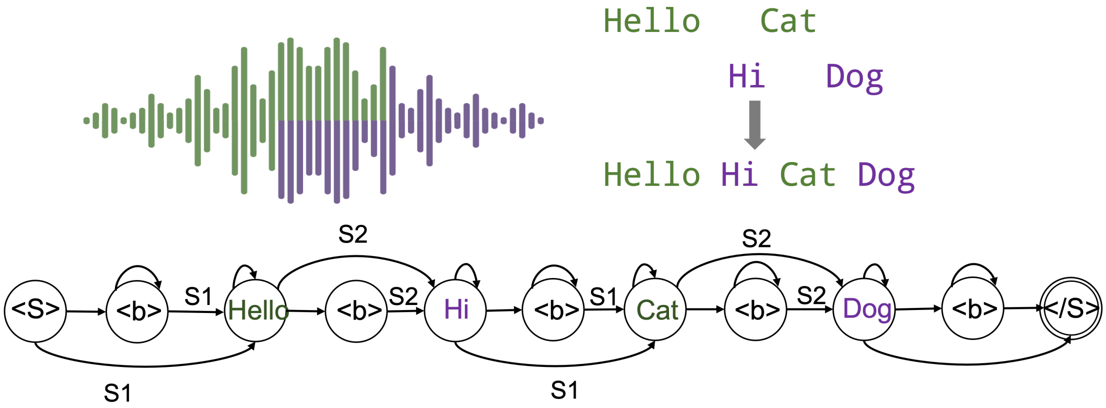


Figure 7.1: Illustration of a GTC-e graph for multi-speaker ASR. In the graph, the nodes represent the tokens (words) from the transcriptions. The edges indicate the speaker transitions.

decoder is trained to generate the hypothesis sequences of different speakers in a predefined order based on heuristic information, a technique called serialized output training (SOT). In (Shi et al., 2020a; Guo et al., 2021b), the model is trained to predict the hypothesis sequence of one speaker in each iteration while utilizing information about the previous speakers' hypotheses as additional input. These existing multi-speaker end-to-end ASR models, which have showed promising results, all share a common characteristic in the way that the predictions can be divided at the level of a whole utterance for each speaker. For example, in the PIT-based methods, label sequences for different speakers are supposed to be output at different output heads, while in the SOT-/conditional-based models, the prediction of the sequence for a speaker can only start when the sequence of the previous speaker completes.

In contrast to previous works, in this chapter, the multi-speaker ASR problem is not implicitly regarded as a source separation problem using separate output layers for each speaker or cascaded processes to recognize each speaker one after another. Instead, the prediction of ASR labels of multiple speakers is regarded as a sequence of acoustic events irrespective of the source shown as in Fig. 7.1, and the belonging to a source is predicted separately to distinguish if an ASR label was uttered by a given speaker. We propose to use an extended GTC (GTC-e) loss to accomplish this, which allows us to train two separate predictions, one for the speakers and one for the ASR outputs, that are aligned at the frame level. In order to exploit the speaker predictions efficiently during decoding, we also modify an existing frame-synchronous beam search algorithm to adapt it to GTC-e. The proposed model is evaluated on a multi-speaker end-to-end ASR task based on the LibriMix data, including various degrees of overlap between speakers. This work proposes a novel approach to address multi-speaker ASR by considering the ASR outputs of multiple speakers as a sequence of intermingled events with a chronologically meaningful ordering.

7.2 Extended GTC Algorithm

In this section, we describe the extended GTC loss function. For the convenience of understanding, we mostly follow the notations in the previous GTC study (Moritz et al., 2021).

GTC was proposed as a loss function to address sequence-to-sequence problems. We assume the input of the neural network is the mixture audio waveform $\mathbf{x}$. We denote it as $\mathbf{x} = (x_1, \ldots, x_T)$, where T stands for the length. The output is a sequence of length L, $\hat{Y} = (\hat{\mathbf{y}}^1, \ldots, \hat{\mathbf{y}}^L)$, where $\hat{\mathbf{y}}^l$ denotes the posterior probability distribution over an alphabet $\mathcal{V}$, and the j-th class's probability is denoted by $\hat{y}^l_j$. We use $\mathcal{G}$ to refer to a graph constructed from references. Then the GTC function computes the posterior probability for graph $\mathcal{G}$ by summing over all alignment sequences in $\mathcal{G}$:

$$p(\mathcal{G}|\mathbf{x}) = \sum_{\pi \in \mathcal{S}(\mathcal{G},L)} p(\pi|\mathbf{x}), \tag{7.1}$$

where $\mathcal{S}$ represents a search function that unfolds $\mathcal{G}$ to all possible node sequences of length L (not counting non-emitting start and end nodes), π denotes a single sequence of nodes, and $p(\pi|\mathbf{x})$ is the posterior probability for π given the input $\mathbf{x}$. The loss function is defined as the following negative log likelihood:

$$\mathcal{L} = -\ln p(\mathcal{G}|\mathbf{x}). \tag{7.2}$$

Following (Moritz et al., 2021), we index the nodes of graph $\mathcal{G}$ using $g = 0, \ldots, G+1$, sorting them in a breadth-first search manner from 0 (non-emitting start node) to $G+1$ (non-emitting end node). We denote by $v(g) \in \mathcal{V}$ the output symbol observed at node g, and by $W_{(g,g')}$ a deterministic transition weight on edge (g, g'). In addition, we denote by $\pi_{l:l'} = (\pi_l, \ldots, \pi_{l'})$ the node sub-sequence of π from time index l to l'. Note that π_0 and π_{L+1} correspond to the non-emitting start and end nodes 0 and $G+1$, respectively.

We modify GTC such that the neural network can generate an additional posterior probability distribution, $\omega^l_{I(g,g')}$, representing a transition weight on edge (g, g') at time l, where $I(g, g') \in \mathcal{I}$ and $\mathcal{I}$ is the index set of all possible transitions. The posterior probabilities are obtained as the output of a softmax. The forward probability, $\alpha_l(g)$, represents the total probability at time l of the sub-graph $\mathcal{G}_{0:g}$ of $\mathcal{G}$ containing all paths from node 0 to node g. It can be computed for $g = 1, \ldots, G$ using

$$\alpha_l(g) = \sum_{\substack{\pi \in \mathcal{S}(\mathcal{G},L): \\ \pi_{0:l} \in \mathcal{S}(\mathcal{G}_{0:g},l)}} \prod_{\tau=1}^{l} W_{\pi_{\tau-1},\pi_\tau} \omega^{\tau}_{I(\pi_{\tau-1},\pi_\tau)} y^{\tau}_{v(\pi_\tau)}. \tag{7.3}$$

Note that $\alpha_0(g)$ equals 1 if g corresponds to the start node and it equals 0 otherwise. The backward probability $\beta_l(g)$ is computed similarly, using

$$\beta_l(g) = \sum_{\substack{\pi \in \mathcal{S}(\mathcal{G},L): \\ \pi_{l:L+1} \in \mathcal{S}(\mathcal{G}_{g:G+1}, L-l+1)}} \left[y^L_{v(\pi_L)} \prod_{\tau=l}^{L-1} W_{\pi_\tau,\pi_{\tau+1}} \omega^{\tau+1}_{I(\pi_\tau,\pi_{\tau+1})} y^\tau_{v(\pi_\tau)} \right], \tag{7.4}$$

where $\mathcal{G}_{g:G+1}$ denotes the sub-graph of $\mathcal{G}$ containing all paths from node g to node $G+1$. Similar to GTC or CTC, the computation of α and β can be efficiently performed using the forward-backward algorithm.

The network is optimized by gradient descent. The gradients of the loss with respect to the label posteriors y^l_j and to the corresponding unnormalized network outputs h^t_j before the softmax is applied, for any symbol $j \in \mathcal{V}$, can be obtained in the same way as in CTC and GTC, where the key idea is to express the probability function $p(\mathcal{G}|\mathbf{x})$ at l using the forward and backward variables:

$$p(\mathcal{G}|\mathbf{x}) = \sum_{g \in \mathcal{G}} \frac{\alpha_l(g)\beta_l(g)}{y^l_{e(g)}}. \tag{7.5}$$

The derivation of the gradient of the loss with respect to the network outputs for the transition probabilities w^l_i, for a transition $i \in \mathcal{I}$, is similar but with some important differences. Here, the key is to express $p(\mathcal{G}|\mathbf{x})$ at l as

$$p(\mathcal{G}|\mathbf{x}) = \sum_{(g,g') \in \mathcal{G}} \alpha_{l-1}(g) W_{g,g'} \omega^l_{I(g,g')} \beta_l(g'). \tag{7.6}$$

The derivative of $p(\mathcal{G}|\mathbf{x})$ with respect to the transition probabilities ω^l_i can then be written as

$$\frac{\partial p(\mathcal{G}|\mathbf{x})}{\partial \omega^l_i} = \sum_{(g,g') \in \Phi(\mathcal{G},i)} \alpha_{l-1}(g) W_{g,g'} \beta_l(g'), \tag{7.7}$$

where $\Phi(\mathcal{G}, i) = \{(g, g') \in \mathcal{G} : I(g, g') = i\}$ denotes the set of edges in $\mathcal{G}$ that correspond to transition i. To backpropagate the gradients through the softmax function of w^l_i, we need the derivative with respect to the unnormalized network outputs h^l_i before the softmax is applied, which is

$$-\frac{\partial \ln p(\mathcal{G}|\mathbf{x})}{\partial h^l_i} = -\sum_{i' \in \mathcal{I}} \frac{\partial \ln p(\mathcal{G}|\mathbf{x})}{\partial \omega^l_{i'}} \frac{\partial \omega^l_{i'}}{\partial h^l_i}. \tag{7.8}$$

The gradients for the transition weights are derived by substituting (7.7) and the derivative of the softmax function $\partial\omega^l_{i'}/\partial h^l_i = \omega^l_{i'}\delta_{ii'} - \omega^l_{i'}\omega^l_k$ into (7.8):

$$-\frac{\partial \ln p(\mathcal{G}|\mathbf{x})}{\partial h^l_i} = \omega^l_i \frac{\omega^l_i}{p(\mathcal{G}|\mathbf{x})} \sum_{(g,g')\in\Phi(\mathcal{G},i)} \alpha_{l-1}(g) W_{g,g'} \beta_l(g'). \tag{7.9}$$

We used the fact that

$$\begin{aligned} -\sum_{i'\in\mathcal{I}} \frac{\partial \ln p(\mathcal{G}|\mathbf{x})}{\partial \omega^l_{i'}} \omega^l_{i'}\delta_{ii'} &= -\frac{\partial \ln p(\mathcal{G}|\mathbf{x})}{\partial \omega^l_i}\omega^l_i, \\ &= -\frac{\omega^l_i}{p(\mathcal{G}|\mathbf{x})} \sum_{(g,g')\in\Phi(\mathcal{G},i)} \alpha_{l-1}(g) W_{g,g'} \beta_l(g'), \end{aligned} \tag{7.10}$$

and that

$$\begin{aligned} &\sum_{i'\in\mathcal{I}} \frac{\partial \ln p(\mathcal{G}|\mathbf{x})}{\partial \omega^l_{i'}} \omega^l_{i'}\omega^l_i \\ &= \sum_{i'\in\mathcal{I}} \frac{\omega^l_{i'}\omega^l_i}{p(\mathcal{G}|\mathbf{x})} \sum_{(g,g')\in\Phi(\mathcal{G},i')} \alpha_{l-1}(g) W_{g,g'} \beta_l(g'), \\ &= \frac{\omega^l_i}{p(\mathcal{G}|\mathbf{x})} \sum_{i'\in\mathcal{I}} \sum_{(g,g')\in\Phi(\mathcal{G},i')} \alpha_{l-1}(g) W_{g,g'} \omega^l_{i'} \beta_l(g'), \\ &= \frac{\omega^l_i}{p(\mathcal{G}|\mathbf{x})} \sum_{(g,g')\in\mathcal{G}} \alpha_{l-1}(g) W_{g,g'} \omega^l_{I(g,g')} \beta_l(g'), \\ &= \frac{\omega^l_i}{p(\mathcal{G}|\mathbf{x})} p(\mathcal{G}|\mathbf{x}) = \omega^l_i. \end{aligned} \tag{7.11}$$

For efficiency reason, we implemented the GTC objective in CUDA as an extension for PyTorch.

7.3 E2E SIMO ASR with GTCe

We apply the extended GTC approach to multi-speaker ASR, which is considered as a challenging task in the field of speech processing. One of the main difficulties of multi-speaker ASR stems from the necessity to find a way to train a network that will be able to reliably group tokens from the same speaker together. Most existing approaches attempt to handle this problem either by splitting the speakers across multiple outputs (Yu et al., 2017b; Chang et al., 2019a) or by making predictions sequentially speaker by speaker (Kanda et al., 2020b; Shi et al., 2020a; Guo et al., 2021b). The ambiguity in how to assign a given output to a given reference at training time is typically broken either by using permutation invariant training or by using an arbitrary criterion such as assigning an output to the speaker with highest energy or with the earliest onset. We here take a completely different approach, motivated by our noticing that a graph can be a good representation for overlapped speech, since it can represent the tokens at each node while the speaker identity can also be labeled at each edge. More specifically, given the transcriptions of all the speakers in an overlapped speech, we can convert them to a sequence of chronologically ordered linguistic tokens where each token has a speaker identity. The temporal alignment of tokens can be acquired by performing CTC alignment on each isolated clean speech, which is like a sequence of sparse spikes, as shown in Fig. 7.2, and merging them based on their time occurrence. Note here that this assumes that the activation period of linguistic tokens from different speakers are not completely the same. In practice, this condition is often satisfied, although overlaps do occur in some small percentage of frames. Based on this, we can construct a graph for multi-speaker ASR for each overlapped speech mixture. We show a simple example graph in Fig. 7.1. In this setup, the alphabet $\mathcal{V}$ for the node labels consists of all the ASR tokens, and the set of transitions $\mathcal{I}$ consists of the speaker indices up to the maximum number of speakers.

As in GTC(Moritz et al., 2021), we can apply a beam search algorithm during decoding. Since the output of GTC-e contains tokens from multiple speakers, we need to make modifications to the existing time-synchronous prefix beam search algorithm (Hannun et al., 2014b; Moritz et al., 2019). The modified beam search is shown in Algorithm 2. The main modifications are three fold. First, we apply the speaker transition probability in the score computation. Second, when expanding the prefixes, we need to consider all possible speakers. Third, when computing the LM scores of a prefix, we need to consider the sub-sequences of different speakers separately.

7.4 Experiment

Setup

We mainly use the LibriMix (Cosentino et al., 2020) data to conduct experiments. LibriMix, as described in Sec. 1.4.3, contains multi-speaker overlapped speech simulated by mixing utterances randomly chosen from different speakers in the LibriSpeech corpus (Panayotov et al., 2015). For fast adaption, we use the 2-speaker train_clean_100 subset of LibriMix. The original LibriMix dataset generates fully overlapped speech by default, which means that one utterance is 100% interfered by the other (assuming they have the same length). However, in realistic conditions, the overlap ratio is usually small (Çetin and Shriberg, 2006; Chen et al., 2020). To simulate such conditions, we use the same utterance selections and signal to noise ratio (SNR) as in LibriMix with smaller overlapping ratios of 0% and 40% to generate additional training data subsets.

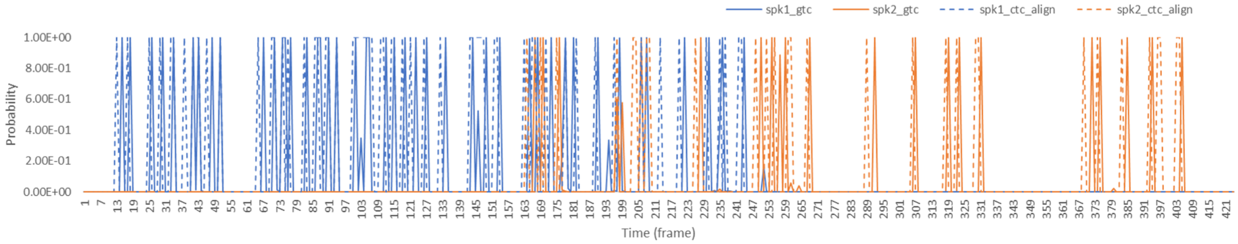


Figure 7.2: An example of speaker transition posterior predicted by GTC. The input 2-speaker utterance's overlap ratio is about 40%. The figure shows the predicted (solid line) and ground truth (dashed line) activations.

For labels, we use the linguistic token sequence of all the speakers in the mixture. First, we generate the token alignments given each source utterance based on the Viterbi alignment of CTC, which indicates the rough activation time of every token. Then we combine the alignments of two speakers by ordering the tokens monotonically along the time axis. In order to reduce the concurrent activations of tokens from different speakers, we make use of byte pair encodings (BPE) as our token units. In our experiments, we use the BPE model with 5000 tokens trained on LibriSpeech data. The concurrent activations of tokens for two speakers are relatively rare, at the rate of 6% and 2% on fully and 40% overlapping ratio training subsets respectively. When these concurrent activations occur, we use a predefined order which makes the label from the speaker with highest energy over the whole utterance come first (allowing multiple permutations in the label graph will be considered in future work).

For ASR models, we simply reused the encoder architecture in PIT-based multi-speaker end-to-end speech recognition models, for the details of which we shall refer the reader to (Chang et al., 2019a). In the model, there are 2 CNN blocks to encode the input acoustic feature, followed by 2 sub-networks each of which has 4 Transformer layers to extract the token and speaker information, respectively. Then 8 shared Transformer layers are used to convert each of the two sequences to some representation. For the two output sequences, one is regarded as token hidden representation and the other one is regarded as speaker prediction. We use a normal single-speaker ASR model trained with CTC (single-speaker CTC) and the original end-to-end PIT-ASR model (Chang et al., 2019a) trained with CTC loss only (PIT-CTC) as our baselines.

Greedy search results

In this section, we describe the ASR performance of the baselines and the proposed GTC-e model using greedy search decoding. The word error rates (WERs) are shown in Table 7.1. From the table, we can see that the proposed model is better than the normal ASR model. Our proposed model also achieves a performance close to the PIT-CTC model, especially in the low-overlap ratio cases

(0%, 20%, 40%). Note that although our model predicts the speaker indices, there exists speaker prediction errors. We further check the oracle token error rates (TER) of PIT-CTC and GTC-e, by only comparing the tokens from all output sequences against all reference sequences, regardless of speaker assignment. As shown in Table 7.2, we obtain averaged test TERs for PIT-CTC and GTC-e of 22.8% and 25.0% respectively, from which we can tell that the token recognition performance is comparable. It indicates that we should consider how to improve the speaker prediction in the next step.

We also show an example of CTC ground truth token alignment together with the speaker transition posterior predictions by our model in Fig. 7.2. From the figure, we can see that our GTC-e model can accurately predict the activations of most tokens.

Beam Search Results

We here present the ASR performance of beam search decoding, shown in Table 7.3. For the language model, we use a 16-layer Transformer-based LM trained on full LibriSpeech data with external text. The beam size of GTC-e is set to 40, while that of PIT-CTC is cut to half to keep the average beam size of every speaker the same. With the beam search, the word error rates are greatly improved. Our approach obtains promising results which are close to the PIT-CTC baseline, albeit with a slightly worse WER. In addition to the average WERs, the WERs for each speaker are also shown in Table 7.4, confirming that the model is not biased towards a particular speaker output.

7.5 Conclusion

In this chapter, we propose GTC-e, an extension of the graph-based temporal classification method using neural networks to predict posterior probabilities for both labels and label transitions. This extended GTC framework opens the way to a wider range of applications. As an example application, we explored the use of GTC-e for multi-speaker end-to-end ASR, a notably challenging task, leading to a multi-speaker ASR system that transcribes speech in a very similar way to single-speaker ASR. We have performed preliminary experiments on the LibriMix 2-speaker dataset, showing promising results demonstrating the feasibility of the approach.

The GTC-e method has proven effective in addressing partially overlapping speech. Due to the sparse activation of tokens, it also shows potential for handling a flexible number of speakers without requiring prior knowledge of the number of speakers, thus eliminating initial assumptions. This concludes our investigation into the Single-Input Multi-Output (SIMO) scenario. In the next part, we will delve into the ASR of Multi-Input Multi-Output (MIMO). MIMO can lead to better speech recognition accuracy and thus wider applications in real scenraio.

Algorithm 2: The modified time-synchronous prefix beam search for extended GTC. We use A_{prev} to store every prefix l at every time step. We denote the alphabet by $\mathcal{V}$ and number of speakers by K. We denote the symbol posterior by $p(\cdot)$ and the speaker transition posterior by $p^{\omega}(\cdot)$.

```
$\ell \leftarrow ((\langle sos \rangle, 0),);$
$p_b(\ell) \leftarrow 1, p_{nb}(\ell) \leftarrow 0;$
$A_{\text{prev}} \leftarrow \{\ell\};$
for $t$=$1,\ldots,T$ do
    $A_{\text{next}} \leftarrow \{\};$
    for $\ell$ in $A_{prev}$ do
        for $c$ in $\mathcal{V}$ do
            if $c = blank$ then
                $p_b(\ell) \leftarrow p(\text{blank}; x_t)p^{\omega}(\text{blank}; x_t)(p_b(\ell; x_{1:t-1}) + p_{nb}(\ell; x_{1:t-1}));$
                add $\ell$ to $A_{\text{next}}$;
            else
                for $s = 1, \ldots, K$ do                                    ▷ *Loop over speaker index
                    $\ell^+ \leftarrow$ append $(c, s)$ to $\ell$ ;
                    if $(c, s) = \ell_{end}$ then
                        $p_{nb}(\ell^+; x_{1:t}) \leftarrow p(c; x_t)p_b(\ell; x_{1:t-1})p^{\omega}(s; x_t);$
                        $p_{nb}(\ell; x_{1:t}) \leftarrow p(c; x_t)p_{nb}(\ell; x_{1:t-1})p^{\omega}(s; x_t);$
                    else
                        $p_{nb}(\ell^+; x_{1:t}) \leftarrow p(c; x_t)(p_b(\ell; x_{1:t-1}) + p_{nb}(\ell; x_{1:t-1}))p^{\omega}(s; x_t);$
                    end
                    if $\ell^+$not in$A_{prev}$ then
                        $p_b(\ell^+; x_{1:t}) \leftarrow$
                          $p(\text{blank}; x_t)(p_b(\ell^+; x_{1:t-1}) + p_{nb}(\ell^+; x_{1:t-1}))p^{\omega}(\text{blank}; x_t)$ ;
                        $p_{nb}(\ell^+; x_{1:t}) \leftarrow p(c; x_t)p_{nb}(\ell^+; x_{1:t-1}) \cdot p^{\omega}(s; x_t);$
                    end
                    add $\ell^+$ to $A_{\text{next}}$
                end
            end
        end
    end
    $A_{\text{prev}} \leftarrow B$ most probable prefixes in $A_{\text{next}}$ ▷ Track the LM scores of different speakers
      separately.
end
```

Table 7.1: WER(%) comparison between baselines and the GTC-e model using greedy search decoding.

	0% overlap		20% overlap		40% overlap		Full overlap	
Model	dev	test	dev	test	dev	test	dev	test
single-speaker CTC	34.6	34.1	37.4	37.0	45.9	45.3	76.3	75.9
PIT-CTC	18.8	19.2	19.9	22.3	22.9	23.5	32.9	33.8
GTC-e	20.5	21.1	22.6	23.3	26.3	27.3	44.6	45.8

Table 7.2: Oracle TER(%) comparison between PIT-CTC and GTC-e.

	0% overlap		20% overlap		40% overlap		Full overlap		Average	
Model	dev	test	dev	test	dev	test	dev	test	dev	test
PIT-CTC	18.5	18.4	19.4	19.5	22.0	22.4	30.1	30.9	22.5	22.8
GTC-e	19.8	20.1	21.1	21.4	24.1	24.6	33.4	33.9	24.6	25.0

Table 7.3: WER(%) comparison between PIT-CTC and GTC-e using beam search decoding.

	0% overlap		20% overlap		40% overlap		Full overlap	
Model	dev	test	dev	test	dev	test	dev	test
PIT-CTC	11.7	12.4	12.6	13.4	16.3	18.1	24.0	26.3
GTC-e	14.8	15.5	16.5	17.2	19.5	20.4	32.7	33.7

Table 7.4: WER(%) for each speaker with GTC-e using beam search decoding.

	0% overlap		20% overlap		40% overlap		Full overlap	
Speaker	dev	test	dev	test	dev	test	dev	test
spk1	15.0	15.1	17.0	17.3	20.6	21.1	33.0	33.7
spk2	14.7	15.7	15.9	17.1	18.4	19.7	32.3	33.7

Part IV

E2E-ASR for Multi-channel-Input Multi-speaker-Output (MIMO)

Chapter 8

MIMO-Speech-RNN: Multi-channel multi-speaker E2E ASR

Summary

In this part, we address the challenging problem of multi-channel multi-speaker speech recognition, which has not been explored in previous chapters. Our focus shifts to introducing a novel end-to-end model called MIMO-Speech designed specifically for this task. The MIMO-Speech model is tailored to process speech signals captured by a microphone array and generate corresponding text sequences for each individual speaker present in the input. The overview of the model is shown in Fig. 8.1. To effectively address the speech separation task inherent in multi-channel multi-speaker scenarios, the model incorporates a neural beamformer as part of its front-end. Notably, our proposed approach adopts the end-to-end modular-based design and enables training without the need for an explicit signal reconstruction criterion.

The key advantage of the MIMO-Speech model lies in its differentiability, allowing optimization through an ASR loss as the target objective. By leveraging the power of neural networks, our model demonstrates promising capabilities in tackling the complex and challenging task of multi-channel multi-speaker speech recognition.

> Chang, Xuankai, Wangyou Zhang, Yanmin Qian, Jonathan Le Roux, and Shinji Watanabe. ASRU 2019. MIMO-Speech: End-to-end multi-channel multi-speaker speech recognition.

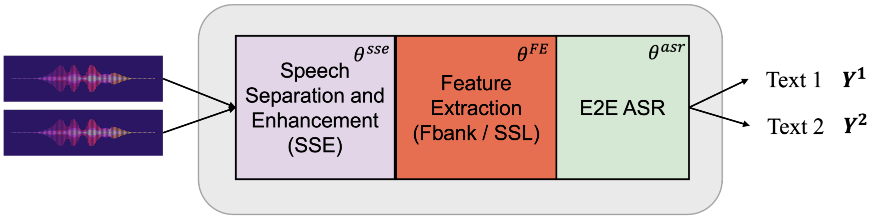


Figure 8.1: Overview of the end-to-end MIMO model.

8.1 Introduction

The cocktail party problem, where the speech of a target speaker is entangled with noise or speech of interfering speakers, has been a challenging problem in speech processing for more than 60 years (Cherry, 1953). In recent years, there have been many research efforts based on deep learning addressing the multi-speaker speech separation and recognition problems. These works can be categorized into two classes depending on the type of input signals, namely single-channel and multi-channel.

In the single-channel multi-speaker speech separation and recognition tasks, several techniques have been proposed, achieving significant progress. One such technique is deep clustering (DPCL) (Hershey et al., 2016a; Isik et al., 2016; Menne et al., 2019). In DPCL, a neural network is trained to map each time-frequency unit to an embedding vector, which is used to assign each unit to a source by a clustering algorithm afterwards. DPCL was then integrated into a joint training framework with end-to-end speech recognition in (Settle et al., 2018), showing promising performance. Another approach called permutation-free training (Hershey et al., 2016a; Isik et al., 2016) or permutation-invariant training (PIT) (Yu et al., 2017c; Kolbæk et al., 2017) relies on training a neural network to estimate a mask for every speaker with a permutation-free objective function that minimizes the reconstruction loss. PIT was later applied to multi-speaker automatic speech recognition (ASR) by directly optimizing a speech recognition loss (Yu et al., 2017a; Qian et al., 2018a) within a DNN-HMM hybrid ASR framework. In recent years, end-to-end models have drawn a lot of attention in single-speaker ASR systems and shown great success (Graves and Jaitly, 2014; Chan et al., 2016; Kim et al., 2017b; Hori et al., 2017). These models have simplified the ASR paradigm by unifying acoustic, language, and phonetic models into a single neural network. In (Seki et al., 2018; Chang et al., 2019b), joint CTC/attention-based encoder-decoder (Kim et al., 2017b) end-to-end models were developed to solve the single-channel multi-speaker speech recognition problem, where the encoder separates the mixed speech features and the attention-based decoder generates the output sequences. Although significant performance improvements have been achieved in the monaural case, there is still a large performance gap compared with that of single-speaker speech recognition systems, making such models not yet ready for widespread application in real scenarios.

The other important case is that of multi-channel multi-speaker speech separation and recognition, where the input signals are collected by microphone arrays. Acquiring multi-channel data is not so limiting nowadays, where microphone arrays are widely deployed in many devices. When multi-channel data is available, the spatial information can be exploited to determine the speaker location and to separate the speech with higher accuracy. Yoshioka et al (Yoshioka et al., 2018c) proposed a method for performing multi-channel speech separation under the PIT framework. A mask-based beamformer called the unmixing transducer was used to separate the overlapped speech. Another method proposed by Wang et al (Wang et al., 2018) leverages the inter-channel differences as spatial features combined with the single-channel spectral features as the input, to separate the multi-channel data using the DPCL technique.

Previous works based on multi-channel multi-speaker input mainly focus on separation. In this work, we propose an end-to-end multi-channel multi-speaker speech recognition system. Such a sequence-to-sequence model is trained to directly map multi-channel input (MI) speech signals where multiple speakers speak simultaneously, to multiple output (MO) text sequences, one for each speaker. We refer to this system as MIMO-Speech. The recent research on single-speaker far-field speech recognition has shown that neural beamforming techniques for denoising (Heymann et al., 2016; Erdogan et al., 2016) can achieve state-of-the art results in robust ASR tasks (Menne et al., 2016; Heymann et al., 2017; Minhua et al., 2019). Several works have shown that it is feasible to design a totally differentiable end-to-end model by integrating the neural beamforming mechanism and the sequence-to-sequence speech recognition together (Ochiai et al., 2017a; Braun et al., 2018; Wang et al., 2019; Shanmugam Subramanian et al., 2019). (Ochiai et al., 2017b) further shows that the neural beamforming function in a multi-channel end-to-end system can enhance the signals. In light of this success, we redesigned the neural-beamformer front-end to allow it to attend to multiple beams at different directions. After getting the separated signals, the log filter bank features are extracted inside the neural network. Finally, a joint CTC/attention-based encoder-decoder recognizes each feature stream. With this framework, the outputs of the beamformer in the middle of the model can also be used as speech separation signals. During the training, a data scheduling strategy using curriculum learning is specially designed and leads to an additional performance boost. To prove the basic concept of our method, we first evaluated our proposed method in the anechoic scenario. From the results, we find that even without explicitly optimizing for separation, the intermediate signals after the beamformer still show very good quality in terms of audibility. Then we also tested the model on the reverberant case to give a preliminary result.

8.2 E2E MIMO ASR

In this section, we first present the proposed end-to-end multi-channel multi-speaker speech recognition model, which is shown in Fig. 8.2. We then describe the techniques applied in scheduling the training data, which have an important role in improving the performance.

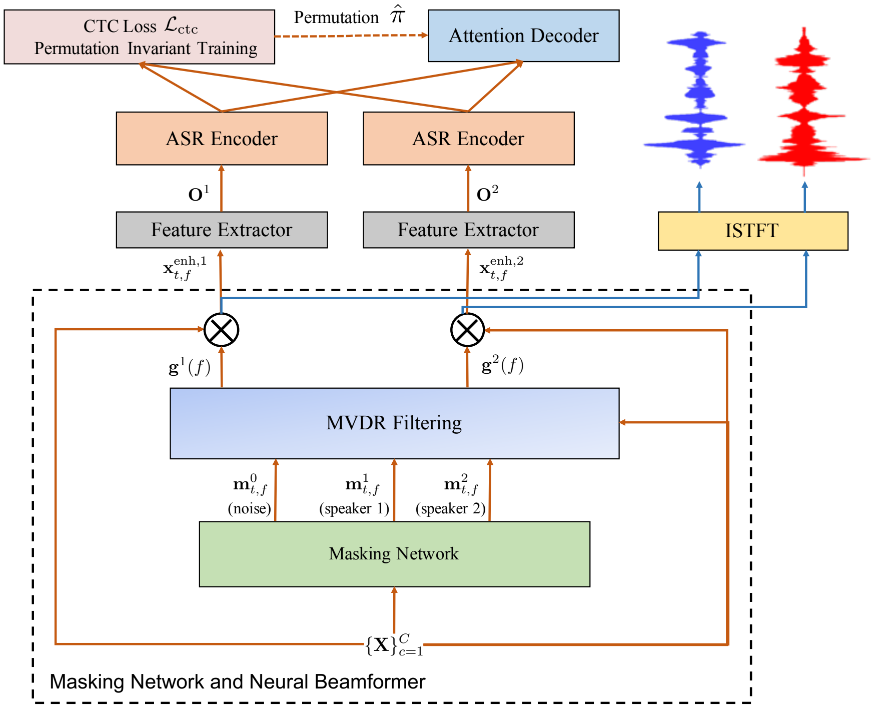

CTC Loss $\mathcal{L}_{\text{ctc}}$
Permutation Invariant Training
Permutation $\hat{\pi}$
Attention Decoder
ASR Encoder
ASR Encoder
$\mathbf{O}^1$
$\mathbf{O}^2$
Feature Extractor
Feature Extractor
ISTFT
$\mathbf{x}_{t,f}^{\text{enh},1}$
$\mathbf{x}_{t,f}^{\text{enh},2}$
$\mathbf{g}^1(f)$
$\mathbf{g}^2(f)$
MVDR Filtering
$\mathbf{m}_{t,f}^0$ (noise)
$\mathbf{m}_{t,f}^1$ (speaker 1)
$\mathbf{m}_{t,f}^2$ (speaker 2)
Masking Network
$\{\mathbf{X}\}_{c=1}^C$
Masking Network and Neural Beamformer

Figure 8.2: End-to-End Multi-channel Multi-speaker Model

Model Architecture

By using the differences in the signals recorded at each sensor, distributed sensors can exploit spatial information. They are thus particularly useful for separating sources that are spatially partitioned. In this work, we present a sequence-to-sequence architecture with multi-channel input and multi-channel output to model the multi-channel multi-speaker speech recognition, shown in Fig. 8.2 for the case of two speakers. The proposed end-to-end multi-channel multi-speaker ASR model can be divided into three stages. The first stage is a single-channel masking network to perform pre-separation by predicting multiple speaker and noise masks for each channel. Then a multi-source neural beamformer is used to spatially separate multiple speaker sources. In the last stage, an end-to-end ASR module with permutation-free training is used to perform the multi-output speech recognition.

We used a similar architecture as in (Ochiai et al., 2017a), where the masking network and the neural beamformer are integrated into an attention-based encoder-decoder neural network, and the whole model is jointly optimized solely via a speech recognition objective. The input of the model can consist of an arbitrary number of channels C, and its output is the text sequence for each speaker directly. We denote by K the number of speakers in the mixed utterances, and for simplicity of notation, we shall consider the noise component as the 0-th source.

Monaural Masking Network

The monaural masking network, shown at the bottom of Fig. 8.2, estimates the masks of each channel for every speaker and an extra noise component. Let us denote by $\mathbf{X}_c = (x_{t,f,c})_{t,f} \in \mathbb{C}^{T\times F}$ the complex STFT of the c-th channel of the observed multi-channel multi-speaker speech, where $1 \le t \le T$, $1 \le f \le F$, $1 \le c \le C$ denote time, frequency, and channel indices, respectively. The mask estimation module produces time-frequency masks $\mathbf{M}_c^i = (m_{t,f,c}^i)_{t,f} \in [0,1]^{T\times F}$, with $i \in \{1,\ldots,K\}$ for each of the K speakers, and $i=0$ for the noise, using the complex STFT of the c-th channel of the observed multi-channel multi-speaker speech as input. The computation is performed independently on each of the input channels:

$$\mathbf{M}_c = \mathrm{MaskNet}(\mathbf{X}_c), \tag{8.1}$$

where $\mathbf{M}_c = (\mathbf{M}_c^i)_i \in [0,1]^{T\times F\times K}$ is the set of estimated masks for the c-th channel.

Multi-source Neural Beamformer

The multi-source neural beamformer is a key component in the proposed model, which produces the separated speech of each speaker. The masks obtained on each channel for each speaker and the noise are used in the computation of the power spectral density (PSD) matrices of each source as follows (Yoshioka et al., 2015b; Heymann et al., 2016):

$$\mathbf{\Phi}^i(f) = \frac{1}{\sum_{t=1}^{T} \mathbf{m}_{t,f}^i} \sum_{t=1}^{T} \mathbf{m}_{t,f}^i \mathbf{x}_{t,f} \mathbf{x}_{t,f}^H \in \mathbb{C}^{C\times C}, \tag{8.2}$$

where $i \in \{0,\ldots,K\}$, $\mathbf{x}_{t,f} = \{x_{t,f,c}\}_{c=1}^{C}$, $\mathbf{m}_{t,f}^i = \{m_{t,f,c}^i\}_{c=1}^{C}$, and ^{H} represents the conjugate transpose.

After getting the PSD matrices of every speaker and the noise, we estimate the beamformer's time-invariant filter coefficients $\mathrm{g}^i(f)$ at frequency f for each speaker $i \in \{1,\cdots,K\}$ via the

MVDR formalization (Souden et al., 2009) as follows:

$$\mathbf{g}^i(f) = \frac{(\sum_{j\neq i} \mathbf{\Phi}^j(f))^{-1}\mathbf{\Phi}^i(f)}{\text{Tr}((\sum_{j\neq i} \mathbf{\Phi}^j(f))^{-1}\mathbf{\Phi}^i(f))}\mathbf{u} \in \mathbb{C}^C, \tag{8.3}$$

where $\mathbf{u} \in \mathbb{R}^C$ is a vector representing the reference microphone that is derived from an attention mechanism (Ochiai et al., 2017a), and Tr$(\cdot)$ denotes the trace operation. Notice that in Eq. 8.3, the formula to derive the filter coefficient is different from that in (Ochiai et al., 2017a) in the way that the noise PSD is replaced by $\sum_{j\neq i} \mathbf{\Phi}^j(f)$. This is because both noise and other speakers are considered as interference when focusing on a given speaker. This is akin to the speech-speech-noise (SSN) model in (Yoshioka et al., 2018c). Such a method is employed to make more accurate estimations of the PSD matrices, in which the traditional PSD matrix is expressed using the PSD matrix of interfering speaker and that of the background noise.

Finally, the beamforming filters $\mathbf{g}^i(f)$ obtained in Eq. 8.3 are used to separate and denoise the input overlapped multi-channel signals $\mathbf{x}_{t,f} \in \mathbb{C}^C$ to obtain a single-channel estimate of the enhanced STFT $\hat{s}^i_{t,f}$ for speaker i:

$$\hat{s}^i_{t,f} = (\mathbf{g}^i(f))^H \mathbf{x}_{t,f} \in \mathbb{C}. \tag{8.4}$$

Each separated speech signal waveform can be obtained by inverse STFT for listening, as iSTFT$(\hat{\mathbf{S}}^i)$ $, i = 1, \ldots, K$.

End-to-End Speech Recognition

The outputs of the neural beamformer are estimates of the separated speech signals for each speaker. Before feeding these streams to the end-to-end speech recognition submodule, we need to convert the STFT features to normalized log filterbank features. A log mel filterbank transformation is first applied on the magnitude of the beamformed STFT signal $\hat{\mathbf{S}}^i = (\hat{S}^i_{t,f})_{t,f}$ for each speaker i, and a global mean-variance normalization is then performed on the log-filterbank feature to produce a proper input $\mathbf{O}^i$ for the speech recognition submodule:

$$\mathbf{FBank}^i = \text{MelFilterBank}(|\hat{\mathbf{S}}^i|), \tag{8.5}$$

$$\mathbf{O}^i = \text{GlobalMVN}(\log(\mathbf{FBank}^i)). \tag{8.6}$$

We briefly introduce the end-to-end speech recognition submodule used here, which is similar to the joint CTC/attention-based encoder-decoder architecture (Kim et al., 2017b). The feature vectors $\mathbf{O}^i$ are first transformed to a hidden representation $\mathbf{H}^i$ by an encoder network. A decoder then generates the output token sequences based on the history information $\mathbf{y}$ and a weighted sum vector $\mathbf{c}$ obtained with an attention mechanism. The end-to-end speech recognition is computed as follows:

$$\mathbf{H}^i = \text{Encoder}(\mathbf{O}^i) \tag{8.7}$$

$$\mathbf{c}_n^i, \alpha_n^i = \text{Attention}(\alpha_{n-1}^i, \mathbf{e}_{n-1}^i, \mathbf{H}^i) \tag{8.8}$$

$$\mathbf{e}_n^i = \text{Update}(\mathbf{e}_{n-1}^i, \mathbf{c}_{n-1}^i, \mathbf{y}_{n-1}^i) \tag{8.9}$$

$$\mathbf{y}_n^i \sim \text{Decoder}(\mathbf{c}_n^i, \mathbf{y}_{n-1}^i), \tag{8.10}$$

where i denotes the index of the source stream and n an output label sequence index.

Typically, the history information $\mathbf{y}$ is replaced by the reference labels $\mathbf{R} = (r_1, \cdots, r_N)$ in a teacher-forcing fashion at training time. However, since there are multiple possible assignments between the inputs and the references, it is necessary to used permutation invariant training (PIT) in the end-to-end speech recognition (Seki et al., 2018; Chang et al., 2019b). The best permutation of the input sequences and the references is determined by the connectionist temporal classification (CTC) loss Loss$_{\text{ctc}}$:

$$\hat{\pi} = \underset{\pi \in \mathcal{P}}{\text{argmin}} \sum_i \text{Loss}_{\text{ctc}}(\mathbf{Z}^i, \mathbf{R}^{\pi(i)}), i = 1, \ldots, K, \tag{8.11}$$

where $\mathbf{Z}^i$ denotes the output sequence computed from the encoder output $\mathbf{H}^i$ for the CTC loss, $\mathcal{P}$ is the set of all permutations on $\{1, \ldots, K\}$, and $\pi(i)$ is the i-th element for permutation π.

The final ASR loss of the model is obtained as:

$$\mathcal{L} = \lambda \mathcal{L}_{\text{ctc}} + (1 - \lambda)\mathcal{L}_{\text{att}}, \tag{8.12}$$

$$\mathcal{L}_{\text{ctc}} = \sum_i \text{Loss}_{\text{ctc}}(\mathbf{Z}^i, \mathbf{R}^{\hat{\pi}(i)}), \tag{8.13}$$

$$\mathcal{L}_{\text{att}} = \sum_i \text{Loss}_{\text{att}}(\mathbf{Y}^i, \mathbf{R}^{\hat{\pi}(i)}), \tag{8.14}$$

where $0 \leq \lambda \leq 1$ is an interpolation factor, and Loss$_{\text{att}}$ is the cross-entropy loss to train the attention-based encoder-decoder networks.

Data Scheduling and Curriculum Learning

From preliminary empirical results, we find that it is relatively difficult to perform straightforward end-to-end training of such a multi-stage model, especially without an intermediate criterion to guide the training. In our model, the speech recognition submodule has the same architecture as the typical end-to-end speech recognition model, and the input is expected to be similar to the log filterbank of single-speaker speech. Thus, in order to train the model properly, we did not only use the spatialized utterances of the multi-speaker corpus but also the single-speaker utterances from the original WSJ training set. During training, every batch is randomly chosen either from the multi-channel multi-speaker set or from the single-channel single-speaker set. For single-speaker batches, the masking network and neural beamformer stages are bypassed, and the input is directly fed to the recognition submodule. Furthermore, the loss is calculated without considering permutations, as there is only a single speaker per input.

With this data scheduling scheme, the model can achieve a decent performance from random initialization. For multi-channel multi-speaker data batches, the loss of the ASR objective function is back-propagated down through the model to the masking network. For data batches consisting of single-speaker utterances, only the speech recognition part is optimized, which leads to more accurate loss computation in the future. The single-speaker data batches rectify the behavior of the ASR model as it performs regularization during the training.

According to previous researches, starting from easier subtasks can lead the model to learn better, an approach called curriculum learning (Bengio et al., 2009; Amodei et al., 2016). To further exploit the data scheduling scheme, we introduce more constraints on the order of the data batches of the training set. As was observed in prior research by (Qian et al., 2018a), the signal-to-noise ratio (SNR, the energy ratio between the target speech and the interfering sources) has a great influence on the final recognition performance. When the speech energy levels of the target speaker and the interfering sources are obviously different, the recognition accuracy of the interfering source speech is very poor. Thus, we sort the multi-speaker data in ascending order of SNR between the loudest and quietest speaker, thus starting with mixtures where both speakers are at similar levels. Furthermore, we sort the single-speaker data from short to long, as short sequences tend to be easier to learn in seq2seq learning. The strategy is formally depicted in Algorithm 3. We applied such a curriculum learning strategy in order to make the model learn step by step and expect it to improve the training.

8.3 Experiment

To check the effectiveness of our proposed end-to-end model, we use the spatialized WSJ-2Mix corpus introduced in the previous section Sec. 1.4.4. More specifically, we evaluated it on the remixed WSJ data used in (Seki et al., 2018), which we here refer to as the wsj-2mix dataset. The multi-speaker speech training set was generated by randomly selecting two utterances from the WSJ SI284 corpus, resulting in a 98.5 h dataset. The signal-to-noise ratio (SNR) of one source against the other was randomly chosen from a uniform distribution in the range of $[-5, 5]$ dB. The validation and evaluation sets were generated in a similar way by selecting source utterances from

Algorithm 3: Curriculum learning strategy

```
Load the training dataset $\mathbf{X}$;
Categorize the training data $\mathbf{X}$ into single-channel single-speaker data $\mathbf{X}_{\text{clean}}$ and
  multi-channel multi-speaker data $\mathbf{X}_{\text{noisy}}$;
Sort the single-channel single-speaker training data in $\mathbf{X}_{\text{clean}}$ in ascending order of the
  utterance lengths, leading to $\mathbf{X}'_{\text{clean}}$;
Sort the multi-channel multi-speaker training data in $\mathbf{X}_{\text{noisy}}$ in ascending order of the SNR
  level, leading to $\mathbf{X}'_{\text{noisy}}$ ;
Divide $\mathbf{X}'_{\text{clean}}$ and $\mathbf{X}'_{\text{noisy}}$ into minibatch sets $\mathcal{B}_{\text{clean}}$ and $\mathcal{B}_{\text{noisy}}$;
Sort batches to alternate between batches from $\mathcal{B}_{\text{clean}}$ and $\mathcal{B}_{\text{noisy}}$;
while model is not converged do
    for each $b$ in all minibatches do
        Feed minibatch $b$ into the model, update the model;
    end
end
while model is not converged do
    Shuffle the training data in $\mathbf{X}_{\text{clean}}$ and $\mathbf{X}_{\text{noisy}}$ randomly and divide them into minibatch
      sets $\mathcal{B}'_{\text{clean}}$ and $\mathcal{B}'_{\text{noisy}}$;
    Select each minibatch randomly from $\mathcal{B}'_{\text{clean}}$ and $\mathcal{B}'_{\text{noisy}}$ and feed it into the model
      iteratively to update the model;
end
```

the WSJ Dev93 and Eval92 respectively, and the durations are 1.3 h and 0.8 h. We then create a new spatialized version of the WSJ-2Mix dataset following the process applied to the wsj0-2mix dataset in (Wang et al., 2018), using a room impulse response (RIR) generator[1], where the characteristics of each two-speaker mixture are randomly generated including room dimensions, speaker locations, and microphone geometry[2].

To train the model, we used the spatialized WSJ-2Mix data with $K = 2$ speakers as well as the train_si284 training set from the WSJ1 dataset to regularize the training procedure. All input data are raw waveform audio signals. The STFT was computed using 25 ms-width Hann window with 10 ms shift, with zero-padding resulting in a spectral dimension $F = 257$. In our experiments, we only report results in the case of $C = 2$ channels, but our model is flexible and can be used with an arbitrary number of channels. We first report recognition and separation results in the anechoic scenario in Sections 8.3 and 8.3. Then we show preliminary results in the reverberant scenario in Section 8.3.

Configurations

Our end-to-end multi-channel multi-speaker model is completely built based on the ESPnet framework (Watanabe et al., 2018) with Pytorch backend. All the network parameters were initialized randomly from uniform distribution in the range $[-0.1, 0.1]$. We used AdaDelta with $\rho = 0.95$ and $\epsilon = 1e^{-8}$ as optimization method. The maximum number of epochs for training is set to 15 but the training process is stopped early if performance does not increase for 3 consecutive epochs. For decoding, a word-based language model (Hori et al., 2018) was trained on the transcripts of the WSJ corpus.

Neural Beamformer

The mask estimation network is a 3-layer bidirectional long-short term memory with projection (BLSTMP) network with 512 cells in each direction. The computation of the reference microphone vector has the same parameters as in (Ochiai et al., 2017a) except the vector dimension which is here set to 512. In the MVDR formula of Eq. 8.3, a small value ϵ is added to the PSD matrix to guarantee that an inverse exists.

[1] Available online at `https://github.com/ehabets/RIR-Generator`

[2] The spatialization toolkit is available at `http://www.merl.com/demos/deep-clustering/spatialize_wsj0-mix.zip`

Encoder-Decoder Network

The encoder network consists of two VGG-motivated CNN blocks and three BLSTMP layers. The CNN layers have a kernel size of 3×3 and the number of feature maps is 64 and 128 in the first and second block, respectively. Every BLSTMP layer has 1024 memory cells in each direction with projection size 1024. 80 dimensional log filterbank features are extracted for each separated speech signals and global mean-variance normalization is applied, using the statistics of the single-speaker WSJ1 training set. In the decoder network, there is only a single layer of unidirectional long-short term memory network (LSTM) and the number of cells is 300. The interpolation factor λ of the loss function in Eq. 8.12 is set to 0.2.

Performance of Multi-Speaker Speech Recognition

In this subsection, we describe the speech recognition performance on the spatialized anechoic WSJ-2Mix data, which only modifies the signals via delays and decays due to the propagation. Note that although beamforming algorithms can address the anechoic case without too much effort, it is still necessary to show that our proposed end-to-end method can address the multi-channel multi-speaker speech recognition problem and both the speech recognition submodule and the neural beamforming separation submodule perform well as they are designed. We shall also note that the whole system is trained solely through an ASR objective, and it is thus not trivial for the system to learn how to properly separate the signals even in the anechoic case.

The multi-speaker speech recognition performance is shown in Table 8.1. There are three single-channel end-to-end speech recognition baseline systems. The first one is a single-channel multi-speaker ASR model trained on the first channel of the spatialized corpus, where the model is the same as the one proposed in (Chang et al., 2019b). The second is a single-channel multi-speaker ASR model trained with speech that is enhanced by BeamformIt (Anguera et al., 2007), which is a well-known delay-and-sum beamformer. And the third one is to use BeamformIt to first separate the speech by choosing its best and second-best output streams, and then to recognize them with a normal single-speaker end-to-end ASR model. The spatialization of the corpus results in a degradation of the performance: the multi-speaker ASR model trained with the 1st channel has a word error rate (WER) of 29.43% on the evaluation set, compared to only 20.43% obtained on the original unspatialized WSJ-2Mix data in (Chang et al., 2019b). Using the BeamformIt tool to enhance the spatialized signal can improve the recognition accuracy of a multi-speaker model, leading to a WER of 21.75% on the evaluation set. However, traditional beamforming algorithms such as BeamformIt can not perfectly separate the overlapped speech signals, and the performance of the single-speaker model in terms of WER is very poor, 98.00%.

The performance of our proposed end-to-end multi-channel multi-speaker model (MIMO-Speech) is shown at the bottom of the table. The curriculum learning strategy described in Section 8.2 is used to further improve performance. From the table, it can be observed that MIMO-Speech is significantly better than traditional methods, achieving 4.51% character error rate (CER) and 8.62% word error rate (WER). Compared against the best baseline model, the relative improvement is over 60% in terms of both CER and WER. When applying our data scheduling scheme by sorting the multi-speaker speech in ascending order of SNRs, an additional performance boost can be realized. The final CER and WER on the evaluation set are 3.75% and 7.55% respectively, with over 12% relative improvement against MIMO-Speech without curriculum learning. Overall, our proposed MIMO-Speech network can achieve good recognition performance on the spatialized anechoic WSJ-2Mix corpus.

Table 8.1: Performance in terms of average CER and WER [%] on the spatialized anechoic WSJ-2Mix corpus.

Model	dev CER	eval CER
2-spkr ASR (1st channel)	22.65	19.07
BeamformIt Enhancement (2-spkr ASR)	15.23	12.45
BeamformIt Separation (1-spkr ASR)	77.30	77.10
MIMO-Speech	7.29	4.51
+ Curriculum Learning (SNRs)	**6.34**	**3.75**
Model	dev WER	eval WER
2-spkr ASR (1st channel)	34.98	29.43
BeamformIt Enhancement (2-spkr ASR)	26.61	21.75
BeamformIt Separation (1-spkr ASR)	98.60	98.00
MIMO-Speech	13.54	8.62
+ Curriculum Learning (SNRs)	**12.59**	**7.55**

Performance of Multi-Speaker Speech Separation

One question regarding our model is whether the front-end of MIMO-Speech, the neural beamformer, learns a proper beamforming behavior as other algorithms do since there is no explicit speech separation criterion to optimize the network. To investigate the role of the neural beamformer, we consider the masks $\mathbf{m}^i$ that are used to compute the PSD matrices and the enhanced separated STFT signals $\hat{\mathrm{s}}^i, i = 1, \ldots, J$ obtained at the output of the beamformer. Example results are shown in Fig. 8.3. Note that in our model, the masks are not constrained to sum to 1 at each time-frequency unit, resulting in a scaling indeterminacy within each frequency. For better readability in the figures, we here renormalize each mask using its median within each frequency. In the figure, the difference between the masks from each speaker is clear. And from the spectrogram, it is also observed that for each separated stream, the signals are less overlapped compared with the input multi-speaker speech signal. The mask and spectrogram examples suggest that MIMO-Speech can separate the speech to some level.

To evaluate the separation quality, we reconstruct the separated waveforms for each speaker from the outputs of the beamformer via inverse STFT, and compare them with the reference signals in terms of PESQ and scale-invariant signal-to-distortion ratio (SI-SDR) (Le Roux et al., 2019b). The results are shown in Table 8.2. As we can see, the separated audios have very good quality. The separated signals from the MIMO-Speech model [3] have an average PESQ value of 3.6 and an average SI-SDR of 23.1 dB. When using curriculum learning, PESQ and SI-SDR degrade slightly, but the quality is still very high. This result suggests that our proposed MIMO-Speech model is capable of learning to separate overlapped speech via beamforming, based solely on an ASR objective.

Table 8.2: Performance in terms of average PESQ and SI-SDR [dB] on the spatialized anechoic WSJ-2Mix corpus.

Model	dev PESQ	eval PESQ
MIMO-Speech	3.6	3.6
+ Curriculum Learning (SNRs)	**3.7**	**3.6**

Model	dev SI-SDR	eval SI-SDR
MIMO-Speech	**22.1**	**23.1**
+ Curriculum Learning (SNRs)	21.1	21.8

In order to further explore the neural beamformer's effect, we show an example of estimated beam pattern (Gannot et al., 2017) for the separated sources. Figure 8.4 shows the beam pattern of two separated signals at frequencies $\{500\text{ Hz}, 1000\text{ Hz}, 2000\text{ Hz}, 4000\text{ Hz}\}$. The value of the beam at different degrees quantifies the reduction of the speech signals received. As we can see from the figures, the crests and troughs of the beams are different for the two speakers, which shows the neural beamformer is trained properly and can tell the difference between the sources.

Table 8.3: Performance in terms of average CER and WER [%] of the baseline single-speaker end-to-end speech recognition model trained on reverberant (R) single-speaker speech and evaluated on reverberant (R) multi-speaker speech.

Model	dev CER (R)	eval CER (R)
End-to-End Model (R)	81.6	82.7

Model	dev WER (R)	eval WER (R)
End-to-End Model (R)	103.9	104.2

[3] Audio samples are available online at `https://simpleoier.github.io/MIMO-Speech/index.html`

Evaluation on the spatialized reverberant data

To give a comprehensive analysis of the MIMO-Speech model, we investigated how the model performs in a more realistic case, using the spatialized reverberant WSJ-2Mix data. As a comparison, we first trained a normal single-speaker end-to-end speech recognition system. The model is trained with the spatialized reverberant speech from each single speaker. The performance is shown in Table 8.3. For the MIMO-Speech model, the spatialized reverberant WSJ-2Mix training dataset was added to the training set for the multi-conditioned training. The results on the speech recognition task are shown in Table 8.4. The reverberant speech is difficult to recognize as the performance shows severe degradation when we tried to infer the reverberant speech using the anechoic multi-speaker model. The multi-conditioned training can release such degradation, improving the WER by over 60%. The results suggest that the proposed MIMO-Speech also has potential for application in complex scenarios. As a complementary experiment, we used Nara-WPE (Drude et al., 2018) to perform speech dereverberation only for the development and evaluation data. The speech recognition results are shown in Table.8.5 which suggests that the speech dereverberation techniques only in the inference stage can lead to further improvement. Note that the results here are just a preliminary study. The main drawback here is that we did not consider any dereverberation techniques in designing our model.

Table 8.4: Performance in terms of average CER and WER [%] on the spatialized WSJ-2Mix corpus of MIMO-Speech trained on either anechoic (A) or reverberant (R) and evaluated on either the anechoic (A) or reverberant (R) evaluation set.

Model	eval CER (A)	eval CER (R)
MIMO-Speech (A)	4.51	62.32
MIMO-Speech (R)	4.08	18.15
Model	eval WER (A)	eval WER (R)
MIMO-Speech (A)	8.62	81.30
MIMO-Speech (R)	8.72	29.99

8.4 Conclusion

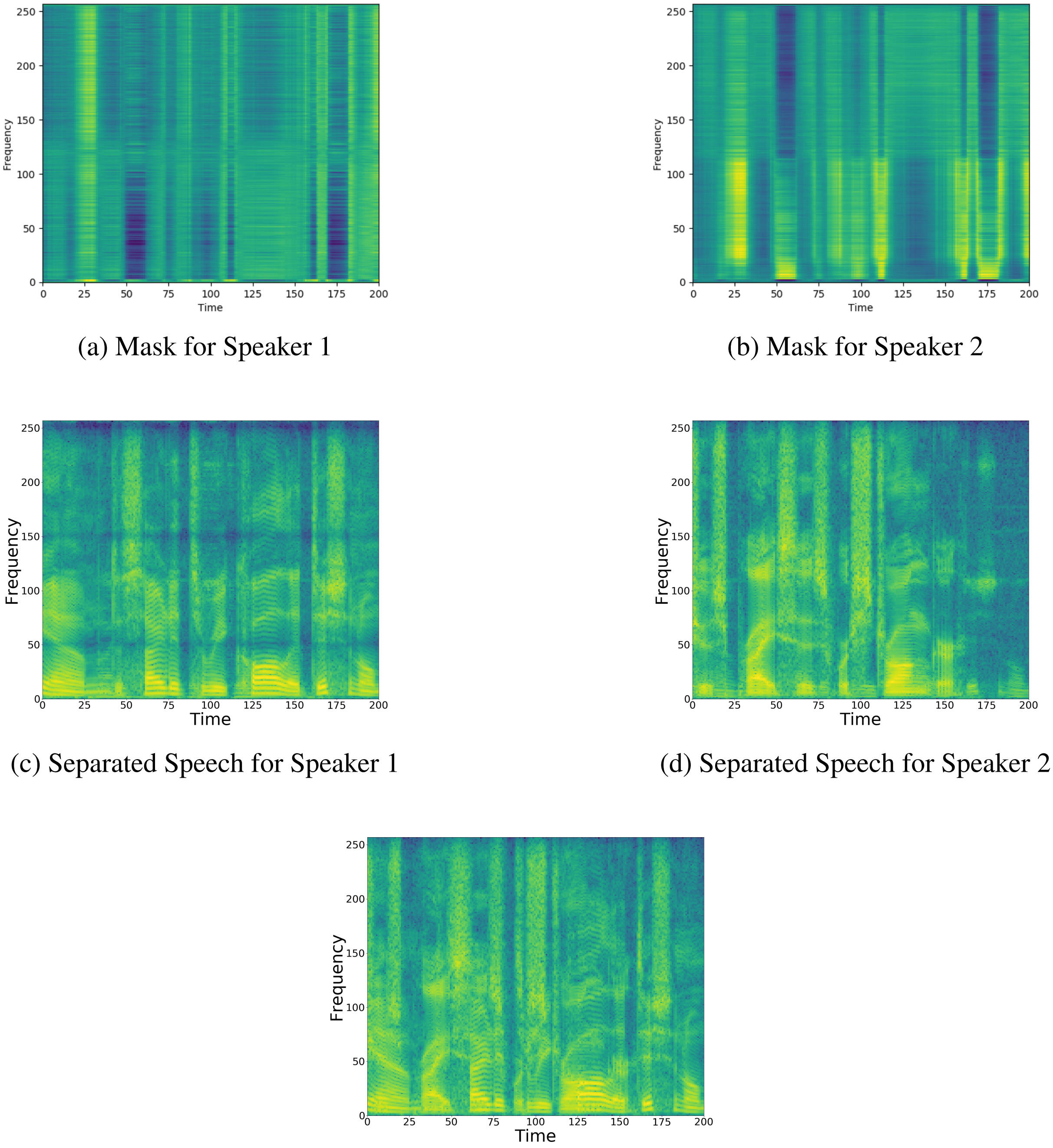
In this chapter, we present an end-to-end multi-channel multi-speaker speech recognition model called MIMO-Speech. More specifically, the model takes multi-speaker speech recorded by a microphone array as input and outputs text sequences for each speaker. Furthermore, the front-end of the model, involving a neural beamformer, learns to perform speech separation even though no explicit signal reconstruction criterion is used. The main advantage of the proposed approach is

Table 8.5: Performance in terms of average CER and WER [%] on the spatialized WSJ-2Mix corpus of MIMO-Speech trained on either anechoic (A) or reverberant (R) and evaluated on the reverberant data after Nara-WPE dereverberation (D).

Model	dev CER (D)	eval CER (D)
MIMO-Speech (A)	51.00	52.02
MIMO-Speech (R)	20.09	15.04
Model	dev WER (D)	dev WER (D)
MIMO-Speech (A)	69.08	69.42
MIMO-Speech (R)	33.09	25.28

that the whole model is differentiable and can be optimized with an ASR loss as target. In order to make the training easier, we utilized single-channel single-speaker speech as well. We also designed an effective curriculum learning strategy to improve the performance. Experiments on a spatialized version of the WSJ-2Mix corpus show that the proposed framework has fairly good performance. However, performance on reverberant data still suffers from a large gap against the anechoic case.

Building on these findings, the next chapter will explore further improvements to the MIMO-Speech model by leveraging the Transformer architecture. The Transformer-based approach is expected to address some of the limitations observed in the RNN-based model, particularly in handling reverberant environments, and will integrate additional techniques to enhance robustness and performance.



(a) Mask for Speaker 1

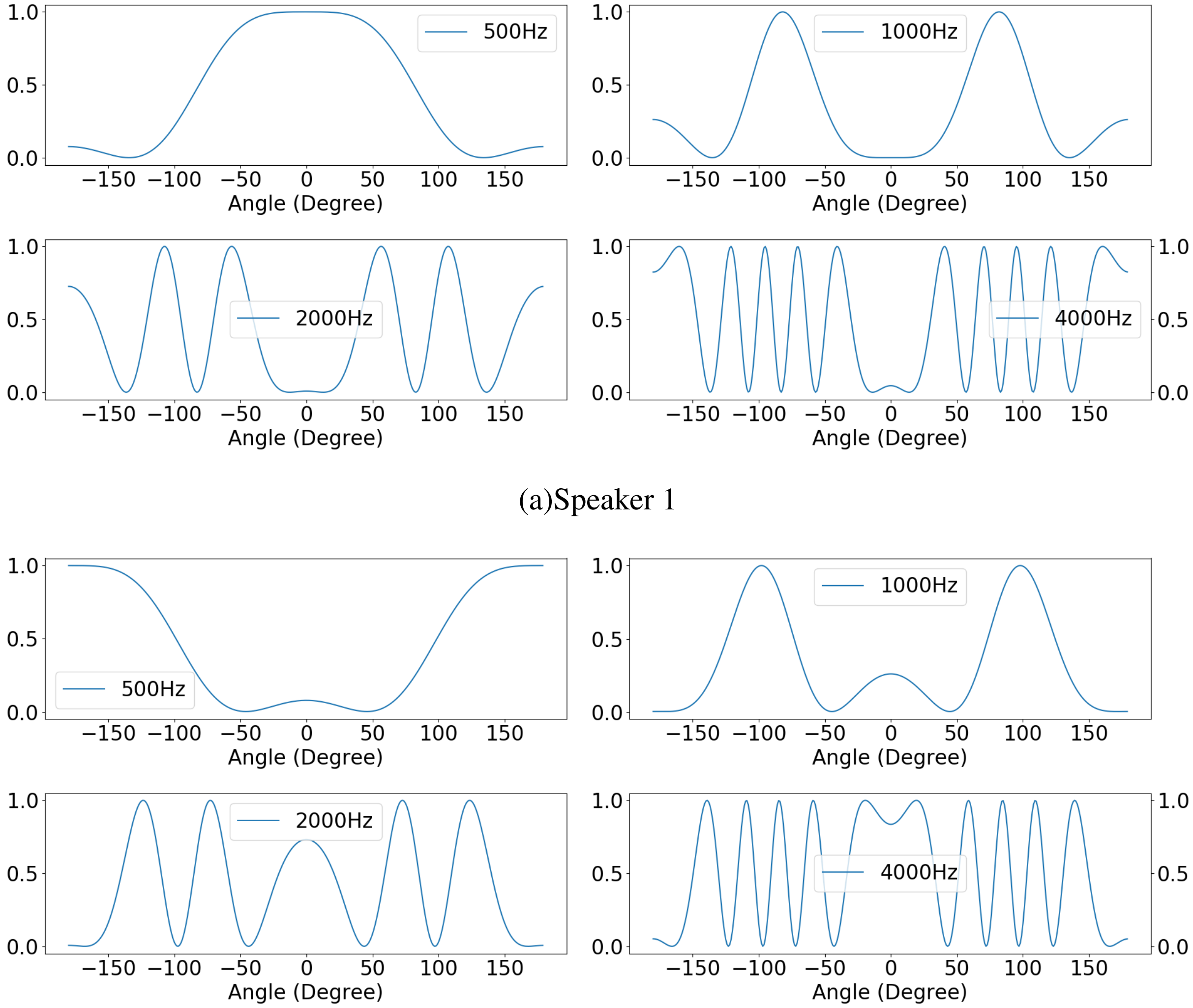
(b) Mask for Speaker 2

(c) Separated Speech for Speaker 1

(d) Separated Speech for Speaker 2

(e) Overlapped Speech

Figure 8.3: Example of masks output by the masking network and separated speech log spectrograms output by the MVDR beamformer.



(a)Speaker 1

(b) Speaker 2

Figure 8.4: Example of beam patterns of the separated speech.

Chapter 9

MIMO-Speech-Transformer: improving multi-channel multi-speaker E2E ASR with Transformer

Summary

In the previous chapter (Ch. 8), we introduced the MIMO-Speech model tailored for multi-channel multi-speaker speech recognition. This model adopted the E2E modular-based design and demonstrated promising results in transcribing multi-speaker overlapping speech by leveraging spatial information from multi-channel input data. The model's neural beamforming-based speech separation successfully handled overlapping speech scenarios. However, the model was based on recurrent neural networks (RNNs). Recognizing the significant advancements and benefits offered by Transformer architectures in sequence-to-sequence tasks since their introduction by Vaswani et al. (Vaswani et al., 2017), in this chapter, we enhance our MIMO-Speech model by adopting the Transformer architecture. This transition to Transformer-based modeling aims to leverage the Transformer's capabilities in capturing long-range dependencies and improving performance in complex speech recognition tasks. Similar to the previous chapter, the proposed model follows the E2E modular-based design, as mentioned in Sec. 2.

Furthermore, to enhance the robustness of our model in reverberant environments, we incorporate an external dereverberation method known as Weighted Prediction Error (WPE) to preprocess reverberated speech signals. This preprocessing step contributes to mitigating the detrimental effects of reverberation, thereby improving the overall performance and accuracy of our MIMO-Speech model in real-world settings.

Chang, Xuankai, Wangyou Zhang, Yanmin Qian, Jonathan Le Roux, and Shinji Watanabe. ICASSP 2020. End-to-end multi-speaker speech recognition with transformer.

9.1 Introduction

Deep learning techniques have dramatically improved the performance of separation and automatic speech recognition (ASR) tasks related to the cocktail party problem (Cherry, 1953), where the speech from multiple speakers overlaps. Two main scenarios are typically considered, single-channel and multi-channel. In single-channel speech separation, various methods have been proposed, among which deep clustering (DPCL) based methods (Hershey et al., 2016a) and permutation invariant training (PIT) based methods (Yu et al., 2017c) are the dominant ones. For ASR, methods combining separation with single-speaker ASR as well as methods skipping the explicit separation step and building directly a multi-speaker speech recognition system have been proposed, using either the hybrid ASR framework (Yu et al., 2017a; Chang et al., 2018b; Menne et al., 2019) or the end-to-end ASR framework (Settle et al., 2018; Seki et al., 2018; Chang et al., 2019a). In the multi-channel condition, the spatial information derived from the inter-channel differences can help distinguish between speech sources from different directions, which makes the problem easier to solve. Several methods have been proposed for multi-channel speech separation, including DPCL-based methods using integrated beamforming (Drude and Haeb-Umbach, 2017) or inter-channel spatial features (Wang et al., 2018), and a PIT-based method using a multi-speaker mask-based beamformer (Yoshioka et al., 2018c). For multi-channel multi-speaker speech recognition, an end-to-end system was proposed in (Chang et al., 2019c), called MIMO-Speech because of the multi-channel input (MI) and multi-speaker output (MO). This system consists of a mask-based neural beamformer frontend, which explicitly separates the multi-speaker speech via beamforming, and an end-to-end speech recognition model backend based on the joint CTC/attention-based encoder-decoder (Kim et al., 2017b) to recognize the separated speech streams. This end-to-end architecture is optimized via only the connectionist temporal classification (CTC) and cross-entropy (CE) losses in the backend ASR, but is nonetheless able to learn to develop relatively good separation abilities.

Recently, Transformer models (Vaswani et al., 2017) have shown impressive performance in many tasks, such as pretrained language models (Radford et al., 2018; Devlin et al., 2018), end-to-end speech recognition (Karita et al., 2019b,a), and speaker diarization (Fujita et al., 2019), surpassing the long short-term memory recurrent neural networks (LSTM-RNNs) based models. One of the key components in the Transformer model is self-attention, which computes the contribution information of the whole input sequence and maps the sequence into a vector at every time step. Even though the Transformer model is powerful, it is usually not computationally practical when the sequence length is very long. It also needs adaptation for specific tasks, such as the subsampling operation in encoder-decoder based end-to-end speech recognition. However, for signal-level processing tasks such as speech separation and enhancement, subsampling is usually not a good option, because these tasks need to maintain the original time resolution.

In this work, we explore the use of Transformer models for end-to-end multi-speaker speech recognition in multi-channel scenarios. First, we replace the LSTMs in the encoder-decoder network of the speech recognition module with Transformers for both scenarios. Second, in order to also apply Transformers in the masking network of the neural beamforming module in the multi-channel case, we modify the self-attention layers to reduce their memory consumption in a time-restricted (or local) manner, as used in (Luong et al., 2015; Povey et al., 2018; Chang et al., 2018b). To the best of our knowledge, this work is the first attempt to use the Transformer model for tasks such as speech enhancement/separation with such very long sequences. Another contribution of this work is to improve the robustness of our model in reverberant environments. To do so, we incorporate an external dereverberation method, the weighed prediction error (WPE) (Yoshioka and Nakatani, 2012), to preprocess the reverberated speech. The experiments show that this straightforward method can lead to a performance boost for reverberant speech.

9.2 Method

We follow the similar model architecture as MIMO-Speech in Chapter 8. In the previous model, the masking network in the neural beamformer and the E2E-ASR are based on long-short-term memory (LSTM) network. Motivated by the recent success in Transformer (Vaswani et al., 2017), we replace the LSTM in the original MIMO-Speech model by Transformers. We will skip most of the details about the Transformers. Please refer to the original paper (Vaswani et al., 2017) for more details. However, we will talk about the self-attention part in this section.

Transformer with Time-restricted Self-Attention

In this part, we describe one of the key components in the Transformer architecture, the multi-head self-attention (Vaswani et al., 2017), and the time-restricted modification (Povey et al., 2018) for its application in the masking network of the frontend.

Transformers employ the dot-product self-attention for mapping a variable-length input sequence to another sequence of the same length, making them different from RNNs. The input consists of queries $\mathbf{Q}$, keys $\mathbf{\Omega}$, and values $\mathbf{V}$ of dimension d^{att}. The weights of the self-attention are obtained by computing the dot-product between the query and all keys and normalizing with softmax. A scaling factor $\sqrt{d^{\text{att}}}$ is used to smooth the distribution:

$$\text{Attention}(\mathbf{Q}, \mathbf{\Omega}, \mathbf{V}) = \text{softmax}\Big(\frac{\mathbf{Q}\mathbf{\Omega}^T}{\sqrt{d^{\text{att}}}}\Big)V. \tag{9.1}$$

To capture information from different representation subspaces, multi-head attention (MHA) is used by multiplying the original queries, keys, and values by different weight matrices:

$$\text{MHA}(\mathbf{Q}, \mathbf{\Omega}, \mathbf{V}) = \text{Concat}([H_h]_{h=1}^{d^{\text{head}}})W^{\text{head}}, \tag{9.2}$$

$$\text{where } H_h = \text{Attention}(\mathbf{Q}W_h^q, \mathbf{\Omega}W_h^k, \mathbf{V}_h^v W_h^v), \tag{9.3}$$

where d^{head} is the number of heads, and $W^{\text{head}} \in \mathbb{R}^{(d^{\text{head}}d^{\text{att}}) \times d^{\text{att}}}$ and $W_h^q, W_h^k, W_h^v \in \mathbb{R}^{d^{\text{att}} \times d^{\text{att}}}$ are learnable parameters.

In general, the speech sequence length can be considerably long, making self-attention computationally difficult. For tasks like speech separation and enhancement, the technique of subsampling is not practical as in speech recognition. Inspired by (Luong et al., 2015; Povey et al., 2018), we adjust the self-attention of the Transformers in the masking network to be performed on a local segment of the speech, because those frames have higher correlation. This time-restricted self-attention for the query at time step t is formalized as:

$$\text{Attention}(\mathbf{Q}, \mathbf{\Omega}', \mathbf{V}') = \text{softmax}\big(\frac{\mathbf{Q}\mathbf{\Omega}'^T}{\sqrt{d^{\text{att}}}}\big)\mathbf{V}', \tag{9.4}$$

where the corresponding keys and values are $\mathbf{\Omega}' = \mathbf{\Omega}_{t-l:t+r}$ and $\mathbf{V}' = \mathbf{V}_{t-l:t+r}$, respectively, with l and r here denoting the left and right context window sizes.

9.3 Experiment

The proposed methods were evaluated on the dataset, spatialized WSJ-2Mix dataset, introduced in Sec. 1.4.4. The number of speakers in mixture utterances is $K = 2$. The multi-channel speech signals were generated[1] from the monaural WSJ-2Mix speech used in (Seki et al., 2018; Chang et al., 2019a). The room impulse responses (RIR) for the spatialization were randomly generated[2], characterizing the room dimensions, speaker locations, and microphone geometry. The final spatialized dataset contains two different environment conditions, anechoic and reverberant. In the anechoic condition, the room is assumed to be anechoic and only the delays and decays due to the

[1] The spatialization toolkit is available at `http://www.merl.com/demos/deep-clustering/spatialize_wsj0-mix.zip`

[2] The RIR generator script is available online at `https://github.com/ehabets/RIR-Generator`

propagation are considered when generating the signals. In the reverberant condition, reverberation is also considered, with randomly drawn T60s from $[0.2, 0.6]$ s. In total, the spatialized corpus under each condition contains 98.5 hr, 1.3 hr, and 0.8 hr in training, development, and evaluation sets respectively.

To better illustrate the performance gain of using Transformer networks, we also conduct experiments with monaural multi-speaker E2E ASR model described in Chapter 5. In the single-channel multi-speaker speech recognition task, we used the 1st channel of the training, development, and evaluation set to train, validate, and evaluate our model respectively. The input features are 80-dimensional log mel-filterbank coefficients with pitch features and their delta and delta delta features.

In the multi-channel multi-speaker speech recognition task, we also followed (Chang et al., 2019c) in including the WSJ train_si284 in the training set to improve the performance. The model takes the raw waveform audio signal as input and converts it to its STFT using a 25 ms-long Hann window with stride 10 ms. The spectral feature dimension is $F = 257$ due to zero-padding. After the frontend computation, 80-dimensional log filterbank features are extracted for each separated speech signal and global mean-variance normalization is applied, using the statistics of the single-speaker WSJ1 training set. All the multi-channel experiments were performed with $C = 2$ channels. However, the model can be extended to an arbitrary number of input channels as described in (Ochiai et al., 2017a).

Experimental Setup

All the proposed end-to-end multi-speaker speech recognition models are implemented with the ESPnet framework (Watanabe et al., 2018) using the Pytorch backend. Some basic parts are the same for all the models. The interpolation factor λ of the loss function in (8.12) is set to 0.2. The word-level language model (Hori et al., 2018) used during decoding was trained with the official text data included in the WSJ corpus. The configurations of the RNN-based models are the same as in (Chang et al., 2019a) and (Chang et al., 2019c) for single-channel and multi-channel experiments, respectively.

In the Transformer-based multi-speaker encoder-decoder ASR model, there is a total of 12 layers in the encoder and 6 layers in the decoder as in (Karita et al., 2019b). Before the Transformer encoder, the log mel-filterbank features are encoded by two CNN blocks. The CNN layers have a kernel size of 3×3 and the number of feature maps is 64 in the first block and 128 in the second block. For the single-channel multi-speaker model inroduced in Chapter 5.2, Encoder$_{\text{Mix}}$ is the same as the CNN embedding layer, and Encoder$_{\text{SD}}$ and Encoder$_{\text{Rec}}$ contain 4 and 8 Transformer layers, respectively. For all the tasks, the configuration of each encoder-decoder layer is $d^{\text{att}} = 256$, $d^{\text{ff}} = 2048$, $d^{\text{head}} = 4$. The masking network in the frontend has 3 layers similar to the encoder-

Table 9.1: Performance in terms of average WER [%] on the **single-channel anechoic** WSJ-2Mix corpus.

Model	dev	eval
RNN-based 1-channel Model (Chang et al., 2019a)	24.90	20.43
Transformer-based 1-channel Model	**17.11**	**12.08**

decoder layer except $d^{\mathrm{ff}} = 768$. The training stage of Transformer runs with the Adam optimizer and Noam learning rate decay as in (Vaswani et al., 2017). Note that the backend ASR module is currently initialized with a pretrained model from the ESPnet recipe of WSJ corpus and kept frozen for the first 15 epochs, for training stability.

Performance in Anechoic Condition

We first provide in Table 9.1 the performance in anechoic condition of the single-channel multi-speaker end-to-end ASR models trained and evaluated on the original single-channel WSJ-2Mix corpus used in (Hori et al., 2018; Chang et al., 2019a). All the layers are randomly initialized. The result shows that using the Transformer model leads to a 40.9% relative word error rate (WER) improvement on the evaluation set, decreasing from 20.43% to 12.08% compared with the RNN-based model in (Chang et al., 2019a).

The multi-channel multi-speaker speech recognition performance is shown in Table 9.2 using the spatialized anechoic WSJ-2Mix dataset. The baseline multi-channel system is the RNN-based model from our previous study (Chang et al., 2019c). Before we move to the fully Transformer-based MIMO-Speech model, we first replace the RNNs with Transformers in the backend ASR only. We see that using Transformers for the ASR backend can achieve 20.5% relative improvement against the RNN-based model in anechoic conditions.

We then also apply Transformers in the masking network of the frontend. Considering the feasibility of computing, in this preliminary study, the left and right context window sizes of the self-attention are set to $l = 14$ and $r = 15$. The parameters of the frontend are randomly initialized. Compared with using a Transformer-based model only for the backend, the fully Transformer-based model leads to a further improvement, achieving a WER of 6.41%. Compared against the whole sequence information available in the RNN-based model, such a small context window greatly limits the power of our model but shows its potential. Overall, the proposed fully Transformer-based model achieves a 25.6% relative WER improvement against the RNN-based model in the multi-channel case. We also see that the multi-channel system is better than the single-channel system, thanks to the availability of spatial information.

Table 9.2: Performance in terms of average WER [%] on the spatialized **two-channel anechoic** WSJ-2Mix corpus.

Model	dev	eval
RNN-based MIMO-Speech (Chang et al., 2019c)	13.54	8.62
+ Transformer backend	**10.73**	6.85
++ Transformer frontend	11.75	**6.41**

Performance in Reverberant Condition

Even though our model can perform very well in anechoic condition, such ideal environments are rarely encountered in practice. It is thus crucial to investigate whether the model can be applied in more realistic environments. In this subsection, we describe preliminary efforts to process the reverberated signal.

We first used a straightforward multi-conditioned training by adding reverberated utterances into the training set. The results of multi-speaker speech recognition on the multi-channel reverberant datasets are shown in Table 9.3. It can be observed that only using the Transformers for the backend is 6.6% better than the RNN-based model. In addition, the fully Transformer-based model achieves 13.2% relative WER improvement on the evaluation set, which is consistent with the anechoic case. However, comparing with the numbers for the anechoic condition in Table 9.2, a large performance degradation can be observed.

To alleviate this, we turned to an existing external dereverberation method to preprocess the input signals as a simple yet effective solution. Nara-WPE (Drude et al., 2018) is a widely used open source software for blind dereverberation of acoustic signals. The dereverberation is performed on the reverberated speech before it is added to the training dataset with anechoic data. Similarly, the reverberant test set is also preprocessed. Speech recognition performance on the multi-channel reverberant speech after Nara-WPE is shown in Table 9.4. In general, the WERs are dramatically decreased with the dereverberation method. For the RNN-based model, the WER on the evaluation set decreased by 41.1% relative, from 29.99% to 17.67%. Similar to the experiments under other conditions, the model with backend Transformer only is better than the RNN-based baseline model on the reverberant evaluation set by 13.8% relative WER. However, the Transformer-based frontend slightly degraded the performance. This may be due the window size of the attention being too small, as it only covers about 0.3 s of speech. Note that our systems are not trained through Nara-WPE, which is left for future work.

At last, we show results in the single-channel task with the 1st channel of the reverberated speech after Nara-WPE dereverberation in Table 9.5. Using the RNN-based model, the WER of the evaluation set is high, at 28.21%, which is influenced greatly by the reverberation, even

Table 9.3: Performance in terms of average WER [%] on the spatialized **two-channel reverberant** WSJ-2Mix corpus.

Model	dev	eval
RNN-based MIMO-Speech (Chang et al., 2019c)	34.98	29.99
+ Transformer backend	32.95	28.01
++ Transformer frontend	**31.93**	**26.02**

Table 9.4: Performance in terms of average WER [%] on the spatialized **two-channel reverberant** WSJ-2Mix corpus **after Nara-WPE**.

Model	dev	eval
RNN-based MIMO-Speech	24.45	17.67
+ Transformer backend	**19.17**	**15.24**
++ Transformer frontend	20.55	15.46

when preprocessing with the dereverberation technique. However, the Transformer-based model can reach a final WER of 16.50%, a 41.5% relative reduction, proving that the Transformer-based model is more robust than the RNN-based model.

9.4 Conclusion

In this work, we applied Transformer models for end-to-end multi-speaker ASR in both the single-channel and multi-channel scenarios, and observed consistent improvements. The RNN-based ASR module is replaced with the Transformers. To alleviate the fatal memory consumption issue when applying Transformers in the frontend with considerably long sequences, we modified the self-attention in the Transformers of the masking network by using a local context window. Furthermore, by incorporating an external dereverberation method, we largely reduced the performance gap between the reverberant condition and the anechoic condition, and hope to further

Table 9.5: Performance in terms of average WER [%] on the **1st channel** of the spatialized **reverberant** WSJ-2Mix corpus **after Nara-WPE**.

Model	dev	eval
RNN-based 1-channel Model	31.21	28.21
Transformer-based 1-channel Model	20.44	16.50

reduce it in the future thanks to tighter integration of the dereverberation within our model.

Building upon the advancements made with the Transformer-based MIMO-Speech model, the next chapter introduces MIMO-IRIS, the ultimate model that combines elements from the previously proposed methods, including IRIS in Chapter 3 and MIMO-Speech. This integrated approach aims to leverage the strengths of each method to achieve superior performance in multi-channel multi-speaker ASR.

Chapter 10

MIMO-IRIS: Multi-Speaker E2E ASR with multi-channel Input robust to noise and reverberation

Summary

In Chapter 3 of this thesis, we explored the integration of self-supervised learning (SSL) models into an end-to-end (E2E) ASR system alongside speech enhancement techniques, yielding remarkable improvements in single-channel single-speaker ASR performance. This integrated approach leveraged SSL models trained on large-scale datasets, such as HuBERT and WavLM, to efficiently extract powerful contextual speech features. The inclusion of speech enhancement modules further mitigated the impacts of environmental noise and reverberations on speech recognition. Building upon the success observed in single-channel scenarios, we extended this approach to multi-channel input signals, where similar performance gains were observed (Masuyama et al., 2023a). By incorporating Wave-Field-Plane-Divergence (WPD)-based multi-channel neural beamforming techniques, the model demonstrated enhanced capabilities in handling reverberations and spatial information effectively, resulting in improved ASR performance.

In the previous chapters (Ch. 8 and Ch. 9), we introduced the MIMO-Speech for multi-channel input and multi-speaker output scenario. In this chapter, we combine these techniques and propose our most powerful ASR model. In both the RNN-based and Transformer-based MIMO-Speech models, the speech separation and enhancement frontend is based on neural beamformer without explicit criterion for separation. In this chapter, we also explored other speech enhancement and separation methodologies tailored for multi-channel input. These techniques can facilitate more accurate and robust multi-speaker ASR, by extracting and separating individual speaker signals

from overlapping speech in multi-channel scenarios. Similar to the previous MIMO models, the proposed model follows the design E2E modular-based, as described in Sec. 2.

Our preliminary results in this direction showcase promising advancements in multi-speaker speech recognition using multi-channel input signals. Moving forward, we aim to refine and optimize these methodologies to further enhance the performance and applicability of our proposed approach in real-world multi-speaker environments.

> Masuyama, Yoshiki*, Chang, Xuankai*, Zhang, Wangyou, Cornell, Samuele, Wang, Zhong-Qiu, Ono, Nobutaka, Qian, Yanmin, and Watanabe, Shinji. IEEE WASPAA 2023. Exploring the Integration of Speech Separation and Recognition with Self-Supervised Learning Representation.

10.1 Introduction

Speech separation and enhancement (SSE) is a crucial front-end for various applications such as speaker diarization, automatic speech recognition (ASR), and spoken language understanding (Ryant et al., 2021; Raj et al., 2021; Li et al., 2017; Lu et al., 2022a). The speech separation field has been revolutionized recently by the invention of deep clustering (Hershey et al., 2016a) and permutation invariant training (PIT) (Yu et al., 2017b), which allow us to train fully supervised speech separation models based on deep neural networks (DNNs). Previous speech separation methods based on time-frequency (T-F) masking (Hershey et al., 2016a; Wang et al., 2018; Yu et al., 2017b; Wang and Chen, 2018b) used a DNN to estimate the T-F mask for each speaker from the short-time Fourier transform (STFT) of the observed mixture. Meanwhile, time-domain methods (Luo and Mesgarani, 2019b; Luo et al., 2020; Subakan et al., 2021) have demonstrated promising results by directly processing time-domain signals in an end-to-end (E2E) manner. Very recently, fully complex STFT-domain methods have been proven to be extremely effective (Williamson et al., 2015; Yang et al., 2022a; Tan et al., 2022; Wang et al., 2022). In particular, TF-GridNet (Wang et al., 2022) has achieved state-of-the-art (SotA) performance on several SSE benchmarks (Hershey et al., 2016a; Wang et al., 2018; Maciejewski et al., 2020; Guizzo et al., 2022), including both monaural and multi-channel cases. Despite these impressive recent improvements in separation performance, it is still unclear how and if these can also lead to better ASR performance.

Most conventional SSE models are trained to minimize signal-level differences between separated and target speech, especially with scale-invariant signal-to-distortion ratio (Luo and Mesgarani, 2019b; Luo et al., 2020). This could lead to mismatches with respect to the subsequent ASR task. To address this issue, several attempts (Seltzer et al., 2004; Li et al., 2016; Heymann et al., 2017; Ochiai et al., 2017a; Minhua et al., 2019; Chang et al., 2019c; Zhang et al., 2021a; von Neumann et al., 2020a) have been made by integrating SSE models with ASR models through

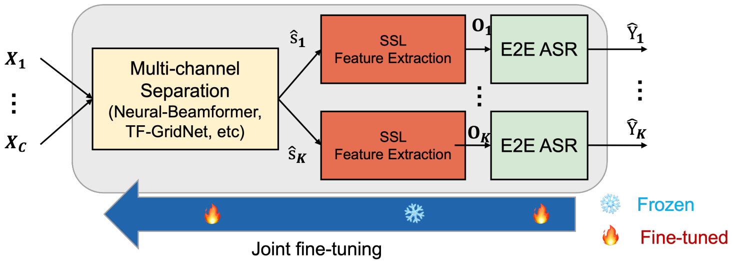


Figure 10.1: Overview of our E2E integration. We pre-train speech separation, SSLR, and ASR models separately, and fine-tune the speech separation and ASR models jointly while freezing WavLM.

joint optimization methods. For robust ASR, a neural beamformer and a joint connectionist temporal classification (CTC)/attention-based encoder-decoder were integrated and optimized with the ASR objectives (Ochiai et al., 2017a). Later, the integration was extended to multi-speaker settings, such as MIMO-Speech (Chang et al., 2019c). This approach aims to directly enhance the performance of multi-speaker ASR while being more explainable than a fully E2E black-box approach (Seki et al., 2018; Kanda et al., 2020b; Sklyar et al., 2021) as the front-end and back-end remain separate. In fact, the separated speech from the neural beamformer achieves a good separation quality (Chang et al., 2019c), although the model was not explicitly optimized with any signal-level criterion.

Self-supervised learning (SSL) models such as Wav2Vec 2.0 (Baevski et al., 2020), HuBERT (Hsu et al., 2021a), and WavLM (Chen et al., 2021b) have shown considerable potential in a wide range of speech processing tasks (Yang et al., 2021; Tsai et al., 2022). Recently, IRIS (Chang et al., 2022) demonstrated impressive results with an E2E model that integrates monaural speech enhancement, WavLM, and ASR models. MultiIRIS (Masuyama et al., 2023a) expanded IRIS to include multi-channel speech enhancement and demonstrated that joint training with an ASR objective could further improve ASR performance under noisy and reverberant conditions.

Building upon MultiIRIS, this work investigates MIMO-IRIS: an integration of speech separation, SSLR, and ASR for multi-channel multi-speaker overlapping scenarios. We explore the combination of SSL-based ASR models (Chang et al., 2021) with TF-GridNet (Wang et al., 2022) as well as well-established beamforming techniques as illustrated in Fig. 10.1. We perform an extensive experimental validation on the spatialized WSJ0-2mix (Wang et al., 2018) and WHAMR! (Maciejewski et al., 2020) datasets, assessing both separation and ASR performance. This allows us to investigate the correlation between the two. Interestingly, our experiments show that the correlation between speech separation and ASR performance is not precisely positive. We find that the separation performance after fine-tuning degrades while the word error rate (WER) decreases. This is especially true for TF-GridNet-based complex spectral mapping, while mask-based beamforming (Heymann et al., 2016; Erdogan et al., 2016) results in less degradation. Despite this, our best MIMO-IRIS model after joint training achieves SotA ASR performance on the WHAMR! dataset with a WER of 2.6%, comparable to SotA results on clean single-speaker WSJ evaluation sets (Chang et al., 2021). Audio examples of our system are available at u18081971.github.io/MIMO-IRIS-demo.

10.2 Method

Given an L-sample, C-channel mixture signal $\mathbf{X} = (\mathbf{x}_c)_{c=1}^{C} \in \mathbb{R}^{C \times L}$ consisting of K speakers and noises $\mathbf{N} = (\mathbf{n}_c)_{c=1}^{C}$, we formulate the mixing process as follows:

$$\mathbf{x}_c = \sum_{k=1}^{K} \mathbf{s}_{k,c} + \mathbf{n}_c, \tag{10.1}$$

where $\mathbf{s}_{k,c} \in \mathbb{R}^L$ is the source image of speaker k at microphone c. For each speaker k, the transcription sequence is denoted as $\mathbf{R}_k$. This section describes each part of the proposed E2E system, depicted in Fig. 10.1, including speech separation, SSLR Extraction, and E2E ASR.

Speech Separation

The goal of speech separation is to estimate each speaker's signal $\hat{\mathrm{s}}_{k,r}$ at a reference microphone $r \in \{1, \ldots, C\}$ from the mixture $\mathbf{X}$, which can be written as:

$$\{\widehat{\mathbf{s}}_1, \ldots, \widehat{\mathbf{s}}_K\} = \mathrm{SS}(\mathbf{X}). \tag{10.2}$$

Depending on the number of input microphones, the task can be divided into monaural and multi-channel speech separation.

Monaural speech separation

While our main focus is on multi-channel speech separation, we briefly explain monaural speech separation as TF-GridNet was originally proposed for the monaural case. In monaural speech separation, masking and mapping are two popular approaches (Wang and Chen, 2018b). Both can be performed in the complex T-F domain or in the time domain.

In masking-based approaches, a DNN is trained to estimate a mask for each speaker, and the mask is point-wisely applied to the encoded representation of the mixture $\mathbf{X}$:

$$\mathbf{Z} = \text{SSEnc}(\mathbf{X}), \tag{10.3}$$

$$\{\widehat{\mathbf{G}}_1, \ldots, \widehat{\mathbf{G}}_K\} = \text{MaskEstimationNet}(\mathbf{Z}), \tag{10.4}$$

$$\widehat{\mathbf{S}}_k = \widehat{\mathbf{G}}_k \odot \mathbf{Z}, \tag{10.5}$$

$$\widehat{\mathbf{s}}_k = \text{SSDec}(\widehat{\mathbf{S}}_k), \tag{10.6}$$

where $\widehat{\mathbf{G}}_k$ denotes the estimated mask for speaker k, and $\odot$ denotes the Hadamard product. In T-F masking, SSEnc and SSDec can respectively be STFT and inverse STFT. Meanwhile, they are usually trainable one-dimensional convolutional layers and deconvolutional layers in the time-domain methods.

In mapping-based approaches, a DNN is trained to directly predict the encoded representation of each speaker. In detail, Eq. 10.4 and Eq. 10.5 are replaced by

$$\{\widehat{\mathbf{S}}_1, \ldots, \widehat{\mathbf{S}}_K\} = \text{MappingNet}(\mathbf{Z}). \tag{10.7}$$

Very recently, mapping-based approaches in the T-F domain, or complex spectral mapping, have gained increasing attention due to the appearance of powerful DNN architecture called TF-GridNet (Wang et al., 2022). In detail, TF-GridNet predicts the complex STFT coefficients of each speaker from those of the observed mixture. TF-GridNet has outperformed the best time-domain masking-based methods (Subakan et al., 2021). Furthermore, it has been successfully adapted to multi-channel speech separation.

Multi-channel speech separation

Multi-channel speech separation takes advantage of spatial information afforded by multiple microphones and has been used in robust ASR (Li et al., 2017; Heymann et al., 2016; Erdogan et al., 2016). For the purpose of robust ASR, two popular approaches have been developed multi-channel separation: using DNN estimates to derive a conventional beamformer and using DNN to directly estimate each speaker's signal.

In the first approach, the minimum variance distortionless response (MVDR) beamformer has been widely used due to its distortionless property and generalization capability (Gannot et al., 2017; Heymann et al., 2016; Erdogan et al., 2016; Yoshioka et al., 2018b). It incurs few processing artifacts by using the constrained time-invariant linear filters and is a preferable front-end of ASR backends (Chang et al., 2019c; Zhang et al., 2021a). Neural mask-based beamforming estimates a T-F mask for each speaker, denoted as $\widehat{\mathbf{G}}_k$ for speaker k, and computes a spatial covariance matrix for each speaker:

$$\widehat{\mathbf{V}}_k[f] = \frac{1}{\sum_t \widehat{G}_k[t,f]} \sum_{t=1}^{T} \widehat{G}_k[t,f]\mathbf{z}[t,f]\mathbf{z}[t,f]^{\mathsf{H}}, \tag{10.8}$$

where $\mathbf{z}[t,f] = [Z_1[t,f], \ldots, Z_C[t,f]]^{\mathsf{T}}$, $Z_c[t,f]$ is the STFT coefficient of $\mathbf{x}_c$, $(\cdot)^{\mathsf{T}}$ denotes the transpose, and $(\cdot)^{\mathsf{H}}$ denotes the Hermitian transpose. An MVDR beamformer $\widehat{\mathbf{w}}_k[f]$ is then computed as follows:

$$\hat{\mathbf{w}}_k[f] = \frac{\widehat{\mathbf{V}}_{\backslash k}^{-1}[f]\widehat{\mathbf{V}}_k[f]}{\text{trace}\left(\widehat{\mathbf{V}}_{\backslash k}^{-1}[f]\widehat{\mathbf{V}}_k[f]\right)}\mathbf{u}, \tag{10.9}$$

where $\widehat{\mathbf{V}}_{\backslash k}[f]$ denotes the sum of the spatial covariance matrices of the noises and all the speakers except speaker k, and $\mathbf{u} \in \mathbb{R}^C$ is a one-hot vector with the element corresponding to the reference microphone being one. The beamforming output is computed as:

$$\widehat{S}_k[t,f] = \widehat{\mathbf{w}}_k^{\mathsf{H}}[f]\mathbf{z}[t,f], \tag{10.10}$$

and converted to the time domain via inverse STFT as in Eq. 10.6.

In the second approach, a DNN directly estimates the encoded representation of each speaker by replacing the input of Eq. 10.7 to the concatenation of the encoded representation of microphone c. Compared to the output of linear beamformers, the output of the second approach tends to have fewer non-target signals but more distortion on the target speech. Although earlier studies suggested that linear beamformers would be preferable for robust ASR (Chang et al., 2019c; Zhang et al., 2021a), modern ASR back-ends and separation front-ends have become much more powerful nowadays. Hence, we expect that modern back-ends could handle speech distortion in separated signals, and modern speech separation models can produce much less distortion in separated signals. We will compare their performance in our experiments, where TF-GridNet (Wang et al., 2022) and a strong back-end (Kim et al., 2017b) are used for speech separation and ASR, respectively.

SSLR Extraction and E2E-ASR

We extract SSLR from each separated signal $\widehat{\mathrm{s}}_k$ in Eq. 10.2 and pass it to E2E-ASR in the same way as in previous studies (Chang et al., 2022; Masuyama et al., 2023a):

$$\widehat{\mathbf{Y}}_k = \text{ASR}(\text{SSLR}(\widehat{\mathrm{s}}_k;\, \theta^{\text{ssl}});\, \theta^{\text{asr}}), \tag{10.11}$$

where θ^{ssl} and θ^{asr} represent the parameters of the SSLR extractor $\text{SSLR}(\cdot)$ and ASR model $\text{ASR}(\cdot)$, respectively. Specifically, WavLM (Chen et al., 2021b) is used to extract robust SSLR by applying the weighted sum of all transformer encoder embeddings. The weights are optimized with the following ASR model during training. E2E-ASR is based on the joint CTC/attention-based encoder-decoder framework (Kim et al., 2017b).

MIMO-IRIS: Integration of Separation, SSLR and ASR

To recognize multi-speaker speech, one can directly send the outputs of the speech separation model to a pre-trained ASR model. This solution is, however, not optimal because ASR models are typically trained with single-speaker speech, while the separated speech usually contain residual interference. Following IRIS (Chang et al., 2022) and MultiIRIS (Masuyama et al., 2023a), we integrate the speech separation model, SSLR extractor, and E2E-ASR model into a single model as shown in Fig. 10.1. The speech separation model can generate multiple streams, one for each speaker, and the ASR model is shared among all separated streams along with the SSLR extractor. During the training, to solve the permutation problem, PIT is applied to the CTC loss in the ASR model to determine the optimal permutation. The following attention-based decoder uses this permutation to select the corresponding reference transcript for each input stream in the teacher-forcing training. Our E2E model can be extended from Eq. 10.11 as:

$$\{\hat{\mathbf{Y}}_1, \dots, \hat{\mathbf{Y}}_K\} = \text{ASR}(\text{SSLR}(\text{SS}(\mathbf{X}; \theta^{\text{ss}}); \theta^{\text{ssl}}); \theta^{\text{asr}}), \tag{10.12}$$

where θ^{ss} represents the parameters of the speech separation model, as discussed in Section 10.2. The loss function of the ASR task is the same as in MIMO-Speech (Chang et al., 2019c). We omit the details here.

The E2E model could be trained from scratch with multi-task learning, including speech separation and ASR objectives. However, such a model has a large footprint and requires intensive computation. In addition, previous studies on the integration of speech enhancement, SSLR, and E2E ASR reported that the integrated model resulted in sub-optimal performance when trained from scratch (Chang et al., 2022; Masuyama et al., 2023a). We thus propose a two-stage approach. First, the speech separation model is pre-trained on commonly-used speech separation datasets, e.g., spatialized WSJ0-2mix (Hershey et al., 2016a; Wang and Chen, 2018b) and WHAMR! (Maciejewski et al., 2020). Second, the ASR model is pre-trained on monaural clean speech datasets, e.g., the WSJ corpus. Finally, the entire integrated model is fine-tuned with the ASR objective, as shown in Fig. 10.1. Following previous studies, we freeze the WavLM, which is pre-trained on a large amount of external data. This strategy is efficient and requires only a few optimization epochs to achieve excellent performance in speech enhancement (Chang et al., 2022; Masuyama et al., 2023a).

10.3 Experiment

We validate the effectiveness of our integration on two-speaker mixtures under anechoic/reverberant and clean/noisy conditions. Our experiments were conducted using the ESPnet-SE++ toolkit (Lu et al., 2022a).

Datasets

We evaluated our systems on the spatialized WSJ0-2mix (Wang et al., 2018) and WHAMR!(Maciejewski et al., 2020) datasets, mentioned in Sec. 1.4.4. Both of the corpora support anechoic and reverberant two-speaker mixture simulations. The training, validation, and test sets of both datasets contain 20,000, 5,000, and 3,000 mixtures, respectively. Room impulse responses were simulated and convolved with dry source signals from WSJ0-2mix (Hershey et al., 2016a). The signal-to-distortion ratio (SDR) (Vincent et al., 2006) with respect to the input mixture is 0.07 dB in spatialized WSJ0-2mix. WHAMR! (Maciejewski et al., 2020) is one of the most challenging datasets for speech separation, as it contains two-channel real-recorded environmental noise. For WHAMR!, the SDR with respect to the input mixture is -4.61 dB. To leverage the pre-trained WavLM (Chen et al., 2021b), which was trained on 16 kHz, we used the 16 kHz version of both datasets in our experiments. We combined both anechoic and reverberant conditions of the training and validation sets to form the new training and validation sets, respectively.

Training Configurations

The ASR model (ASR$(\cdot)$ in Eq. 10.11 and Eq. 10.12) consists of a Conformer-based encoder of 12 layers and a Transformer-based decoder of 6 layers by following a previous study (Masuyama et al., 2023a). The encoder and decoder have 2,048 hidden units and 4 attention heads. We reduced the dimensions of the speaker-wise SSLR from 1,024 to 80 by a fully-connected layer before feeding it to the ASR model. The ASR model and the learnable weight for the WavLM embeddings were pre-trained on the clean WSJ corpus. We used the Adam optimizer with a warm-up and the peak learning rate of 1.0×10^{-3}. During inference, we also used a Transformer-based character-level language model. On the clean single-speaker WSJ evaluation set, the ASR model achieved a WER of 1.3%.

As the speech separation model (SS$(\cdot)$ in Eq. 10.2 and Eq. 10.12), our mask-based MVDR beamformer employed a 3-layer bidirectional long short-term memory of 512 units with a projection layer to estimate the T-F masks as in (Chang et al., 2019c; Zhang et al., 2022b). STFT was implemented with the Hann window of 512 samples with a 128-sample shift. The mask estimation network was optimized with the convolutive transfer function invariant signal-to-distortion ratio (CI-SDR) loss (Boeddeker et al., 2021) on beamforming outputs. Meanwhile, TF-GridNet consists of 6 blocks, where the TF-unit embedding dimension was 48. To reduce the computation, we increased the window shift size to 256 samples in STFT. TF-GridNet was optimized with a sum of the L_1 loss on the waveform and on the STFT magnitude with a scaling factor[1], following (Lu et al., 2022a). Both mask estimation network and TF-GridNet were pre-trained with the Adam optimizer. Then, the joint fine-tuning was performed using the stochastic gradient descent method with a learning rate of 1.0×10^{-3} and momentum of 0.9. We used the *max* condition of the spatialized WSJ0-2mix and WHAMR! datasets, mixtures of the non-trimmed utterances, in the joint fine-tuning of the speech separation and ASR models.

Results on Clean Multi-channel Speech Separation

Table 10.1 presents the results on the spatialized WSJ0-2mix dataset. First, we show the results of the monaural TF-GridNet and ASR performance in a *cascaded* manner, achieving an SDR of 19.4 dB and a WER of 4.8%. We then show the results in multi-channel cases, where the mask-based MVDR beamformer and TF-GridNet-based complex spectral mapping were fine-tuned with the ASR objective. The TF-GridNet model consistently outperformed the MVDR beamformer not only in terms of separation performance but also in terms of WERs. This result demonstrates that the unconstrained complex spectral mapping is advantageous as an ASR front-end when using modern speech separation models. Furthermore, even the monaural TF-GridNet is more effective than the MVDR beamformer without joint fine-tuning. That is, the monaural TF-GridNet can avoid severe distortion of the target signals without any constraints. To clarify the effectiveness of WavLM as a robust SSLR extractor, we evaluated the ASR model using filterbank features without joint fine-tuning. According to the bottom row of Table 10.1, its WER was degraded to 28.2% from 2.4% with the WavLM in the reverberant condition. This result confirms the importance of the robust SSLR even with the powerful complex spectral mapping.

As an interesting finding, joint fine-tuning further reduced the WERs in both anechoic and reverberant conditions while degrading the separation performance. This degradation was less severe for the MVDR beamforming as the output is constrained to be distortion-less. Meanwhile, TF-GridNet-based unconstrained complex spectral mapping faced severe performance degradation, despite the better WER. In the anechoic case, the multi-channel TF-GridNet can achieve an SDR of 27.01 dB and a WER of 3.2% without fine-tuning. However, the separation performance dropped to 16.09 dB after joint fine-tuning. Examples of spectrogram and audio are available at

[1] In our preliminary experiments, we also used the loss presented in (Lu et al., 2022a) to train the mask-based beamformer. This resulted in worse WERs on the validation sets than using the CI-SDR loss (Boeddeker et al., 2021)

Table 10.1: Separation and WER results on single-channel WSJ0-2mix and spatialized WSJ0-2mix.

	SDR [dB]	PESQ	STOI	WER (%)
		Monaural		
TF-GridNet*	19.40	3.41	0.976	4.8
		Anechoic eight-channel		
MVDR (**proposed**)	12.83	3.86	0.987	2.1
- w/o fine-tuning	14.53	3.90	0.989	7.8
TF-GridNet (**proposed**)	16.09	3.20	0.983	**1.9**
- w/o fine-tuning	**27.01**	**4.10**	**0.995**	3.2
- w/o WavLM				6.3
		Reverberant eight-channel		
MVDR (**proposed**)	4.56	2.76	0.859	3.6
- w/o fine-tuning	5.11	2.76	0.864	30.5
TF-GridNet (**proposed**)	12.96	3.22	0.959	**1.9**
- w/o fine-tuning	**19.2**	**3.88**	**0.982**	2.4
- w/o WavLM				28.2

* The monaural TF-GridNet was not jointly fine-tuned.

u18081971.github.io/MIMO-IRIS-demo. Investigation of the degradation is included in our future work.

Results on Noisy Multi-channel Speech Separation

In this section, we present our experimental results of the WHAMR! dataset, which are summarized in Table10.2. In the top panel, we report the performance of monaural TF-GridNet on both noisy anechoic and reverberant conditions. As with the results on the spatialized WSJ0-2mix, the monaural TF-GridNet outperformed the mask-based MVDR beamformer integrated with weighted prediction error dereverberation (Zhang et al., 2020a). The difference is even more significant due to the limitation of the number of microphones and noisy/reverberant characteristics of the data. The best model overall is again the multi-channel TF-GridNet, which reached the best signal-level metrics before fine-tuning. After fine-tuning, the SDR decreased significantly, but the WER improved by over 400% relative factor in noisy/reverberant conditions. The performance is outstanding with WERs of 2.3% and 2.6% in anechoic and reverberant conditions, respectively, which are close to the performance achieved on the clean WSJ dataset. We emphasize that the ASR performance without fine-tuning still outperformed the previous MIMO-Speech (Zhang et al., 2020a) and the cascade combination of the time-domain speech separation and ASR models (Zhang et al., 2021b).

Table 10.2: Separation and WER results on WHAMR!.

	Noisy/Anechoic		Noisy/Reverberant	
	SDR [dB]	WER (%)	SDR [dB]	WER (%)
Monaural				
TF-GridNet*	9.27	14.5	9.07	18.3
Two-channel				
MIMO-Speech (Zhang et al., 2022b)	-	-	-2.27	28.9
Time-domain (Zhang et al., 2021b)	-	-	-	20.9
MVDR (**proposed**)	-1.42	42.2	-1.30	44.4
TF-GridNet (**proposed**)	9.29	**2.3**	7.96	**2.6**
- w/o fine-tuning	**12.74**	7.4	**10.82**	11.1

* The monaural TF-GridNet was not jointly fine-tuned.

10.4 Conclusion

In this chapter, we investigated the integration of speech separation, SSLR, and ASR with well-established beamforming techniques as well as the latest SotA techniques including TF-GridNet. We performed our experiments under anechoic/reverberant and clean/noisy conditions using the spatialized WSJ0-2mix and WHAMR! datasets. In detail, we explored how both separation performance and WER are affected when joint fine-tuning is performed. Our experimental results show that the purely DNN-based speech separation method, TF-GridNet-based complex spectral mapping, can considerably outperform the mask-based MVDR beamforming preferred as an ASR front-end. Joint fine-tuning degraded the separation performance while significantly improving the WER, which is inconsistent with the tendency reported in a speech enhancement paper (Masuyama et al., 2023a). Overall our best system, based on multi-channel TF-GridNet, WavLM, and E2E ASR, was able to reach performance on par with the one achieved on clean, single-speaker WSJ (Chang et al., 2021).

With the MIMO-IRIS model, we have achieved superior ASR performance for the Multi-Input Multi-Output scenario, concluding our investigations on MIMO. Moving forward, we can further explore combining approaches from previous chapters to tackle real conversational speech recognition. This is just one possible direction for continued research. However, we may also need to rely on more recent and innovative methods to achieve our targets.

Chapter 11

Thesis Conclusion and Future Work

In this chapter, I will conclude my thesis and discuss the future directions of everyday conversational speech recognition systems based on End-to-End neural networks.

11.1 Thesis Conclusion

This thesis endeavors to confront the obstacles associated with recognizing everyday conversational speech by employing end-to-end neural network models. Various factors inherent in conversational speech pose challenges to ASR performance, encompassing speech quality, overlapping segments, and speaking styles, among others. Throughout this thesis, the primary emphasis has been on mitigating the impacts of environmental noise, reverberations, and overlapping speech.

In the first section of the thesis, we focus on enhancing ASR performance in the presence of environmental noise (SISO). To tackle this challenge, we integrated self-supervised learning, trained on a very large scale dataset, into end-to-end ASR models, referred to as IRIS in Ch. 3. Additionally, we introduced a speech enhancement module and integrated it into a joint-training framework. Furthermore, we devised an efficient training algorithm to facilitate the stable training of this integrated system. The resulting system elevated ASR performance on noisy speech to unprecedented levels.

Moving forward, inspired by recent advancements in speech foundation models like Whisper (Radford et al., 2023), we extended our approach to accept multi-channel inputs in real-world applications. Unlike the modular-based design used in the SISO case, our proposed model in Chapter 4 leverages a data-driven approach, providing a compelling alternative for enhancing model capabilities.

In the third part of the thesis, we introduced various model architectures aimed at addressing single-channel overlapping speech (SIMO). Operating under the premise of known overlapping segments and speaker count, we enhanced the vanilla joint CTC/AED architecture through permutation invariant training (PIT) in Ch. 5. This demonstrated that end-to-end neural networks can achieve reasonable performance in recognizing overlapping speech. Subsequently, we advanced to tackle the challenge of unknown speaker counts. Leveraging the conditional-chain model in Ch. 6, we sequentially recognized utterances from different speakers, utilizing previously recognized words as memory to prevent redundant efforts. Finally, we proposed a novel approach for representing reference transcriptions and speaker identities in overlapping speech scenarios. This representation eliminates the need for assumptions and provides supervision signals for model training, alongside introducing a new training criterion termed extended graphical temporal classification (GTCe) in Ch. 7.

Lastly, we target at the multi-channel input multi-speaker speech (MIMO) with environmental noise and reverberation. We designed a new end-to-end ASR framework, called MIMO-Speech in Ch. 8, to perform multi-channel speech separation and recognition. Combining the techniques we proposed for SISO and the MIMO-Speech, we achieved a powerful model called MIMO-IRIS in Ch. 10. MIMO-IRIS reaches promising performance in multi-channel overlapping speech recognition with noise and reverberation, very close to the ASR performance on the corresponding clean speech counterpart.

In summary, this thesis represents a comprehensive exploration of advanced techniques and methodologies to overcome the challenges of recognizing conversational speech. Our work paves the way for robust and efficient ASR systems in real-world settings, demonstrating the potential of end-to-end neural network models in handling complex acoustic and linguistic properties of everyday speech. Through these advancements, we aim to contribute to the development of more capable and adaptable ASR systems for diverse applications.

11.2 Future Work

As we conclude this thesis, several avenues emerge for further exploration and advancement in the field of conversational speech recognition. These directions hold the potential to inspire future research endeavors and foster innovation in the realm of ASR technology.

One promising avenue for future research involves leveraging and integrating knowledge from diverse data sources to advance conversational speech recognition. Beyond speech-specific data, incorporating general audio tasks, such as audio captioning (Drossos et al., 2017) or sound event detection (Barchiesi et al., 2015), can provide valuable insights into environmental contexts and interference signals, enriching the understanding of conversational dynamics. This expanded data availability can support the exploration and training of large foundation models.

Furthermore, integrating visual input alongside auditory signals holds promise for deeper comprehension of conversational interactions. Visual cues, such as speaker emotions, actions, locations, and environmental factors, offer valuable context that can enhance ASR accuracy and contextual understanding.

Moreover, the emergence of large language models (LLMs) presents an exciting opportunity to augment conversational speech recognition. Trained on extensive corpora of formal and informal text data, LLMs offer rich knowledge that can address challenges such as transcription variability, disfluencies, and context understanding. One notable limitation in this thesis is the lack of the capability for long-form continuous conversation ASR. The ability of LLMs to manage large context windows is particularly beneficial for long-form speech recognition and transcription fusion. By incorporating insights from LLMs, researchers can elevate ASR quality and develop more robust and contextually aware systems. LLMs can support flexible natural language-based prompts that allow for steering the target source or modifying transcriptions.

By synthesizing insights from these complementary data sources, researchers can unlock new capabilities and drive advancements in ASR technology, ultimately enabling more accurate and adaptive systems for diverse conversational contexts. This multidimensional approach to data integration holds great promise for pushing the boundaries of conversational speech recognition and enhancing user experiences across various applications.

Bibliography

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182. PMLR.

Xavier Anguera, Chuck Wooters, and Javier Hernando. 2007. Acoustic beamforming for speaker diarization of meetings. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):2011–2022.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Proc. NeurIPS*, 33:12449–12460.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. ICLR*.

Daniele Barchiesi, Dimitrios Giannoulis, Dan Stowell, and Mark D Plumbley. 2015. Acoustic scene classification: Classifying environments from the sounds they produce. *IEEE Signal Processing Magazine*, 32(3):16–34.

Jon Barker, Ricard Marxer, Emmanuel Vincent, and Shinji Watanabe. 2017. The third 'chime'speech separation and recognition challenge: Analysis and outcomes. *Computer speech & language*, 46:605–626.

Jon Barker, Shinji Watanabe, Emmanuel Vincent, and Jan Trmal. 2018. The fifth 'CHiME' speech separation and recognition challenge: Dataset, task and baselines. *arXiv preprint arXiv:1803.10609*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Proc. Conference on Neural Information Processing Systems (NIPS)*, pages 1171–1179.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proc. International Conference on Machine Learning (ICML)*, pages 41–48.

Christoph Boeddeker, Wangyou Zhang, Tomohiro Nakatani, Keisuke Kinoshita, Tsubasa Ochiai, Marc Delcroix, Naoyuki Kamo, Yanmin Qian, and Reinhold Haeb-Umbach. 2021. Convolutive transfer function invariant sdr training criteria for multi-channel reverberant speech separation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8428–8432. IEEE.

Stefan Braun, Daniel Neil, Jithendar Anumula, Enea Ceolini, and Shih-Chii Liu. 2018. Multi-channel attention for end-to-end speech recognition. In *Proc. ISCA Interspeech*, pages 17–21.

Jean Carletta. 2006. Announcing the ami meeting corpus. *The ELRA Newsletter*, 11(1):3–5.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al. 2005. The AMI meeting corpus: A pre-announcement. In *Proc. International Workshop on Machine Learning for Multimodal Interaction (MLMI)*, pages 28–39.

Özgür Çetin and Elizabeth Shriberg. 2006. Analysis of overlaps in meetings by dialog factors, hot spots, speakers, and collection site: Insights for automatic speech recognition. In *Proc. ISCA Interspeech*.

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964.

William Chan, Daniel Park, Chris Lee, Yu Zhang, Quoc Le, and Mohammad Norouzi. 2021. SpeechStew: Simply mix all available speech recognition data to train one large neural network. In *Proc. Interspeech*.

William Chan, Chitwan Saharia, Geoffrey Hinton, Mohammad Norouzi, and Navdeep Jaitly. 2020. Imputer: Sequence modelling via imputation and dynamic programming. In *Proc. ICML*, pages 1403–1413. PMLR.

Xuankai Chang, Takashi Maekaku, Yuya Fujita, and Shinji Watanabe. 2022. End-to-end integration of speech recognition, speech enhancement, and self-supervised learning representation. *arXiv preprint arXiv:2204.00540*.

Xuankai Chang, Takashi Maekaku, Pengcheng Guo, Jing Shi, Yen-Ju Lu, Aswin Shanmugam Subramanian, Tianzi Wang, Shu-wen Yang, Yu Tsao, Hung-yi Lee, et al. 2021. An exploration of self-supervised pretrained representations for end-to-end speech recognition. In *Proc. ASRU*.

Xuankai Chang, Yanmin Qian, and Dong Yu. 2018a. Adaptive permutation invariant training with auxiliary information for monaural multi-talker speech recognition. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Xuankai Chang, Yanmin Qian, and Dong Yu. 2018b. Monaural multi-talker speech recognition with attention mechanism and gated convolutional networks. In *Proc. ISCA Interspeech*, pages 1586–1590.

Xuankai Chang, Yanmin Qian, Kai Yu, and Shinji Watanabe. 2019a. End-to-end monaural multi-speaker ASR system without pretraining. In *Proc. ICASSP*, pages 6256–6260.

Xuankai Chang, Yanmin Qian, Kai Yu, and Shinji Watanabe. 2019b. End-to-end monaural multi-speaker ASR system without pretraining. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6256–6260.

Xuankai Chang, Wangyou Zhang, Yanmin Qian, Jonathan Le Roux, and Shinji Watanabe. 2019c. MIMO-Speech: End-to-end multi-channel multi-speaker speech recognition. In *Proc. ASRU*, pages 237–244. IEEE.

Xuankai Chang, Wangyou Zhang, Yanmin Qian, Jonathan Le Roux, and Shinji Watanabe. 2020. End-to-end multi-speaker speech recognition with Transformer. In *Proc. ICASSP*, pages 6134–6138. IEEE.

Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, et al. 2021a. GigaSpeech: An evolving, multi-domain ASR corpus with 10,000 hours of transcribed audio. In *Proc. Interspeech*.

Nanxin Chen, Shinji Watanabe, Jesús Villalba, and Najim Dehak. 2019. Listen and fill in the missing letters: Non-autoregressive Transformer for speech recognition. *arXiv preprint arXiv:1911.04908*.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. 2021b. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *arXiv preprint arXiv:2110.13900*.

Szu-Jui Chen, Aswin Shanmugam Subramanian, Hainan Xu, and Shinji Watanabe. 2018a. Building state-of-the-art distant speech recognition using the chime-4 challenge with a setup of speech enhancement baseline. *Proc. Interspeech*, pages 1571–1575.

Zhehuai Chen, Jasha Droppo, Jinyu Li, and Wayne Xiong. 2017. Progressive joint modeling in unsupervised single-channel overlapped speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(1):184–196.

Zhehuai Chen, Qi Liu, Hao Li, and Kai Yu. 2018b. On modular training of neural acoustics-to-word model for LVCSR. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4819–4823.

Zhuo Chen, Takuya Yoshioka, Liang Lu, Tianyan Zhou, Zhong Meng, Yi Luo, Jian Wu, Xiong Xiao, and Jinyu Li. 2020. Continuous speech separation: Dataset and analysis. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7284–7288.

E Colin Cherry. 1953. Some experiments on the recognition of speech, with one and with two ears. *The Journal of the Acoustical Society of America*, 25(5):975–979.

Ethan A Chi, Julian Salazar, and Katrin Kirchhoff. 2021. Align-refine: Non-autoregressive speech recognition via iterative realignment. In *Proc. NAACL*, pages 1920–1927. ACL.

Chung-Cheng Chiu, James Qin, Yu Zhang, Jiahui Yu, and Yonghui Wu. 2022. Self-supervised learning with random-projection quantizer for speech recognition. In *International Conference on Machine Learning*, pages 3915–3924. PMLR.

Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al. 2018. State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4774–4778. IEEE.

Seungjin Choi, Andrzej Cichocki, Hyung-Min Park, and Soo-Young Lee. 2005. Blind source separation and independent component analysis: A review. *Neural Information Processing-Letters and Reviews*, 6(1):1–57.

Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. End-to-end continuous speech recognition using attention-based recurrent NN: first results. *arXiv preprint arXiv:1412.1602*.

Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. *Advances in neural information processing systems*, 28.

Israel Cohen. 2003. Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging. *IEEE Transactions on Speech and Audio Processing*, 11(5):466–475.

Martin Cooke, John R Hershey, and Steven J Rennie. 2010. Monaural speech separation and recognition challenge. *Computer Speech & Language*, 24(1):1–15.

Samuele Cornell, Matthew Wiesner, Shinji Watanabe, Desh Raj, Xuankai Chang, Paola Garcia, Yoshiki Masuyama, Zhong-Qiu Wang, Stefano Squartini, and Sanjeev Khudanpur. 2023. The chime-7 dasr challenge: Distant meeting transcription with multiple devices in diverse scenarios. *arXiv preprint arXiv:2306.13734*.

Joris Cosentino, Manuel Pariente, Samuele Cornell, Antoine Deleforge, and Emmanuel Vincent. 2020. LibriMix: An open-source dataset for generalizable speech separation. *arXiv preprint arXiv:2005.11262*.

Marc Delcroix, Takuya Yoshioka, Atsunori Ogawa, Yotaro Kubo, Masakiyo Fujimoto, Nobutaka Ito, Keisuke Kinoshita, Miquel Espi, Shoko Araki, Takaaki Hori, et al. 2015. Strategies for distant speech recognitionin reverberant environments. *EURASIP Journal on Advances in Signal Processing*, 2015:1–15.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. ACL*, pages 4171–4186.

Linhao Dong, Shuang Xu, and Bo Xu. 2018. Speech-Transformer: A no-recurrence sequence-to-sequence model for speech recognition. In *Proc. ICASSP*, pages 5884–5888.

Konstantinos Drossos, Sharath Adavanne, and Tuomas Virtanen. 2017. Automated audio captioning with recurrent neural networks. In *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 374–378. IEEE.

Lukas Drude and Reinhold Haeb-Umbach. 2017. Tight integration of spatial and spectral features for bss with deep clustering embeddings. In *Proc. ISCA Interspeech*, pages 2650–2654.

Lukas Drude, Jahn Heymann, Christoph Boeddeker, and Reinhold Haeb-Umbach. 2018. NARA-WPE: A Python package for weighted prediction error dereverberation in Numpy and Tensorflow for online and offline processing. In *ITG Fachtagung Sprachkommunikation (ITG)*.

Jun Du, Yan-Hui Tu, Lei Sun, Feng Ma, Hai-Kun Wang, Jia Pan, Cong Liu, Jing-Dong Chen, and Chin-Hui Lee. 2016. The ustc-iflytek system for chime-4 challenge. *Proc. CHiME*, 4:36–38.

Hakan Erdogan, John R Hershey, Shinji Watanabe, Michael I Mandel, and Jonathan Le Roux. 2016. Improved MVDR beamforming using single-channel mask prediction networks. In *Proc. ISCA Interspeech*, pages 1981–1985.

Ruchao Fan, Wei Chu, Peng Chang, and Jing Xiao. 2020. CASS-NAT: CTC alignment-based single step non-autoregressive Transformer for speech recognition. *arXiv preprint arXiv:2010.14725*.

Yusuke Fujita, Naoyuki Kanda, Shota Horiguchi, Yawen Xue, Kenji Nagamatsu, and Shinji Watanabe. 2019. End-to-end neural speaker diarization with self-attention. *arXiv preprint arXiv:1909.06247*.

Yusuke Fujita, Shinji Watanabe, Shota Horiguchi, Yawen Xue, Jing Shi, and Kenji Nagamatsu. 2020. Neural speaker diarization with speaker-wise chain rule. *arXiv preprint arXiv:2006.01796*.

Sharon Gannot, Emmanuel Vincent, Shmulik Markovich-Golan, and Alexey Ozerov. 2017. A consolidated perspective on multimicrophone speech enhancement and source separation. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 25(4):692–730.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proc. EMNLP-IJCNLP*, pages 6112–6121. ACL.

Marjan Ghazvininejad, Omer Levy, and Luke Zettlemoyer. 2020. Semi-autoregressive training improves mask-predict decoding. *arXiv preprint arXiv:2001.08785*.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proc. ICML*, pages 369–376.

Alex Graves and Navdeep Jaitly. 2014. Towards end-to-end speech recognition with recurrent neural networks. In *Proc. International Conference on Machine Learning (ICML)*, pages 1764–1772.

Alex Graves, Abdelrahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Proc. ICASSP*, pages 6645–6649.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *Proc. ICLR*.

Jiatao Gu, Changhan Wang, and Jake Zhao. 2019. Levenshtein Transformer. In *Proc. NeurIPS*, pages 11181–11191.

Eric Guizzo, Christian Marinoni, Marco Pennese, Xinlei Ren, Xiguang Zheng, Chen Zhang, Bruno Masiero, Aurelio Uncini, and Danilo Comminiello. 2022. L3das22 challenge: Learning 3d audio sources in a real office environment. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 9186–9190. IEEE.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, et al. 2020. Conformer: Convolution-augmented Transformer for speech recognition. In *Proc. Interspeech*, pages 5036–5040.

Pengcheng Guo, Florian Boyer, Xuankai Chang, Tomoki Hayashi, Yosuke Higuchi, Hirofumi Inaguma, Naoyuki Kamo, Chenda Li, Daniel Garcia-Romero, Jiatong Shi, et al. 2021a. Recent developments on ESPnet toolkit boosted by Conformer. In *Proc. ICASSP*, pages 5874–5878.

Pengcheng Guo, Xuankai Chang, Shinji Watanabe, and Lei Xie. 2021b. Multi-speaker ASR combining non-autoregressive conformer CTC and conditional speaker chain. *arXiv preprint arXiv:2106.08595*.

Reinhold Haeb-Umbach, Jahn Heymann, Lukas Drude, Shinji Watanabe, Marc Delcroix, and Tomohiro Nakatani. 2021. Far-field automatic speech recognition. *Proc. IEEE*, 109(2):124–148.

Reinhold Haeb-Umbach, Shinji Watanabe, Tomohiro Nakatani, Michiel Bacchiani, Bjorn Hoffmeister, Michael L Seltzer, Heiga Zen, and Mehrez Souden. 2019. Speech processing for digital home assistants: Combining signal processing with deep-learning techniques. *IEEE Signal Processing Magazine*, 36(6):111–124.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. 2014a. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Awni Hannun, Vineel Pratap, Jacob Kahn, and Wei-Ning Hsu. 2020. Differentiable weighted finite-state transducers. *arXiv preprint arXiv:2010.01003*.

Awni Y Hannun, Andrew L Maas, Daniel Jurafsky, and Andrew Y Ng. 2014b. First-pass large vocabulary continuous speech recognition using bi-directional recurrent DNNs. *arXiv preprint arXiv:1408.2873*.

Hynek Hermansky, Daniel PW Ellis, and Sangita Sharma. 2000. Tandem connectionist feature extraction for conventional HMM systems. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1635–1638.

John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. 2016a. Deep clustering: Discriminative embeddings for segmentation and separation. In *Proc. ICASSP*, pages 31–35.

John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. 2016b. Deep clustering: Discriminative embeddings for segmentation and separation. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 31–35.

Jahn Heymann, Lukas Drude, Christoph Boeddeker, Patrick Hanebrink, and Reinhold Haeb-Umbach. 2017. Beamnet: End-to-end training of a beamformer-supported multi-channel ASR system. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5325–5329.

Jahn Heymann, Lukas Drude, and Reinhold Haeb-Umbach. 2016. Neural network based spectral mask estimation for acoustic beamforming. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 196–200.

Jahn Heymann, Lukas Drude, Reinhold Haeb-Umbach, Keisuke Kinoshita, and Tomohiro Nakatani. 2019. Joint optimization of neural network-based WPE dereverberation and acoustic model for robust online ASR. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6655–6659.

Yosuke Higuchi, Hirofumi Inaguma, Shinji Watanabe, Tetsuji Ogawa, and Tetsunori Kobayashi. 2021. Improved Mask-CTC for non-autoregressive end-to-end ASR. In *Proc. ICASSP*, pages 8363–8367. IEEE.

Yosuke Higuchi, Shinji Watanabe, Nanxin Chen, Tetsuji Ogawa, and Tetsunori Kobayashi. 2020. Mask CTC: Non-autoregressive end-to-end ASR with CTC and mask predict. In *Proc. Interspeech*, pages 3655–3659. ISCA.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97.

Takaaki Hori, Jaejin Cho, and Shinji Watanabe. 2018. End-to-end speech recognition with word-based RNN language models. In *Proc. IEEE Spoken Language Technology Workshop (SLT)*, pages 389–396.

Takaaki Hori, Chiori Hori, Yasuhiro Minami, and Atsushi Nakamura. 2007. Efficient wfst-based one-pass decoding with on-the-fly hypothesis rescoring in extremely large vocabulary continuous speech recognition. *IEEE Transactions on audio, speech, and language processing*, 15(4):1352–1365.

Takaaki Hori, Shinji Watanabe, and John Hershey. 2017. Joint CTC/attention decoding for end-to-end speech recognition. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 1, pages 518–529.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021a. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

Wei-Ning Hsu, Yao-Hung Hubert Tsai, Benjamin Bolte, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021b. HuBERT: How much can a bad teacher benefit ASR pre-training? In *Proc. ICASSP*, pages 6533–6537.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *Proc. International Conference on Learning Representations*.

Yan Huang, Dong Yu, Yifan Gong, and Chaojun Liu. 2013. Semi-supervised gmm and dnn acoustic model training with multi-system combination and confidence re-calibration. In *Interspeech*, pages 2360–2364.

Zili Huang, Desh Raj, Paola García, and Sanjeev Khudanpur. 2023. Adapting self-supervised models to multi-talker speech recognition using speaker embeddings. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R. Hershey. 2016. Single-channel multi-speaker separation using deep clustering. In *Proc. ISCA Interspeech*.

Kazuma Iwamoto, Tsubasa Ochiai, Marc Delcroix, Rintaro Ikeshita, Hiroshi Sato, Shoko Araki, and Shigeru Katagiri. 2022. How bad are artifacts?: Analyzing the impact of speech enhancement errors on asr. *arXiv preprint arXiv:2201.06685*.

Kazuma Iwamoto, Tsubasa Ochiai, Marc Delcroix, Rintaro Ikeshita, Hiroshi Sato, Shoko Araki, and Shigeru Katagiri. 2023. How does end-to-end speech recognition training impact speech enhancement artifacts? *arXiv preprint arXiv:2311.11599*.

Wenbin Jiang and Kai Yu. 2023. Speech enhancement with integration of neural homomorphic synthesis and spectral masking. *IEEE Transactions on Audio, Speech, and Language Processing*, 31:1758–1770.

Jacob Kahn, Morgane Rivière, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al. 2020. Libri-light: A benchmark for ASR with limited or no supervision. In *Proc. ICASSP*, pages 7669–7673.

Naoyuki Kanda, Yashesh Gaur, Xiaofei Wang, Zhong Meng, Zhuo Chen, Tianyan Zhou, and Takuya Yoshioka. 2020a. Joint speaker counting, speech recognition, and speaker identification for overlapped speech of any number of speakers. In *Proc. Interspeech*, pages 36–40. ISCA.

Naoyuki Kanda, Yashesh Gaur, Xiaofei Wang, Zhong Meng, and Takuya Yoshioka. 2020b. Serialized output training for end-to-end overlapped speech recognition. *arXiv preprint arXiv:2003.12687*.

Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, et al. 2019a. A comparative study on Transformer vs RNN in speech applications. In *Proc. ASRU*, pages 449–456.

Shigeki Karita, Nelson Enrique Yalta Soplin, Shinji Watanabe, Marc Delcroix, Atsunori Ogawa, and Tomohiro Nakatani. 2019b. Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration. In *Proc. ISCA Interspeech*, pages 1408–1412.

Chanwoo Kim, Ehsan Variani, Arun Narayanan, and Michiel Bacchiani. 2017a. Efficient implementation of the room simulator for training deep neural network acoustic models. *arXiv preprint arXiv:1712.03439*.

Suyoun Kim, Takaaki Hori, and Shinji Watanabe. 2017b. Joint CTC-attention based end-to-end speech recognition using multi-task learning. In *Proc. ICASSP*, pages 4835–4839.

Keisuke Kinoshita, Marc Delcroix, Sharon Gannot, Emanuël A P. Habets, Reinhold Haeb-Umbach, Walter Kellermann, Volker Leutnant, Roland Maas, Tomohiro Nakatani, Bhiksha Raj, et al. 2016. A summary of the reverb challenge: state-of-the-art and remaining challenges in reverberant speech processing research. *EURASIP Journal on Advances in Signal Processing*, 2016:1–19.

Keisuke Kinoshita, Marc Delcroix, Takuya Yoshioka, Tomohiro Nakatani, Emanuel Habets, Reinhold Haeb-Umbach, Volker Leutnant, Armin Sehr, Walter Kellermann, Roland Maas, et al. 2013. The REVERB challenge: A common evaluation framework for dereverberation and recognition of reverberant speech. In *Proc WASPAA*, pages 1–4.

Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur. 2017. A study on data augmentation of reverberant speech for robust speech recognition. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5220–5224. IEEE.

Morten Kolbæk, Dong Yu, Zheng-Hua Tan, and Jesper Jensen. 2017. Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 25(10):1901–1913.

Kenichi Kumatani, John McDonough, and Bhiksha Raj. 2012. Microphone array processing for distant speech recognition: From close-talking microphones to far-field sensors. *IEEE Signal Processing Magazine*, 29(6):127–140.

Lori Lamel, Jean-Luc Gauvain, and Gilles Adda. 2002. Lightly supervised and unsupervised acoustic model training. *Computer speech & language*, 16(1):115–129.

Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R Hershey. 2019a. SDR–half-baked or well done? In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 626–630.

Jonathan Le Roux, Scott T. Wisdom, Hakan Erdogan, and John R. Hershey. 2019b. SDR – half-baked or well done? In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Jaesong Lee and Shinji Watanabe. 2021. Intermediate loss regularization for ctc-based speech recognition. In *Proc. ICASSP*, pages 6224–6228. IEEE.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proc. EMNLP*, pages 1173–1182. ACL.

Bo Li, Tara N Sainath, Arun Narayanan, Joe Caroselli, Michiel Bacchiani, Ananya Misra, Izhak Shafran, Haşim Sak, Golan Pundak, Kean Chin, et al. 2017. Acoustic modeling for google home. *Proc. Interspeech*, pages 399–403.

Bo Li, Tara N Sainath, Ron J Weiss, Kevin W Wilson, and Michiel Bacchiani. 2016. Neural network adaptive beamforming for robust multichannel speech recognition. *Proc. Interspeech*, pages 1976–1980.

Chenda Li, Yao Qian, Zhuo Chen, Naoyuki Kanda, Dongmei Wang, Takuya Yoshioka, Yanmin Qian, and Michael Zeng. 2023. Adapting multi-lingual asr models for handling multiple talkers. *arXiv preprint arXiv:2305.18747*.

Jinyu Li et al. 2022. Recent advances in end-to-end automatic speech recognition. *APSIPA Transactions on Signal and Information Processing*, 11(1).

Jindřich Libovickỳ and Jindřich Helcl. 2018. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In *Proc. EMNLP*, pages 3016–3021. ACL.

Philipos C Loizou. 2007. *Speech enhancement: theory and practice*. CRC press.

Yen-Ju Lu, Xuankai Chang, Chenda Li, Wangyou Zhang, Samuele Cornell, Zhaoheng Ni, Yoshiki Masuyama, Brian Yan, Robin Scheibler, Zhong-Qiu Wang, et al. 2022a. Espnet-se++: Speech enhancement for robust speech recognition, translation, and understanding. *arXiv preprint arXiv:2207.09514*.

Yen-Ju Lu, Samuele Cornell, Xuankai Chang, Wangyou Zhang, Chenda Li, Zhaoheng Ni, Zhong-Qiu Wang, and Shinji Watanabe. 2022b. Towards low-distortion multi-channel speech enhancement: The espnet-se submission to the l3das22 challenge. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 9201–9205.

Yi Luo, Zhuo Chen, and Takuya Yoshioka. 2020. Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 46–50.

Yi Luo and Mesgarani. 2018. Tasnet: time-domain audio separation network for real-time, single-channel speech separation. In *Proc. ICASSP*, pages 696–700.

Yi Luo and Nima Mesgarani. 2019a. Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 27(8):1256–1266.

Yi Luo and Nima Mesgarani. 2019b. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM TASLP*, 27(8):1256–1266.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proc. EMNLP*, pages 1412–1421.

Matthew Maciejewski, Gordon Wichern, Emmett McQuinn, and Jonathan Le Roux. 2020. Whamr!: Noisy and reverberant single-channel speech separation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 696–700. IEEE.

Yoshiki Masuyama, Xuankai Chang, Samuele Cornell, Shinji Watanabe, and Nobutaka Ono. 2023a. End-to-end integration of speech recognition, dereverberation, beamforming, and self-supervised learning representation. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 260–265. IEEE.

Yoshiki Masuyama, Xuankai Chang, Wangyou Zhang, Samuele Cornell, Zhong-Qiu Wang, Nobutaka Ono, Yanmin Qian, and Shinji Watanabe. 2023b. Exploring the integration of speech separation and recognition with self-supervised learning representation. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 1–5.

Tobias Menne, Jahn Heymann, Anastasios Alexandridis, Kazuki Irie, Albert Zeyer, Markus Kitza, Pavel Golik, Ilia Kulikov, Lukas Drude, Ralf Schlüter, et al. 2016. The RWTH/UPB/FORTH system combination for the 4th CHiME challenge evaluation. In *Proc. CHiME workshop*.

Tobias Menne, Ilya Sklyar, Ralf Schlüter, and Hermann Ney. 2019. Analysis of deep clustering as preprocessing for automatic speech recognition of sparsely overlapping speech. In *Proc.Interspeech*, pages 2638–2642. ISCA.

Yajie Miao, Mohammad Gowayyed, and Florian Metze. 2015. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 167–174.

Wu Minhua, Kenichi Kumatani, Shiva Sundaram, Nikko Ström, and Björn Hoffmeister. 2019. Frequency domain multi-channel acoustic modeling for distant speech recognition. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6640–6644.

Abdelrahman Mohamed, Hung-yi Lee, Lasse Borgholt, Jakob D Havtorn, Joakim Edin, Christian Igel, Katrin Kirchhoff, Shang-Wen Li, Karen Livescu, Lars Maaløe, et al. 2022a. Self-supervised speech representation learning: A review. *Proc. IEEE Journal of Selected Topics in Signal Processing*, 16(6):1179–1210.

Abdelrahman Mohamed, Hung-yi Lee, Lasse Borgholt, Jakob D Havtorn, Joakim Edin, Christian Igel, Katrin Kirchhoff, Shang-Wen Li, Karen Livescu, Lars Maaløe, et al. 2022b. Self-supervised speech representation learning: A review. *IEEE Journal of Selected Topics in Signal Processing*.

Mehryar Mohri, Fernando Pereira, and Michael Riley. 2002. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88.

Vishal Monga, Yuelong Li, and Yonina C Eldar. 2021. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2):18–44.

Niko Moritz, Takaaki Hori, and Jonathan Le Roux. 2019. Streaming end-to-end speech recognition with joint CTC-attention based models. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 936–943.

Niko Moritz, Takaaki Hori, and Jonathan Le Roux. 2021. Semi-supervised speech recognition via graph-based temporal classification. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6548–6552.

Arun Narayanan and DeLiang Wang. 2014. Joint noise adaptive training for robust automatic speech recognition. In *Proc. ICASSP*, pages 2504–2508.

Tsubasa Ochiai, Shinji Watanabe, Takaaki Hori, and John R Hershey. 2017a. Multichannel end-to-end speech recognition. In *Proc. ICML*, pages 2632–2641.

Tsubasa Ochiai, Shinji Watanabe, and Shigeru Katagiri. 2017b. Does speech enhancement work with end-to-end ASR objectives?: Experimental analysis of multichannel end-to-end ASR. In *Proc. IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *Proc. ICASSP*, pages 5206–5210.

Ashutosh Pandey and DeLiang Wang. 2019. Tcnn: Temporal convolutional neural network for real-time speech enhancement in the time domain. In *Proc. ICASSP*, pages 6875–6879.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. 2019. SpecAugment: A simple data augmentation method for automatic speech recognition. *Proc. Interspeech*, pages 2613–2617.

Ankita Pasad, Ju-Chieh Chou, and Karen Livescu. 2021. Layer-wise analysis of a self-supervised speech representation model. In *Proc. ASRU*.

Yifan Peng, Jinchuan Tian, Brian Yan, Dan Berrebbi, Xuankai Chang, Xinjian Li, Jiatong Shi, Siddhant Arora, William Chen, Roshan Sharma, et al. 2023. Reproducing whisper-style training using an open-source toolkit and publicly available data. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 1–8.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The Kaldi speech recognition toolkit. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*.

Daniel Povey, Hossein Hadian, Pegah Ghahremani, Ke Li, and Sanjeev Khudanpur. 2018. A time-restricted self-attention layer for ASR. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5874–5878.

Rohit Prabhavalkar, Takaaki Hori, Tara N Sainath, Ralf Schlüter, and Shinji Watanabe. 2023. End-to-end speech recognition: A survey. *IEEE/ACM Transactions on Audio, Speech and Language Processing*.

Yanmin Qian, Mengxiao Bi, Tian Tan, and Kai Yu. 2016. Very deep convolutional neural networks for noise robust speech recognition. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 24(12):2263–2276.

Yanmin Qian, Xuankai Chang, and Dong Yu. 2018a. Single-channel multi-talker speech recognition with permutation invariant training. *Speech Communication*, 104:1–11.

Yanmin Qian, Chao Weng, Xuankai Chang, Shuai Wang, and Dong Yu. 2018b. Past review, current progress, and challenges ahead on the cocktail party problem. *Frontiers of Information Technology & Electronic Engineering*, 19(1):40–63.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proc. International Conference on Machine Learning (ICML)*, pages 28492–28518. PMLR.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI preprint*.

Desh Raj, Pavel Denisov, Zhuo Chen, Hakan Erdogan, Zili Huang, Maokui He, Shinji Watanabe, Jun Du, Takuya Yoshioka, Yi Luo, et al. 2021. Integration of speech separation, diarization, and recognition for multi-speaker meetings: System description, comparison, and analysis. In *2021 IEEE spoken language technology workshop (SLT)*, pages 897–904. IEEE.

Neville Ryant, Prachi Singh, Venkat Krishnamohan, Rajat Varma, Kenneth Church, Christopher Cieri, Jun Du, Sriram Ganapathy, and Mark Liberman. 2021. The Third DIHARD Diarization Challenge. In *Proc. Interspeech*, pages 3570–3574.

Chitwan Saharia, William Chan, Saurabh Saxena, and Mohammad Norouzi. 2020. Non-autoregressive machine translation with latent alignments. In *Proc. EMNLP*, pages 1098–1108. ACL.

Tara N Sainath, Abdel-rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran. 2013. Deep convolutional neural networks for LVCSR. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8614–8618.

Hiroshi Seki, Takaaki Hori, Shinji Watanabe, Jonathan Le Roux, and John R Hershey. 2018. A purely end-to-end system for multi-speaker speech recognition. In *Proc. ACL*, pages 2620–2630.

Michael L Seltzer, Bhiksha Raj, and Richard M Stern. 2004. Likelihood-maximizing beamforming for robust hands-free speech recognition. *IEEE Trans. Speech, Audio process.*, 12(5):489–498.

Shane Settle, Jonathan Le Roux, Takaaki Hori, Shinji Watanabe, and John R Hershey. 2018. End-to-end multi-speaker speech recognition. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4819–4823.

Aswin Shanmugam Subramanian, Xiaofei Wang, Shinji Watanabe, Toru Taniguchi, Dung Tran, and Yuya Fujita. 2019. An investigation of end-to-end multichannel speech recognition for reverberant and mismatch conditions. *arXiv preprint arXiv:1904.09049*.

Jing Shi, Xuankai Chang, Pengcheng Guo, Shinji Watanabe, Yusuke Fujita, Jiaming Xu, Bo Xu, and Lei Xie. 2020a. Sequence to multi-sequence learning via conditional chain mapping for mixture signals. In *Proc. NeurIPS*, pages 3735–3747.

Jing Shi, Jiaming Xu, Yusuke Fujita, Shinji Watanabe, and Bo Xu. 2020b. Speaker-conditional chain model for speech separation and extraction. In *Proc. Interspeech*, pages 2707–2711. ISCA.

Yusuke Shinohara. 2016a. Adversarial multi-task learning of deep neural networks for robust speech recognition. In *Proc. ISCA Interspeech*, pages 2369–2372.

Yusuke Shinohara. 2016b. Adversarial multi-task learning of deep neural networks for robust speech recognition. In *Proc. Interspeech*, pages 2369–2372.

Ilya Sklyar, Anna Piunova, and Yulan Liu. 2021. Streaming multi-speaker ASR with RNN-T. In *Proc. ICASSP*, pages 6903–6907.

Mehrez Souden, Jacob Benesty, and Sofiène Affes. 2009. On optimal frequency-domain multichannel linear filtering for noise reduction. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(2):260–276.

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. 2019. Insertion Transformer: Flexible sequence generation via insertion operations. In *Proc. ICML*, pages 5976–5985. PMLR.

Cem Subakan, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong. 2021. Attention is all you need in speech separation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 21–25. IEEE.

Aswin Shanmugam Subramanian, Xiaofei Wang, Murali Karthick Baskar, Shinji Watanabe, Toru Taniguchi, Dung Tran, and Yuya Fujita. 2019. Speech enhancement using end-to-end speech recognition objectives. In *Proc. WASPAA*, pages 234–238.

Ying Sun, Prabhu Babu, and Daniel P Palomar. 2016. Majorization-minimization algorithms in signal processing, communications, and machine learning. *IEEE Transactions on Signal Processing*, 65(3):794–816.

Ke Tan, Zhong-Qiu Wang, and DeLiang Wang. 2022. Neural spectrospatial filtering. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:605–621.

Tian Tan, Yanmin Qian, and Dong Yu. 2018. Knowledge transfer in permutation invariant training for single-channel multi-talker speech recognition. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Zhengkun Tian, Jiangyan Yi, Jianhua Tao, Ye Bai, Shuai Zhang, and Zhengqi Wen. 2020. Spike-triggered non-autoregressive Transformer for end-to-end speech recognition. In *Proc. Interspeech*, pages 5026–5020. ISCA.

Hsiang-Sheng Tsai, Heng-Jui Chang, Wen-Chin Huang, Zili Huang, Kushal Lakhotia, Shu-wen Yang, Shuyan Dong, Andy Liu, Cheng-I Lai, Jiatong Shi, et al. 2022. Superb-sg: Enhanced speech processing universal performance benchmark for semantic and generative capabilities. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8479–8492.

Nicolas Turpault, Scott Wisdom, Hakan Erdogan, John R. Hershey, Romain Serizel, Eduardo Fonseca, Prem Seetharaman, and Justin Salamon. 2020. Improving sound event detection in domestic environments using sound separation. In *DCASE*.

Barry D Van Veen and Kevin M Buckley. 1988. Beamforming: A versatile approach to spatial filtering. *IEEE ASSP Magazine*, 5(2):4–24.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. 2006. Performance measurement in blind audio source separation. *IEEE Trans. Audio, Speech, Lang. Process.*, 14(4):1462–1469.

Emmanuel Vincent, Shinji Watanabe, Aditya Arie Nugraha, Jon Barker, and Ricard Marxer. 2017a. An analysis of environment, microphone and data simulation mismatches in robust speech recognition. *Computer Speech & Language*, 46:535–557.

Emmanuel Vincent, Shinji Watanabe, Aditya Arie Nugraha, Jon Barker, and Ricard Marxer. 2017b. An analysis of environment, microphone and data simulation mismatches in robust speech recognition. *Computer Speech & Language*, 46:535–557.

Thilo von Neumann, Christoph Boeddeker, Lukas Drude, Keisuke Kinoshita, Marc Delcroix, Tomohiro Nakatani, and Reinhold Haeb-Umbach. 2020a. Multi-talker ASR for an unknown number of sources: Joint training of source counting, separation and ASR. *Proc. Interspeech*, pages 3097–3101.

Thilo von Neumann, Keisuke Kinoshita, Marc Delcroix, Shoko Araki, Tomohiro Nakatani, and Reinhold Haeb-Umbach. 2019. All-neural online source separation, counting, and diarization for meeting analysis. In *Proc. ICASSP*, pages 91–95. IEEE.

Thilo von Neumann, Keisuke Kinoshita, Lukas Drude, Christoph Boeddeker, Marc Delcroix, Tomohiro Nakatani, and Reinhold Haeb-Umbach. 2020b. End-to-end training of time domain audio separation and recognition. In *Proc. ICASSP*, pages 7004–7008. IEEE.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. 2021a. Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In *Proc. ACL*, pages 993–1003.

DeLiang Wang and Guy J. Brown. 2006. *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Wiley-IEEE Press.

DeLiang Wang and Jitong Chen. 2018a. Supervised speech separation based on deep learning: An overview. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 26(10):1702–1726.

DeLiang Wang and Jitong Chen. 2018b. Supervised speech separation based on deep learning: An overview. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(10):1702–1726.

Xiaofei Wang, Ruizhi Li, Sri Harish Mallidi, Takaaki Hori, Shinji Watanabe, and Hynek Hermansky. 2019. Stream attention-based multi-array end-to-end speech recognition. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7105–7109.

Yiming Wang, Jinyu Li, Heming Wang, Yao Qian, Chengyi Wang, and Yu Wu. 2021b. Wav2vec-switch: Contrastive learning from original-noisy speech pairs for robust speech recognition. *arXiv preprint arXiv:2110.04934*.

Zhong-Qiu Wang, Samuele Cornell, Shukjae Choi, Younglo Lee, Byeong-Yeol Kim, and Shinji Watanabe. 2022. Tf-gridnet: Integrating full-and sub-band modeling for speech separation. *arXiv preprint arXiv:2211.12433*.

Zhong-Qiu Wang, Jonathan Le Roux, and John R Hershey. 2018. Multi-channel deep clustering: Discriminative spectral and spatial embeddings for speaker-independent speech separation. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Zhong-Qiu Wang, Peidong Wang, and DeLiang Wang. 2020. Complex spectral mapping for single- and multi-channel speech enhancement and robust asr. *IEEE/ACM TASLP*, 28:1778–1787.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al. 2018. ESPnet: End-to-end speech processing toolkit. In *Proc. Interspeech*, pages 2207–2211.

Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi. 2017. Hybrid CTC/attention architecture for end-to-end speech recognition. *IEEE J. Sel. Topics Signal Process.*, 11(8):1240–1253.

Shinji Watanabe, Michael Mandel, Jon Barker, Emmanuel Vincent, Ashish Arora, Xuankai Chang, Sanjeev Khudanpur, Vimal Manohar, Daniel Povey, Desh Raj, David Snyder, Aswin Shanmugam Subramanian, Jan Trmal, Bar Ben Yair, Christoph Boeddeker, Zhaoheng Ni, Yusuke Fujita, Shota Horiguchi, Naoyuki Kanda, Takuya Yoshioka, and Neville Ryant. 2020. CHiME-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings. In *Proc. 6th CHiME*, pages 1–7.

Gordon Wichern, Joe Antognini, Michael Flynn, Licheng Richard Zhu, Emmett McQuinn, Dwight Crow, Ethan Manilow, and Jonathan Le Roux. 2019. Wham!: Extending speech separation to noisy environments. In *Proc. Interspeech*, pages 1368–1372. ISCA.

Donald S Williamson, Yuxuan Wang, and DeLiang Wang. 2015. Complex ratio masking for monaural speech separation. *IEEE/ACM transactions on audio, speech, and language processing*, 24(3):483–492.

Bo Wu, Kehuang Li, Fengpei Ge, Zhen Huang, Minglei Yang, Sabato Marco Siniscalchi, and Chin-Hui Lee. 2017. An end-to-end deep learning approach to simultaneous speech dereverberation and acoustic modeling for robust speech recognition. *Proc. IEEE Journal of Selected Topics in Signal Processing*, 11(8):1289–1300.

Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig. 2017. The Microsoft 2016 conversational speech recognition system. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5255–5259.

Yong Xu, Chao Weng, Like Hui, Jianming Liu, Meng Yu, Dan Su, and Dong Yu. 2019. Joint training of complex ratio mask based beamformer and acoustic model for noise robust asr. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6745–6749. IEEE.

Lei Yang, Wei Liu, and Weiqin Wang. 2022a. Tfpsnet: Time-frequency domain path scanning network for speech separation. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6842–6846. IEEE.

Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y Lin, Andy T Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, et al. 2021. SUPERB: Speech processing Universal PERformance Benchmark. In *Proc. Interspeech*, pages 1194–1198.

Yufeng Yang, Peidong Wang, and DeLiang Wang. 2022b. A conformer based acoustic model for robust automatic speech recognition. *arXiv preprint arXiv:2203.00725*.

Takuya Yoshioka, Hakan Erdogan, Zhuo Chen, and Fil Alleva. 2018a. Multi-microphone neural speech separation for far-field multi-talker speech recognition. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5739–5743.

Takuya Yoshioka, Hakan Erdogan, Zhuo Chen, and Fil Alleva. 2018b. Multi-microphone neural speech separation for far-field multi-talker speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5739–5743. IEEE.

Takuya Yoshioka, Hakan Erdogan, Zhuo Chen, Xiong Xiao, and Fil Alleva. 2018c. Recognizing overlapped speech in meetings: A multichannel separation approach using neural networks. In *Proc. ISCA Interspeech*.

Takuya Yoshioka, Nobutaka Ito, Marc Delcroix, Atsunori Ogawa, Keisuke Kinoshita, Masakiyo Fujimoto, Chengzhu Yu, Wojciech J Fabian, Miquel Espi, Takuya Higuchi, Shoko Araki, and Tomohiro Nakatani. 2015a. The NTT CHiME-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 436–443.

Takuya Yoshioka, Nobutaka Ito, Marc Delcroix, Atsunori Ogawa, Keisuke Kinoshita, Masakiyo Fujimoto, Chengzhu Yu, Wojciech J Fabian, Miquel Espi, Takuya Higuchi, et al. 2015b. The NTT CHiME-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 436–443.

Takuya Yoshioka and Tomohiro Nakatani. 2012. Generalization of multi-channel linear prediction methods for blind mimo impulse response shortening. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 20(10):2707–2720.

Takuya Yoshioka, Armin Sehr, Marc Delcroix, Keisuke Kinoshita, Roland Maas, Tomohiro Nakatani, and Walter Kellermann. 2012. Making machines understand us in reverberant rooms: Robustness against reverberation for automatic speech recognition. *IEEE Signal Processing Magazine*, 29(6):114–126.

Steve Young. 1996. A review of large-vocabulary continuous-speech. *IEEE signal processing magazine*, 13(5):45.

Dong Yu, Xuankai Chang, and Yanmin Qian. 2017a. Recognizing multi-talker speech with permutation invariant training. In *Proc. ISCA Interspeech*, pages 2456–2460.

Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen. 2017b. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In *Proc. ICASSP*, pages 241–245.

Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen. 2017c. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 241–245.

Albert Zeyer, Kazuki Irie, Ralf Schlüter, and Hermann Ney. 2018. Improved training of end-to-end attention models for speech recognition. *arXiv preprint arXiv:1805.03294*.

Jisi Zhang, Cătălin Zorilă, Rama Doddipatla, and Jon Barker. 2021a. Time-domain speech extraction with spatial information and multi speaker conditioning mechanism. In *ICASSP 2021-2021*

IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6084–6088. IEEE.

Jisi Zhang, Cătălin Zorilă, Rama Doddipatla, and Jon Barker. 2021b. Time-domain speech extraction with spatial information and multi speaker conditioning mechanism. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6084–6088. IEEE.

Wangyou Zhang, Xuankai Chang, Christoph Boeddeker, Tomohiro Nakatani, Shinji Watanabe, and Yanmin Qian. 2022a. End-to-end dereverberation, beamforming, and speech recognition in a cocktail party. *IEEE Transactions on Audio, Speech, and Language Processing*, 30:3173–3188.

Wangyou Zhang, Xuankai Chang, Christoph Boeddeker, Tomohiro Nakatani, Shinji Watanabe, and Yanmin Qian. 2022b. End-to-end dereverberation, beamforming, and speech recognition in a cocktail party. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:3173–3188.

Wangyou Zhang, Jing Shi, Chenda Li, Shinji Watanabe, and Yanmin Qian. 2021c. Closing the gap between time-domain multi-channel speech enhancement on real and simulation conditions. In *Proc. WASPAA*, pages 146–150.

Wangyou Zhang, Aswin Shanmugam Subramanian, Xuankai Chang, Shinji Watanabe, and Yanmin Qian. 2020a. End-to-end far-field speech recognition with unified dereverberation and beamforming. *Proc. Interspeech 2020*, pages 324–328.

Yu Zhang, Wei Han, James Qin, Yongqiang Wang, Ankur Bapna, Zhehuai Chen, Nanxin Chen, Bo Li, Vera Axelrod, Gary Wang, et al. 2023. Google usm: Scaling automatic speech recognition beyond 100 languages. *arXiv preprint arXiv:2303.01037*.

Yu Zhang, James Qin, Daniel S Park, Wei Han, Chung-Cheng Chiu, Ruoming Pang, Quoc V Le, and Yonghui Wu. 2020b. Pushing the limits of semi-supervised learning for automatic speech recognition. *arXiv preprint arXiv:2010.10504*.