Joint speech separation, diarization, and recognition for automatic meeting transcription

Can Cui

UNIVERSITÉ DE LORRAINE

Inria

Loria
Laboratoire lorrain de recherche en informatique et ses applications

Thèse de Doctorat

Can Cui

Mémoire présenté en vue de l'obtention du
grade de Docteur de l'Université de Lorraine
Mention Informatique

École doctorale:
Informatique, Automatique, Electronique-Electrotechnique, Mathématiques et Sciences de L'architecture

Unité de recherche:
Laboratoire Lorraine de Recherche en Informatique et ses Applications
UMR 7503

Soutenue le 1er octobre 2024

Séparation, diarisation et reconnaissance de la parole conjointes pour la transcription automatique de réunions

Composition Du Jury

Rapporteur :	**Reinhold Häb-Umbach**, Professeur, Université de Paderborn, Allemagne
Rapporteur :	**Yannick Estève**, Professeur, Avignon Université, France
Présidente du jury :	**Marie Tahon**, Professeure, Université du Mans, France
Directeur de thèse :	**Emmanuel Vincent**, Directeur de recherche, Centre Inria de l'U. de Lorraine, France
Co-directeur de thèse :	**Mostafa Sadeghi**, Inria Starting Faculty Position, Centre Inria de l'U. de Lorraine, France
Co-directeur de thèse :	**Imran Sheikh**, Ingénieur de Recherche, Vivoka, France

Octobre, 2024
Nancy, France

UNIVERSITÉ DE LORRAINE

Inria

Loria
Laboratoire lorrain de recherche en informatique et ses applications

Doctoral Thesis

Can Cui

Dissertation presented in order to obtain the
Doctoral Degree from the University of Lorraine
Computer Science

Doctoral School:
Computer Science, Automation, Electronics, Mathematics and Architectural Sciences (IAEM)

Research Unit:
Lorraine Laboratory for Research in Computer Science and its Applications
UMR 7503

Defended on October 1, 2024

Joint speech separation, diarization and recognition for automatic meeting transcription

Jury Members

Rapporteur:	**Reinhold Häb-Umbach**, Professor, University of Paderborn, Germany
Rapporteur:	**Yannick Estève**, Professor, Avignon Université, France
Examiner:	**Marie Tahon**, Professor, Université du Mans, France
PhD supervisor:	**Emmanuel Vincent**, Senior Research Scientist, Centre Inria de l'U. de Lorraine, France
PhD co-supervisor:	**Mostafa Sadeghi**, Inria Starting Faculty Position, Centre Inria de l'U. de Lorraine, France
PhD co-supervisor:	**Imran Sheikh**, Research engineer, Vivoka, France

October, 2024
Nancy, France

Acknowledgements

Pursuing a PhD is not an easy path, much like the traditional Chinese story of the Journey to the West, where one must overcome 81 hardships to obtain the true scripture. I am grateful to everyone who has been part of this journey, but in the following paragraphs, I will focus on expressing my gratitude to a few special individuals.

I would like to first thank my mother, Yuehong Zhou, and my father, Baoliang Cui. As an only child, most Chinese parents in similar families would want their child to stay close to them, yet I have been living and studying in France for seven years, five of which I was unable to return to China due to the pandemic. However, my parents have always supported my decisions, which is something truly remarkable.

I would also like to sincerely thank my supervisors, Emmanuel Vincent, Mostafa Sadeghi, and Imran Sheikh. I am deeply grateful to Emmanuel for choosing me to work on this project from the very beginning, even though I might not have been the most qualified in terms of educational background. His choice gave me the determination to overcome difficulties again and again. I've often told myself, "Someone as brilliant as him wouldn't make a wrong judgment about people." Beyond the moral support, Emmanuel has a solid foundation of knowledge and extensive research experience in the field of speech processing. He always provides precise answers to my technical questions, and I learn something new every time we meet. I am also very thankful to Mostafa for his help and support. As my co-director, he is both professional and patient, always there when I need assistance, offering valuable advice. I would also like to take this opportunity to thank my co-encadrant from Vivoka, Imran, who provided timely, professional, and efficient solutions when I encountered issues with running and implementing code. I truly admire his coding skills.

I would also like to express my gratitude to my defense examiner, Marie Tahon, as well as the rapporteurs, Yannick Estève and Reinhold Häb-Umbach. I am very thankful to Marie for chairing the defense and ensuring that all procedures were followed smoothly. I also appreciate the questions she raised at the end, which allowed the audience to gain a more detailed understanding of my dissertation. I am deeply grateful to Yannick and Reinhold for their thorough reading of my dissertation before the defense and for writing very detailed reports. Thank you all three; without your support, my defense could not have

gone so smoothly.

Three years ago, when I first stepped into the Loria lab, it was also autumn. Now, as the story comes to an end, the ginkgo leaves at the entrance have turned yellow once again. Every year, stories begin and end here, and I am fortunate to have been part of one of them. I would like to express my gratitude to my colleagues at the Multispeech team, many of whom have become great friends, for their immense support and companionship over the past three years. I want to thank Paul Magron, Sewade Ogun, Marina Krémé, Sandipana Dowerah, Tulika Bose, Prerak Srivastava, Louis Delebecque, Taous Iatariene, Louis Abel, Michel Olvera, Marie-Anne Lacroix, Vinicius Ribeiro, Nicolas Zampieri, Nasser Monir, Sofiane Azzouz, Hélène Zganic, Nicolas Furnon, Joris Cosentino, Théo Biasutto-Lervat, Mickaella Verdon, Shakeel Sheikh, Pierre Champion, Seyed Hosseini, Romain Serizel, Hugo Bergerat, Ashwin Geet D'Sa, Ali Golmakani, Soklay Heng, Soklong Him, Louis Lalay, Louis Bahrman, Colleen Beaumard, Berne Nortier, Antoine Bruez, Jean-Eudes Ayilo, Robin San Roman, Stéphane Dilungana, Vincent Colotte, Georgios Zervakis, Ajinkya Kulkarni, Raphaël Bagat, Constance Douwes, Stéphane Rossignol, Natalia Tomashenko, Sam Bigeard, Slim Ouni, Emmanuelle Deschamps, and Denis Jouvet.

During my time at Vivoka, I was fortunate to meet a group of wonderful colleagues who, in addition to engaging in discussions on professional and academic topics, brought endless joy to my life. I would like to especially thank the R&D team members: Nora Lindvall, Cassio Batista, Firas Hmida, Denys Vivdenko, and Vincent Leroy. Throughout my PhD journey, I also met many friends who supported me both in my work and personal life. A heartfelt thank you to Tingting Wang, Maiyu Wang, Nishan Tang, Haolian Shi, Jarvis Looi, Lucas Oliverio, Aurore Lenoir, Martin Boillat, Adirana Vicentijevic, Yuxuan Kou and Bei Zhou. I would also like to extend a special thanks to my two adopted cats, Hertz and Decibel. Through my time with them, I've learned patience, and they have brought both joy and frustration into my life, amplifying my emotional spectrum and helping me embrace the range of feelings that life brings. They've shown me the meaning of having a sense of responsibility for other living beings.

Last but not least, I want to acknowledge myself. I am grateful to the version of me from three years ago who fearlessly chose the path of pursuing a PhD. Over these three years, I have cried, laughed, been anxious, and found relief—experiencing every emotion to its fullest. I have also come to understand that the road of scientific research is long and challenging, with no clear endpoint. I hope that this experience will inspire my future self with the courage to not give up when facing difficulties, and the determination to keep moving forward.

Figure 3: Multispeech team.

Résumé

La transcription de réunions enregistrées par une antenne de microphones distante est particulièrement difficile en raison de la superposition des locuteurs, du bruit ambiant et de la réverbération. Pour résoudre ces problèmes, nous avons exploré trois approches. Premièrement, nous utilisons un modèle de séparation de sources multicanal pour séparer les locuteurs, puis un modèle de reconnaissance automatique de la parole (ASR) monocanal et mono-locuteur pour transcrire la parole séparée et rehaussée. Deuxièmement, nous proposons un modèle multicanal multi-locuteur de bout-en-bout (MC-SA-ASR), qui s'appuie sur un modèle multi-locuteur monocanal (SA-ASR) existant et inclut un encodeur multicanal par Conformer avec un mécanisme d'attention multi-trame intercanale (MFCCA). Contrairement aux approches traditionnelles qui nécessitent un modèle de rehaussement de la parole multicanal en amont, le modèle MC-SA-ASR traite les microphones distants de bout-en-bout. Nous avons également expérimenté différentes caractéristiques d'entrée, dont le banc de filtres Mel et les caractéristiques de phase, pour ce modèle. Enfin, nous utilisons un modèle de formation de voies et de rehaussement multicanal comme pré-traitement, suivi d'un modèle SA-ASR monocanal pour traiter la parole multi-locuteur rehaussée. Nous avons testé différentes techniques de formation de voies fixe, hybride ou neuronale et proposé d'apprendre conjointement les modèles de formation de voies neuronale et de SA-ASR en utilisant le coût d'apprentissage de ce dernier. En plus de ces méthodes, nous avons développé un pipeline de transcription de réunions qui intègre la détection de l'activité vocale, la diarisation et le SA-ASR pour traiter efficacement les enregistrements de réunions réelles.

Les résultats expérimentaux indiquent que, même si l'utilisation d'un modèle de séparation de sources peut améliorer la qualité de la parole, les erreurs de séparation peuvent se propager à l'ASR, entraînant des performances sous-optimales. Une approche guidée de séparation de sources s'avère plus efficace. Notre modèle MC-SA-ASR proposé démontre l'efficacité de l'intégration des informations multicanales et des informations partagées entre les modules d'ASR et de locuteur. Des expériences avec différentes catactéristiques d'entrée révèlent que les modèles appris avec les caractéristiques de Mel Filterbank fonctionnent mieux en termes de taux d'erreur sur les mots (WER) et de taux d'erreur sur les

locuteurs (SER) lorsque le nombre de canaux et de locuteurs est faible (2 canaux avec 1 ou 2 locuteurs). Cependant, pour les configurations à 3 ou 4 canaux et 3 locuteurs, les modèles appris sur des caractéristiques de phase supplémentaires surpassent ceux utilisant uniquement les caractéristiques Mel. Cela suggère que les informations de phase peuvent améliorer la transcription du contenu vocal en exploitant les informations de localisation provenant de plusieurs canaux.

Bien que MC-SA-ASR basé sur MFCCA surpasse les modèles SA-ASR et MC-ASR monocanal sans module de locuteur, les modèle de formation de voies et de SA-ASR conjointes permet d'obtenir des résultats encore meilleurs. Plus précisément, l'apprentissage conjoint de la formation de voies neuronale et de SA-ASR donne les meilleures performances, ce qui indique que l'amélioration de la qualité de la parole pourrait être une approche plus directe et plus efficace que l'utilisation d'un modèle MC-SA-ASR de bout-en-bout pour la transcription de réunions multicanales. En outre, l'étude du pipeline de transcription de réunions réelles souligne le potentiel pour des meilleurs modèles de bout-en-bout. Dans notre étude sur l'amélioration de l'attribution des locuteurs par SA-ASR, nous avons constaté que le module d'ASR n'est pas sensible aux modifications du module de locuteur. Cela met en évidence la nécessité d'architectures améliorées qui intègrent plus efficacement l'ASR et l'information de locuteur.

Abstract

Far-field microphone-array meeting transcription is particularly challenging due to overlapping speech, ambient noise, and reverberation. To address these issues, we explored three approaches. First, we employ a multichannel speaker separation model to isolate individual speakers, followed by a single-channel, single-speaker automatic speech recognition (ASR) model to transcribe the separated and enhanced audio. This method effectively enhances speech quality for ASR. Second, we propose an end-to-end multichannel speaker-attributed ASR (MC-SA-ASR) model, which builds on an existing single-channel SA-ASR model and incorporates a multichannel Conformer-based encoder with multi-frame cross-channel attention (MFCCA). Unlike traditional approaches that require a multichannel front-end speech enhancement model, the MC-SA-ASR model handles far-field microphones in an end-to-end manner. We also experimented with different input features, including Mel filterbank and phase features, for that model. Lastly, we incorporate a multichannel beamforming and enhancement model as a front-end processing step, followed by a single-channel SA-ASR model to process the enhanced multi-speaker speech signals. We tested different fixed, hybrid, and fully neural network-based beamformers and proposed to jointly optimize the neural beamformer and SA-ASR models using the training objective for the latter. In addition to these methods, we developed a meeting transcription pipeline that integrates voice activity detection, speaker diarization, and SA-ASR to process real meeting recordings effectively.

Experimental results indicate that, while using a speaker separation model can enhance speech quality, separation errors can propagate to ASR, resulting in suboptimal performance. A guided speaker separation approach proves to be more effective. Our proposed MC-SA-ASR model demonstrates efficiency in integrating multichannel information and the shared information between the ASR and speaker blocks. Experiments with different input features reveal that models trained with Mel filterbank features perform better in terms of word error rate (WER) and speaker error rate (SER) when the number of channels and speakers is low (2 channels with 1 or 2 speakers). However, for settings with 3 or 4 channels and 3 speakers, models trained with additional phase information outperform those using only Mel filterbank features. This suggests that phase information can enhance

ASR by leveraging localization information from multiple channels.

Although MFCCA-based MC-SA-ASR outperforms the single-channel SA-ASR and MC-ASR models without a speaker block, the joint beamforming and SA-ASR model further improves the performance. Specifically, joint training of the neural beamformer and SA-ASR yields the best performance, indicating that improving speech quality might be a more direct and efficient approach than using an end-to-end MC-SA-ASR model for multichannel meeting transcription. Furthermore, the study of the real meeting transcription pipeline underscores the potential for better end-to-end models. In our investigation on improving speaker assignment in SA-ASR, we found that the speaker block does not effectively help improve the ASR performance. This highlights the need for improved architectures that more effectively integrate ASR and speaker information.

Contents

List of figures

List of tables

List of acronyms

ASR Automatic speech recognition
CHiME Computational hearing in multisource environments
CNN Convolutional neural networks
CTC Connectionist temporal classification
CRDNN Convolutional, recurrent, and dense neural network
Conv-TasNet Fully-convolutional time-domain audio separation network
DAS Delay-and-sum
DP-RNN Dual-path RNN
DNN Deep neural network
DOA Direction-of-arrival
ECAPA-TDNN Emphasized channel attention, propagation and aggregation TDNN
EEND End-to-end neural diarization
FaSNet Filter-and-sum Network
FC Fully connected
GMM Gaussian mixture model
GSS Guided source separation
HMM Hidden Markov model
IHM Individual headset microphone
LSTM Long short-term memory
MFCC Mel-frequency cepstral coefficient
MFCCA Multi-frame cross-channel attention
MDM Multiple distant microphone
MVDR Minimum variance distortionless response
PIT Permutation invariant training
RIR Room impulse response
RNN Recurrent neural network
SA-ASR Speaker-attributed ASR
SD Speaker diarization
SDM Single distant microphone

SER Speaker error rate
SI-SDR Scale-invariant signal-to-distortion ratio
SI-SDRi Scale-invariant signal-to-distortion ratio improvement
SOT Serialized output training
STFT Short-time Fourier transform
TDNN Time delay neural network
VAD Voice activity detection
WER Word error rate
WPE Weighted prediction error

1 Introduction

山重水复疑无路，
柳暗花明又一村。
Mountains loom, waters double, doubt the path; willows dim, flowers bright, another village.

Chinese proverb

1.1 Motivation

In today's fast-paced world, taking manual notes during meetings can be time-consuming and laborious. Coordinating schedules, gathering input, and ensuring accurate documentation often require significant effort and attention, detracting from the focus and productivity of the meeting itself. Imagine a solution where the meeting's audio is automatically transcribed into comprehensive notes, eliminating the need for manual note-taking. Meeting transcription tools have become popular in online meetings leveraging Automatic Speech Recognition (ASR) technology to convert spoken dialogue into written text.

On the academic front, researchers have been working to enhance ASR performance across the more challenging acoustic conditions encountered in face-to-face meetings, including noise, overlapping speakers, and reverberation, with the goal of making ASR models more robust. Research tailored to meeting scenarios aims to address these specific challenges. Speech processing conferences frequently feature challenges on this topic, such as the Multi-Channel Multi-Party Meeting Transcription Challenge (M2MeT) (Yu et al., 2022b) at ICASSP 2022 and the series of CHiME challenges (Barker et al., 2013; Vincent et al., 2013; Barker et al., 2017; Vincent et al., 2016; Barker et al., 2018; Watanabe et al., 2020; Cornell et al., 2023). The combined interest from industry and academia is driving significant advances in meeting transcription research (Boeddeker et al., 2018; Chang et al., 2019; Sklyar et al., 2021; Kanda et al., 2021b).

1.2 Objective and key challenges

Our objective in this thesis is to generate meeting transcripts that accurately indicate who said what and when, based on the speech audio recorded by a microphone array (see Figure 1.1). The main challenges include recognizing speech amidst ambient noise and reverberation, dealing with overlapping speech, and identifying speakers in such conditions.

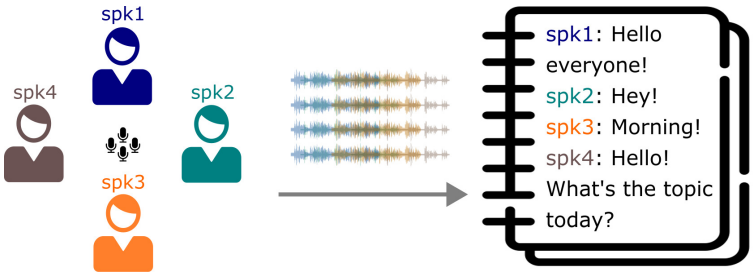


Figure 1.1: Meeting transcription from multichannel audio recorded by a microphone array.

Distant-microphone meeting transcription is a challenging task. To improve performance, many studies have employed a front-end multichannel speech separation module or a series of (fixed, statistical, or neural) beamformers steered towards the speakers to extract individual speech signals from the overlapping speech mixture and subsequently feed each of them to a single-speaker ASR module (Yoshioka et al., 2018b; Chen et al., 2019; Kanda et al., 2019a). The separation error then propagates to the ASR module. Later studies (Chang et al., 2019; Li et al., 2021b; Wu et al., 2021; Zhang et al., 2021; Shi et al., 2022) have proposed to back-propagate the ASR training losses for all speakers to the front-end separation module using a permutation invariant training (PIT) criterion to optimize the two modules jointly. However, the system can often handle only a fixed number of speakers.

To address this, end-to-end multi-speaker ASR and diarization systems for single-channel (Guo et al., 2021; Lu et al., 2021; Sklyar et al., 2021; Kanda et al., 2021b) and multi-channel recordings (Chang et al., 2019, 2020; Scheibler et al., 2023; Shi et al., 2023b) have recently emerged, demonstrating promising results on meeting transcription tasks. Specially, Kanda et al. (2021b) proposed an end-to-end single-channel speaker-count invariant Transformer-based speaker-attributed ASR (SA-ASR) system based on serialized output training (SOT) (Kanda et al., 2020b). This system shares both speech and speaker representations across the multi-speaker ASR and speaker diarization tasks. Similar to their single-channel counterparts, the ASR and speaker identification modules in end-to-end multi-channel ASR and diarization approaches (Chang et al., 2019, 2020; Scheibler et al., 2023) do not share speech and speaker representations. Interestingly, the approach of multichan-

nel word-level diarization with SOT (MC-WD-SOT) of Shi et al. (2023b) performs a fusion of ASR and speaker information. It uses multi-frame cross-channel attention (MFCCA) (Yu et al., 2023) in the ASR encoder to integrate information from different channels and hidden-layer embeddings from the ASR decoder to assist speaker identification.

In addition, past studies on end-to-end meeting transcription have focused on model architecture and have mostly been evaluated on simulated meeting data. Training and inference on real long-length multi-speaker audio recordings is a non-trivial issue due to computational and memory requirements. Existing methods typically rely on ground-truth speaker activity information to segment the long recording at silence positions (Kanda et al., 2021c). In real-life applications, this information is not available and a Voice Activity Detection (VAD) system becomes essential for obtaining speech segments. A second problem with training and inference on real multi-speaker audio is how to obtain the reference speaker embeddings used as inputs by the SA-ASR system. In real-life applications, front-end Speaker Diarization (SD) becomes essential for this purpose. While studies on VAD (Medennikov et al., 2020; Yang et al., 2010) and SD (Park et al., 2022; Xiao et al., 2021) exist independently, there is little discussion on how to optimally integrate them into an SA-ASR pipeline to create a ready-to-use meeting transcription system.

1.3 Contributions

1.3.1 Joint speaker separation, ASR and speaker identification

We start by designing a transcription system that integrates multichannel speech separation with single-channel, single-speaker ASR and diarization. To train the separation model, i.e., FaSNet (Luo et al., 2019), we generated overlapping speech from the AMI meeting corpus (Carletta et al., 2005). We tested different overlap ratios and compared the results to real overlapping speech. Additionally, we evaluated the method using guided source separation (Kanda et al., 2019a) to extract the target speaker.

1.3.2 End-to-end multichannel speaker-attributed ASR and input feature analysis

We then propose an end-to-end multichannel SA-ASR (MC-SA-ASR) system that combines a Conformer-based encoder with MFCCA and a speaker-attributed Transformer-based decoder. To the best of our knowledge, this is the first model that efficiently integrates

ASR and speaker identification modules in a multichannel setting.

Furthermore, MC-SA-ASR can exploit spatial information which is generally advantageous for localizing different speakers. However, the predominant approaches in end-to-end multichannel ASR typically use Mel filterbank features as inputs and discard phase information (Yu et al., 2023; Zhao et al., 2021). To address this, we investigate the impact of various input features, including multichannel magnitude and phase information, on ASR performance. Additionally, we explore the effects of different multichannel inputs, including the original signals from the microphone array and beamformed multichannel signals from different angles.

1.3.3 Joint beamforming and speaker-attributed ASR

We also propose to combine SA-ASR with a beamforming-based noise and reverberation reduction front-end to improve speech and speaker recognition in far-field conditions. We then evaluate the differences in performance between statistical and neural beamformers. Finally, we jointly optimize the neural beamformer and the SA-ASR model. Our experiments on the AMI meeting corpus reveal that, while MFCCA-based channel fusion does not improve ASR performance, fine-tuning SA-ASR on the fixed beamformer's output and jointly fine-tuning SA-ASR with the neural beamformer can significantly reduce the Word Error Rate (WER).

1.3.4 Real-life meeting transcription and its optimization

Finally, we present a novel study aiming to optimize the use of an SA-ASR system in real-life scenarios, such as the AMI meeting corpus, for improved speaker assignment of speech segments. First, we propose a pipeline tailored to real-life applications involving VAD, SD, and SA-ASR. Second, we conduct several tests to enhance the pipeline's performance. On the one hand, we advocate using VAD output segments to fine-tune the SA-ASR model, considering that it is also applied to VAD segments during test, and show that this results in a significant reduction of Speaker Error Rate (SER). On the other hand, we explore strategies to enhance the extraction of the reference speaker embeddings used as inputs by the SA-ASR system. Finally, we analyse the impact of the speaker module within the SA-ASR architecture to propose long-term improvements for a more effective system design.

1.4 Publication list

Two of our contributions have been published in the following conferences:

1. **Can Cui**, Imran Sheikh, Mostafa Sadeghi, Emmanuel Vincent (2024). Improving speaker assignment in Speaker-Attributed ASR for real meeting applications. In *The Speaker and Language Recognition Workshop (Odyssey)*, pages 99–106.

2. **Can Cui**, Imran Sheikh, Mostafa Sadeghi, Emmanuel Vincent (2023). End-to-end multichannel Speaker-Attributed ASR: Speaker guided decoder and input feature analysis. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8.

This PhD work was partly conducted at Vivoka. Punctuated ASR was studied during this period. Although it is not the focus of the thesis, this work resulted in a report.

1. **Can Cui**, Imran Sheikh, Mostafa Sadeghi, Emmanuel Vincent (Nov. 2023). End-to-end joint rich and normalized ASR with a limited amount of rich training data. *arXiv preprint arXiv:2311.17741*.

1.5 Organization of the thesis

Chapter 2 introduces state-of-the-art methods for generating meeting transcripts. First, we present the individual modules in the pipeline method, such as speech separation and enhancement, speaker diarization, and ASR. Next, we discuss end-to-end systems that integrate these modules to simultaneously accomplish multiple tasks.

Chapter 3 details the pipeline method, covering speaker separation, identification, and ASR. We present our experiments on real meeting data and discuss the differences between general speaker separation and target speaker separation methods.

Chapter 4 introduces our proposed end-to-end MC-SA-ASR system, which includes an MFCCA-based multichannel encoder and a speaker-attributed decoder. We discuss experiments conducted on both synthetic and real data, examining the impact of different multichannel input features.

Chapter 5 explores the combination of beamforming and SA-ASR. Using real data, we compare various beamformers and their impact on the SA-ASR model. We also propose to jointly optimize a neural beamformer and the SA-ASR system.

Chapter 6 covers the entire pipeline for transcribing meetings from the overall meeting audio, including modules such as VAD, SD, and SA-ASR. We discuss potential improvements to the pipeline's performance and the impact of the speaker module on SA-ASR.

Chapter 7 concludes the thesis by summarizing our contributions and proposing future research perspectives based on our findings.

2 State-of-the-art

莫道君行早，更有早行人。
Do not say that you start early, for there are people earlier than you.

Chinese proverb

This chapter introduces state-of-the-art methods for generating meeting transcripts. Generating meeting transcripts involves transforming multi-speaker meeting audio, whether single-channel or multichannel, into a chronological record of what each speaker said. Existing methods can be broadly categorized into two groups. The first group involves a pipeline of functional modules, which perform different sub-tasks such as speech separation and enhancement, speaker diarization, and speech recognition. In this pipeline, the output of each module serves as the input to the subsequent module, ultimately addressing the overall task. We discuss these individual modules in Section 2.1, and the different pipeline systems from the literature in Section 2.2.1. The second group consists of end-to-end systems. Section 2.2.2 explores methods that connect different modules to simultaneously accomplish the above sub-tasks. In this chapter, and throughout the thesis for ease of annotation, the term *joint* is used broadly to describe the interconnections between different modules. It refers both to pipeline systems and end-to-end systems.

2.1 Individual modules in pipeline systems

The completion of meeting transcripts can involve several modules. As shown in Figure 2.1, speech separation separates overlapping speech signals from different speakers or sources into individual speaker or source signals, speaker diarization provides timestamps for the speaking regions of each speaker, and speech recognition transcribes input speech into text. Note that, while not every pipeline system for meeting transcription incorporates speech separation, ASR and speaker diarization are essential.

For each module, the raw audio needs to undergo specific processing to enhance

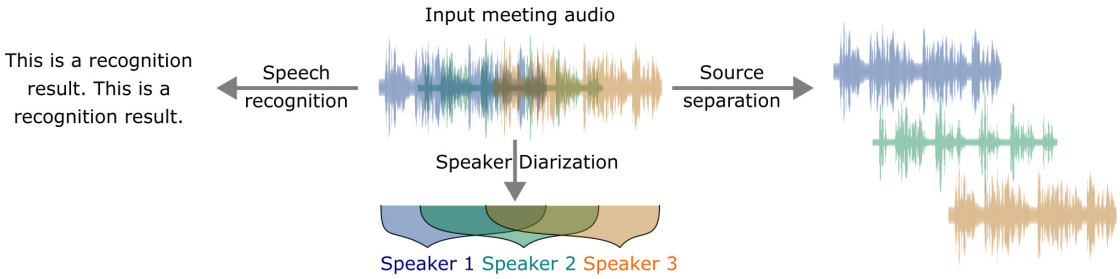


Figure 2.1: Functionalities of individual modules: speech separation, speaker diarization and speech recognition.

its features for the corresponding task, as illustrated in Figure 2.2. Feature extraction involves transforming raw speech signals into a representation that captures relevant information for subsequent analysis. The main techniques employed for feature extraction in speech separation and enhancement primarily include **Spectrograms** based on Short-Time Fourier Transform (**STFT**), which captures the spectral content within time frames. Mel-Frequency Cepstral Coefficients (**MFCCs**) are commonly used for speaker diarization and speech recognition, , which encode the spectral envelope on a perceptually motivated frequency scale. Inspired from the STFT and Mel filterbanks, **gammatone** filterbanks can be applied as alternative to capture spectral characteristics.

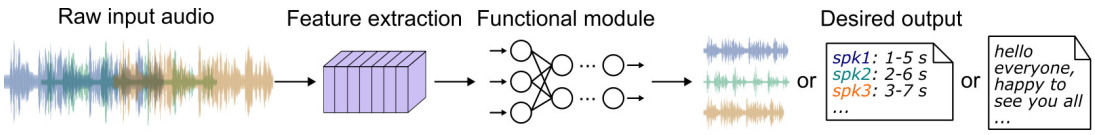


Figure 2.2: Processing diagram for each module.

The extracted features are subsequently fed into dedicated modules, such as those for speech separation, to obtain the intended outputs (see Figure 2.2). In this section, we will focus on various methods for each module, without specifically detailing the types of individual feature extraction.

2.1.1 Speech separation and enhancement

In real-world scenarios, the target speech signal is often mixed with overlapping speech, environmental sounds, and reverberation. These factors can impact the performance of Automatic Speech Recognition (ASR) systems. To address this issue, speech separation and enhancement are commonly employed. The objective of these techniques is to retrieve a signal originating from one or more sources from an observed signal that includes

additional sources and/or reverberation (Vincent et al., 2018). In this section, we will explore the problem formulation and review the existing methods for speech separation and enhancement.

2.1.1.1 Problem statement

A mixture of speech signals $x(t)$ may consist of multiple source signals, denoted as $c_j(t), j = 1, ..., J$, where J is the number of sources. For example, these sources could include multiple speakers or noises from appliances in a room. This mixture can be expressed as

$$x(t) = \sum_{j=1}^{J} c_j(t). \tag{2.1}$$

If we consider spatial effects, each source signal $c_j(t)$ is essentially a source spatial image signal (Vincent et al., 2012). In other words, if we denote the signal emitted by the actual sound source as $s_j(t)$, the source spatial image $c_j(t)$ is the signal of $s_j(t)$ after acoustic propagation reaching the recording device. More specifically, for an I-channel recording device, the mixture signal $\mathbf{x}(t) = [x_1(t), ..., x_I(t)]$, is composed of I-channel J source spatial image signals.

In the following, assuming that the first K sources are speech sources and the remaining $J - K$ ones are noise sources, we call **speech separation and enhancement** the task of extracting the speech signal $s_j(t)$ emitted by each speaker. This includes separation, dereverberation and denoising. Similarly, **speech enhancement** refers to the task of dereverberating and denoising the speech mixture without separating the sources, meaning that the desired outcome is $\sum_{j=1}^{K} s_j(t)$..

2.1.1.2 Single-channel separation and enhancement

Firstly, we discuss existing separation methods for single-channel scenarios. Mask-based separation in the time-frequency domain is a traditional method of speech separation that has been in existence since the early 2000s. Time-frequency masks operate in the time-frequency domain, where the acoustic signal is represented as a spectrogram. These masks, such as the **Ideal Binary Mask** (IBM) (Wang, 2005) and the **Ideal Ratio Mask** (IRM) (Hummersone et al., 2014), are used to selectively attenuate or enhance specific components of the spectrogram, corresponding to different sources or background noise. The IBM is a binary mask that whether the target speaker is predominant or not in each time-frequency bin. The IRM is a soft mask that represents the ratio of the target speaker's magnitude to

the total magnitude in each time-frequency bin. Both the IBM and the IRM are used in the context of mask-based methods for speech separation. These masks are applied to the mixture's time-frequency representation to obtain the separated signals corresponding to each speaker.

Traditional mask estimation methods based on signal statistics include **Spectral Subtraction** (Boll, 1979; Berouti et al., 1979), **Non-negative Matrix Factorization** (NMF) (Lee and Seung, 1999), as well as **Wiener Filtering**, among others. Spectral Subtraction estimates the mask by subtracting an estimated noise spectrum from the observed spectrum. In contrast, NMF decomposes the observed spectrogram into a non-negative linear combination of basis spectrograms. Meanwhile, Wiener filtering is a signal processing technique that aims to estimate a desired signal from a noisy observation by applying a linear filter to minimize the mean square error.

In recent years, various Deep Neural Network (DNN) architectures have been explored for single-channel speech separation. **Deep Clustering** (Hershey et al., 2016) and its extension, **Deep Attractor Networks (DAN)** (Chen et al., 2017), focus on leveraging DNNs to cluster time-frequency bins, associating them with individual speakers. More recent models prioritize end-to-end processing in the time domain. Typically, an designed encoder is employed to extract a time-frequency like representation from the input waveform, followed by a masking network that generates individual masks for each source speaker and a decoder that transforms the masked representations back into waveforms. **Conv-TasNet**, for instance, is a waveform-based approach proposed by Luo and Mesgarani (2019), which leverages a convolutional neural network (CNN) to directly operate on the time-domain waveform. **Dual-Path RNN** (DP-RNN), introduced by Luo et al. (2020b), employs a dual-path recurrent neural network architecture that captures long-term dependencies in sequential data (Figure 2.3).

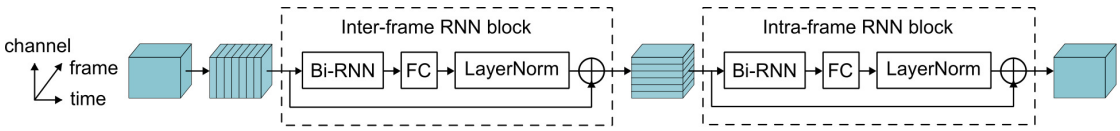


Figure 2.3: Dual-Path RNN (Luo et al., 2020b).

While RNN-based models encounter challenges in memorizing long sequences and face limitations in computational parallelization, an alternative Transformer-based model, **SepFormer** (Subakan et al., 2021), addresses these issues by implementing a masking network within a dual-path framework using the Transformer architecture (Vaswani et al., 2017). Transformers substitute recurrent computations with a multi-head attention mech-

anism (see Figure 2.4), that allows them to attend to different parts of the input sequence simultaneously. It involves running several attention mechanisms in parallel and combining their outputs, enabling the model to capture diverse aspects of the input information. This innovative design enables SepFormer to effectively capture both short and long-term dependencies, overcoming the constraints associated with traditional RNN architectures.

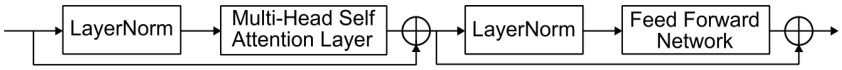


Figure 2.4: Main architecture of Transformer (Vaswani et al., 2017).

Recently, various variants of SepFormer have been introduced to address different challenges. **Tiny-SepFormer** (Luo et al., 2022) is a model designed to reduce the number of model parameters and memory size. **Av-SepFormer** (Lin et al., 2023a) employs cross- and self-attention mechanisms to integrate and model features from both audio and visual modalities. Additionally, **X-SepFormer** (Liu et al., 2023) utilizes two novel loss schemes that assess reconstruction improvement performance at a small chunk-level and leverage associated distribution information. These developments reflect a growing trend in harnessing the capabilities of Transformers.

2.1.1.3 Multichannel separation and enhancement

Multichannel audio processing introduces a wealth of spatial information, making it a valuable approach for speech separation and enhancement. By utilizing multiple microphones, spatial cues from various directions can be harnessed to improve the overall robustness and performance of these systems. One fundamental method is the **Delay-and-Sum** beamformer, where signals from different microphones are time-aligned and summed to enhance the target source and attenuate interference. Given the multichannel signal $\mathbf{x}(t)$, the delayed and summed signal y can be expressed as

$$y(t) = \sum_{i=1}^{I} x_i(t - d_i) \tag{2.2}$$

where d_i is the time delay applied to the i-th channel.

Another method called **data-independent beamformer** by Vincent et al. (2018, Chap.-10.3) is designed to extract signals from specific angles by leveraging prior knowledge about the positions of sources. Assuming that the sources are in the far field of the array (Johnson and Dudgeon, 1993), the problem can be defined as finding the beamformer $w(f)$ whose

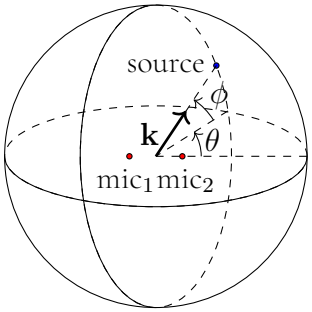


Figure 2.5: Geometrical illustration of the azimuth θ, the elevation ϕ and the unit vector $\mathbf{k}$ between the source and the microphone array center.

spatial response as a function of azimuth θ and elevation ϕ (see Figure 2.5) is closest to a predefined target $b^{\text{tgt}}(\theta, \phi, f)$:

$$\hat{\mathbf{w}}(f) = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{4\pi} \int_0^{2\pi} \int_{-\pi/2}^{\pi/2} ||\mathbf{w}^H \mathbf{d}(\theta, \phi, f) - b^{\text{tgt}}(\theta, \phi, f)||_2^2 \cos\theta, d\theta, d\phi \tag{2.3}$$

where

$$\mathbf{d}(\theta, \phi, f) = \begin{bmatrix} e^{-2\jmath\pi \mathbf{k}^T(\theta,\phi,f)\mathbf{m}_1/\lambda} \\ \vdots \\ e^{-2\jmath\pi \mathbf{k}^T(\theta,\phi,f)\mathbf{m}_I/\lambda} \end{bmatrix}, \tag{2.4}$$

is the steering vector with $\mathbf{m}_i$ the Cartesian coordinates of microphone i, where λ is the center wavelength of the narrowband signal and

$$\mathbf{k} = \begin{bmatrix} \cos\theta \cos\phi \\ \sin\theta \cos\phi \\ \sin\theta \end{bmatrix} \tag{2.5}$$

is the unit vector from the center of the microphone array to the target source.

Other approaches are also grounded in mathematical and statistical principles, such as the **Minimum Variance Distortionless Response** (MVDR) beamformer (Affes and Grenier, 1997; Gannot et al., 2001) and the **Multichannel Wiener Filter** (MWF) beamformer (Doclo et al., 2009). MVDR beamforming estimates the beamformer weights to minimize interference while preserving sound in a given target direction-of-arrival (DOA). The MWF, by contrast, relies on the multichannel covariance matrices of the target source and interference to estimate the target source in the mean squared error sense. A range of methods

based on multichannel Gaussian models have been proposed to estimate those multichannel covariance matrices, such as multichannel nonnegative matrix factorization (Ozerov and Févotte, 2009).

In recent times, increased attention has been directed towards signal separation through the utilization of DNNs. Nugraha et al. (2016) employ DNNs to model source spectra, which are then integrated with the classical multichannel Gaussian model to leverage spatial information. More recently, advanced end-to-end multichannel separation models have been proposed. For instance, Gu et al. (2019) introduced an end-to-end model that replaces the conventional (STFT) and inter-channel phase difference features by a function of time-domain convolution with a specialized kernel. Luo et al. (2019) introduced a **Filter-and-Sum Network** (FaSNet) characterized by a two-step process. In the initial stage, the model devises frame-level time-domain adaptive beamforming filters specifically for a chosen reference channel. Subsequently, it computes filters for all other channels. The ultimate output is produced by summing the filtered outputs across all channels. Additionally, a **Transform-Average-Concatenate** (TAC) (Luo et al., 2020a) scheme has been suggested to enhance the FasNet framework. It focuses on achieving channel permutation and number invariance for multi-channel speech separation within the core architecture of DP-RNN (see Figure 2.6).

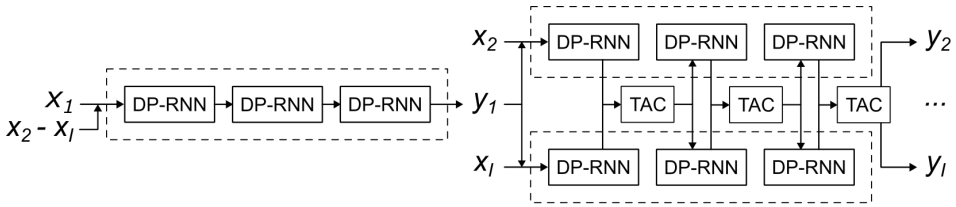


Figure 2.6: FasNet architecture with TAC scheme (FasNet-TAC) (Luo et al., 2020a).

Research interest in applying Transformer architecture to multichannel separation is growing. Quan and Li (2022) introduced a multichannel speech separation method incorporating a **narrow-band Conformer** (Gulati et al., 2020), which autonomously leverages narrow-band speech separation information. Wang et al. (2022) proposed **DE-DPCTnet**, utilizing a dual-path convolutional Transformer in the FaSNet-TAC modules. While many state-of-the-art methods typically address single-channel and multichannel scenarios separately, Wang et al. (2023b) introduced a unified architecture, **DasFormer**, that employs multi-head self-attention in different blocks for both scenarios.

2.1.2 Speaker diarization

Speaker diarization is the process of assigning speaker identities to segments of an audio recording. Diarization in meeting transcription involves automatically distinguishing and labeling individual speakers in a recorded meeting. This enables the accurate attribution of spoken content to specific participants, enhancing the quantity and clarity of meeting transcriptions. It relies on speaker recognition techniques, particularly clustering of speaker representation embeddings. Two primary diarization approaches exist: modular clustering-based methods and end-to-end methods (Park et al., 2022; Haeb-Umbach, 2023). Modular methods involve distinct steps like audio segmentation, feature extraction and clustering, while end-to-end methods aim for a seamless integration of these processes.

2.1.2.1 Speaker embedding for speaker recognition

Fundamental speaker recognition relies on clustering speaker representations, where the representations of the same speaker exhibit similarity. Evolutionary strides in speaker representation techniques unfold a trajectory from **i-vectors** (Dehak et al., 2010), where factor analysis establishes a low-dimensional space capturing speaker-specific and channel-related variabilities, to **d-vectors** (Variani et al., 2014) that leverage DNNs for frame-level acoustic feature-based classification. The journey continues with **x-vectors**, as introduced by Snyder et al. (2018), employing a Time Delay Neural Network (TDNN) (Lang et al., 1990) for effective temporal context representation. Departing from recurrent LSTM units, TDNNs allow seamless sentence-level training, offering various x-vector variations like **ECAPA-TDNN** (Desplanques et al., 2020). As shown in Figure 2.7, ECAPA-TDNN introduces 1D Residual Neural Networks (Res2Net) modules and Squeeze-and-Excitation blocks for enhanced channel interdependencies and temporal context expansion. Additionally, hierarchical features are leveraged through aggregation and propagation, and an improved statistics pooling module with channel-dependent frame attention is introduced.

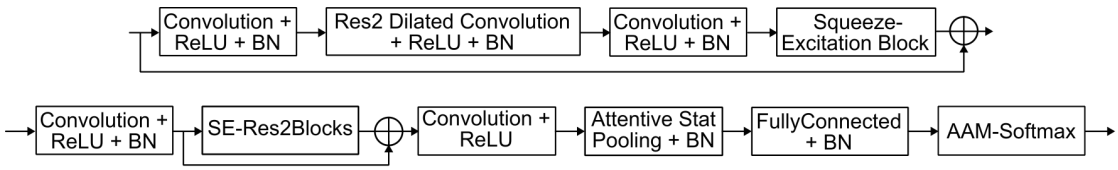


Figure 2.7: SE-Res2Block (top) and the overall ECAPA-TDNN (Desplanques et al., 2020) architecture (down). BN: Batch normalization, AAM: Additive angular margin loss.

In the current speech processing landscape, the impact of the Transformer architecture extends beyond speech separation and enhancement into the domain of speaker

embedding. Mary et al. (2021) introduced **s-vectors**, derived from a specially trained Transformer encoder for speaker classification. Additionally, Sang et al. (2023) **enhanced Transformer-based speaker embedding** through improved locality modeling, introducing the Locality-Enhanced Conformer (LE-Confomer) with depth-wise convolution and channel-wise attention. They also adapted the Swin Transformer for vision tasks into the Speaker Swin Transformer (SST) for further advancements in speaker embedding.

2.1.2.2 Modular clustering based methods

Traditional speaker diarization systems are modular, consisting of various modules. As outlined in the review by Park et al. (2022), those modules include **pre-processing techniques, voice activity detection, segmentation, speaker embedding** (see Figure 2.8), **clustering** and **post processing**. The front-end processing involves tasks such as speech enhancement, dereverberation, and speech separation, discussed in Section 2.1.1. The objective is to enhance speech quality to facilitate the subsequent speaker recognition task. Following this, a voice activity detection (VAD) module is employed to extract speech segments from the overall audio. This step is important for speaker recognition, as speaker embeddings are meaningful only within non-silent regions. Contemporary VAD systems heavily leverage **DNNs**. Noteworthy approaches include a **CNN-based method** proposed by Thomas et al. (2014), an **RNN-based** one by Gelly and Gauvain (2017), and the widely adopted **CRDNN** (Convolutional, Recurrent, and Dense Neural Network) by Sainath et al. (2015b) and Xiang et al. (2021). CRDNNs integrate convolutional layers for handling frequency variations, recurrent layers for temporal modeling, and dense layers for effective feature mapping within VAD applications.

Figure 2.8: Modules for speaker diarization (Park et al., 2022).

In speaker diarization, speech segmentation involves dividing the input audio stream into segments to achieve speaker-uniform sections (Park et al., 2022). Initially, segmentation relied on speaker-change detection (Chen and Gopalakrishnan, 1998; Kemp et al., 2000). It was later replaced by uniform segmentation schemes (Wang et al., 2018; Sell et al., 2018), where the audio stream is segmented using fixed-length overlapping windows. The segmented audio is subsequently transformed into speaker embeddings, such as **d-vectors** or **x-vectors** as discussed in the previous subsection, before undergoing the clustering process. Clustering methods include **Agglomerative Hierarchical Clustering** using metrics like

KL distance (Siegler et al., 1997), **GLR** (Gish et al., 1991; Bonastre et al., 2000; Gangadharaiah et al., 2004), or **BIC** (Chen and Gopalakrishnan, 1998; Tritschler and Gopinath, 1999). **Spectral Clustering** stands out as a widely employed technique (Ning et al., 2006; Park et al., 2019b). By harnessing the eigenvalues and eigenvectors of a similarity matrix, spectral clustering adeptly partitions data points into cohesive groups. This method's strength lies in its ability to transform data into a lower-dimensional spectral space, facilitating the effective identification of intricate non-linear structures within the underlying dataset. Finally, some post-processing in a diarization system involves refining the output of the initial diarization process to improve its accuracy and coherence (Park et al., 2022).

2.1.2.3 End-to-end diarization methods

The second category of diarization systems involves end-to-end approaches. Fujita et al. (2019a, 2020) introduced and developed an End-to-End Neural Diarization (**EEND**) method which employs a bidirectional LSTM network to directly produce speaker diarization outcomes for multi-speaker recordings. Subsequently, they enhanced it to a Self-Attention (SA) based EEND (Fujita et al., 2019b) method. Following this, Horiguchi et al. (2020) proposed the Encoder-Decoder based Attractor (**EDA**), adding flexibility to EEND in terms of the number of speakers. A similar objective was pursued in another work by Han et al. (2021). Liu et al. (2021) evolved **SA-EEND** from a Transformer-based to a Conformer-based architecture, enhancing its capabilities. More recently, variants such as **EEND-DEMUX** (Mun et al., 2023) and **EEND-M2F** (Härkönen et al., 2024) were introduced to further improve performance.

As highlighted in the review of Park et al. (2022) and discussed by Haeb-Umbach (2023), modular-based methods excel in preserving global speaker representations and exhibit versatility across various scenarios. However, they face challenges in effectively handling overlapping regions and are less adept at managing rapid speaker changes. In contrast, end-to-end methods prove to be more effective in addressing these challenges, but have limited performance across different scenarios.

To enhance the robustness of diarization systems, several methods have been proposed. Araki et al. (2008) introduced a DOA-based system aimed at reducing reverberation and noise. Huang et al. (2020) developed a region proposal network-based Speaker Diarization system that effectively handles overlapped speech. More recently, Zheng et al. (2021) proposed a spatial spectrum-based approach that facilitates speaker localization and identification, thereby minimizing noise from other directions.

2.1.3 Automatic speech recognition

Automatic Speech Recognition (ASR) converts spoken language into text. In the process of meeting transcription, ASR serves as a foundational step, providing the textual representation of the meeting content (see Figure 2.1).

2.1.3.1 GMM-HMM and early neural network systems

During the 1990s and early 2000s, comprehensive research efforts were dedicated to exploring the framework of Hidden Markov Models with Gaussian Mixture Model emission distributions (GMM-HMM). HMMs are widely utilized in various fields, including speech recognition, due to their ability to model sequences of observations with discrete hidden states. In the context of ASR, HMMs are employed to represent the temporal dynamics of spoken language, capturing transitions between successive phonemes (Pujol et al., 2004; Trentin and Gori, 2001). By utilizing a combination of Gaussian distributions, GMMs can capture the various acoustic realizations of each phoneme inherent to spoken language (Wang et al., 2019). While HMM-based models prove effective in various scenarios, they encounter significant limitations that restrict their applicability in real-world environments, such as poor discriminative power and reliance on strong statistical assumptions (Trentin and Gori, 2001).

Numerous studies on neural networks for recognition tasks based on acoustic features have been proposed since the 1980s (Waibel, 1989; Hampshire and Waibel, 1989; Waibel et al., 1989). However, the initial implementations of these technologies were not highly efficient. Later, a hybrid system, known as DNN-HMM was proposed by Bourlard and Morgan (1994). This system leverages the power of DNNs for feature extraction and modeling, while HMMs are used for sequence modeling and decoding.

The introduction of DNN-HMM systems marked a significant improvement in recognition performance by combining the strengths of neural networks and HMMs. Despite this, early DNN-HMM systems faced challenges, such as limited computational power and insufficient training data, which constrained their effectiveness. However, advancements in hardware and the availability of large datasets in the 2010s led to substantial progress. Researchers like Sainath et al. (2015a), Deng and Li (2013), and Saon and Chien (2012) made significant contributions to the development of DNNs for speech recognition during this period. These deep architectures enabled more accurate and robust acoustic modeling, ultimately improving the performance of DNN-HMM systems.

2.1.3.2 End-to-end recognition models

End-to-end models entail a direct mapping from input audio sequences to text sequences. In contrast to HMM-based models whose performance heavily depends on acoustic conditions, end-to-end methods exhibit greater robustness across various audio conditions. This increased robustness is largely due to their access to larger and more diverse training datasets. End-to-end ASR systems are structured around two key components: the Encoder and the Decoder. The Encoder processes input audio sequences, extracting relevant features and representing them in a condensed form. Subsequently, the Decoder takes this condensed representation and generates the corresponding text sequences, completing the end-to-end mapping from spoken language to transcriptions. This architecture eliminates the need for intermediate steps like phoneme recognition or language modeling, providing a more seamless and direct approach to transforming audio into text.

In the realm of end-to-end ASR systems, various types of **encoders** have been proposed to transform input audio sequences into meaningful representations. The earliest end-to-end ASR systems use DNNs as encoders, thereby eliminating the need for GMM models (Graves and Jaitly, 2014; Hannun et al., 2014; Chorowski et al., 2014; Chan et al., 2016; Bahdanau et al., 2016). Then, Convolutional Neural Network **(CNN) encoders** (Hayashi et al., 2018; Hori et al., 2017) harness convolutional layers to capture local patterns and hierarchical structures in the audio spectrogram. Recurrent Neural Network **(RNN) encoders** (Li et al., 2019; Karita et al., 2019), specifically Long Short-Term Memory (**LSTM**) architectures (Zeyer et al., 2019; Li et al., 2021a), effectively capture temporal dependencies and the latter mitigate the vanishing gradient problem. Building on the Transformer architecture (Vaswani et al., 2017), **Transformer-based encoders** (Dong et al., 2018; Karita et al., 2019) leverage self-attention mechanisms to capture long-range dependencies and facilitate parallelization. In more recent developments, Gulati et al. (2020) introduced a **Conformer-encoder** architecture for ASR, seamlessly integrating convolutions with self-attention. This innovative design is characterized by a configuration where these components are interposed between two feed-forward modules, as depicted in Figure 2.9.

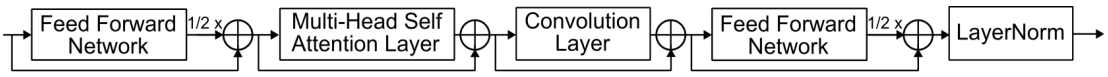


Figure 2.9: Conformer encoder architecture in end-to-end ASR (Gulati et al., 2020).

End-to-end ASR decoders serve the crucial function of converting acoustic representations into textual transcriptions. Three main types of ASR decoders are commonly employed: Connectionist Temporal Classification (**CTC**) (Graves et al., 2006), **Transducer**

(Graves, 2012) and **Attention-based Encoder-Decoders (AED)** (Dong et al., 2018; Karita et al., 2019; Zeyer et al., 2019). CTC establishes an alignment between input and output sequences, without considering the connections between the outputs. The Transducer decoder expands upon CTC by creating a distribution over output sequences of varying lengths and by simultaneously modeling the dependencies between inputs and outputs, as well as among the outputs themselves. Furthermore, the Transducer only considers the preceding states during prediction, whereas AED takes into account the entire sequence through an attention mechanism. Specifically, in a Transformer decoder (see Figure 2.10), in addition to self-attention, Encoder-Decoder attention is another crucial component that aligns the Decoder's current position with relevant information in the Encoder's output sequence. This attention mechanism enables the decoder to focus on different parts of the input sequence during the decoding process, facilitating effective sequence-to-sequence transformations.

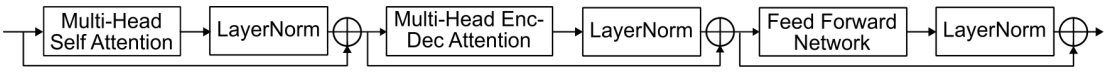


Figure 2.10: Transformer decoder architecture (Zeyer et al., 2019) in end-to-end ASR.

In conclusion, ASR has undergone a transformative shift from the traditional GMM-HMM framework to contemporary end-to-end encoder-decoder systems. This evolution leverages neural network advancements to directly capture intricate speech patterns from raw audio, offering enhanced efficiency and robustness.

In addition, several methods have been proposed to improve the robustness of ASR models. Maas et al. (2012) introduced an RNN encoder for denoising input features. Methods such as noise adaptive training (Kalinli et al., 2010) and SpecAugment (Park et al., 2019a) aim to vary the training conditions. Additionally, some studies have proposed integrating a speech enhancement module with ASR (Menne et al., 2019; O'Malley et al., 2021).

2.2 Meeting transcription systems

In the above section, we have provided a concise overview of the three key functionalities required for meeting transcription. Speech separation and enhancement are essential for isolating distinct voices in a multi-speaker environment. Speaker diarization aids in identifying and distinguishing speakers. ASR converts the audio into text. A meeting transcription system requires the integration of all these functionalities. In this section, we delve into state-of-the-art joint systems designed for generating meeting transcriptions.

These joint systems fall into two categories: pipeline systems, consisting of multiple modules where the output of previous module(s) feeds into the next one; and end-to-end systems, where a single module produces outputs for various functionalities (as illustrated in Figure 2.11).

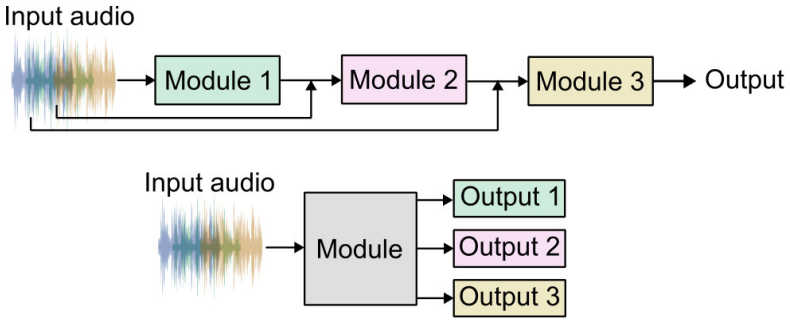


Figure 2.11: Pipeline system (top) and end-to-end system (down) for automatic meeting transcriptions.

2.2.1 Pipeline systems

Meeting transcription pipeline systems can feature various combinations of diarization, separation, and ASR modules. The arrangement of these modules can impact performance differently. Raj et al. (2021) and Haeb-Umbach (2023) examined the three primary types of meeting transcription pipelines: diarization + separation + ASR, separation + ASR + diarization, and separation + diarization + ASR. We provide detailed examples of each of these pipelines below.

2.2.1.1 CHiME-7 DASR Pipeline: Diarization + Separation + ASR

The challenges associated with multi-talker distant speech recognition have prompted the establishment of challenges and datasets, such as AMI (Carletta et al., 2005), ASpiRE (Harper, 2015), ICSI (Janin et al., 2003) and the CHiME 1-6 challenges (Barker et al., 2013; Vincent et al., 2013; Barker et al., 2017; Vincent et al., 2016; Barker et al., 2018; Watanabe et al., 2020). The latest iteration, CHiME-7 distant ASR (DASR) challenge (Cornell et al., 2023), underscores the importance of developing systems capable of generalizing across diverse real-world scenarios and delivering robust ASR performance in challenging acoustic conditions. The CHiME-7 authors have introduced a baseline pipeline, outlined in Figure 2.12, to address these objectives.

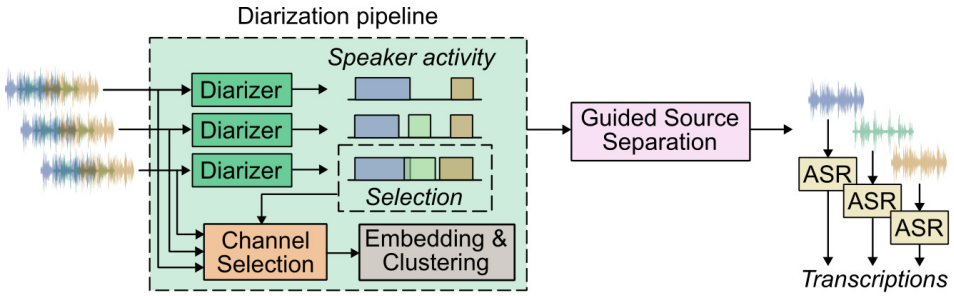


Figure 2.12: CHiME-7 DASR pipeline (Cornell et al., 2023).

The CHiME-7 DASR pipeline comprises several key components, including multi-channel diarization, channel selection, guided source separation (GSS), and single-channel ASR. Due to variations in diarization errors across different microphones within the same array (Arora et al., 2020; Cornell et al., 2023), the Diarizer is applied individually to each channel. The channel selection module is employed to identify the channel with the most speech activity, which is then utilized to extract speaker embeddings and perform clustering in the diarization pipeline. For segments encompassing multiple speakers, the chosen channel is used in the GSS system for speaker separation, leveraging diarization annotations. The resulting single-speaker speech segments are then passed to an ASR system for transcription.

The Diarizer relies on the end-to-end diarization (EEND) (Fujita et al., 2019b; Bredin and Laurent, 2021) method implemented in the Pyannote toolkit (Bredin et al., 2020). For dereverberation in the GSS system proposed by Boeddeker et al. (2018), front-end processing techniques like Weighted Prediction Error are utilized. The separation system leverages time annotations from the diarization system, indicating when a specific speaker is active, to enhance active speaker separation. The GSS system also incorporates a Minimum Variance Distortionless Response (MVDR) beamformer (Yoshioka et al., 2019). In terms of ASR, the authors employ a hybrid CTC/Attention Transformer (Vaswani et al., 2017) encoder-decoder ASR model with WavLM-based features. Additionally, they employ an augmentation scheme (Watanabe et al., 2020) that utilizes close-talk microphones and external datasets for Room Impulse Responses and noises, specifically SLR26 (Ko et al., 2017) and MUSAN (Snyder et al., 2015).

Several recent studies have responded to the CHiME-7 DASR challenge by proposing modifications to the original pipeline. For example, Wang et al. (2023a) developed a channel selection method that adapts to various array geometries using the signal-to-interference-plus-noise ratio. Additionally, they created a speaker diarization system that utilizes long-

term spatial information iteratively. Ye et al. (2023) employed target speaker voice activity detection (TS-VAD)-based speaker diarization and time-domain SpeakerBeam-based single-channel target speaker extraction (TSE) to enhance speaker diarization. Kamo et al. (2023) replaced the baseline diarization system with an end-to-end diarization with vector clustering (EEND-VC) system and developed four powerful ASR back-ends.

2.2.1.2 CSS Pipelines: Separation + ASR + Diarization

While the CHiME-7 DASR pipeline places the diarization module before separation and ASR, other pipelines, such as those presented by Chen et al. (2020) and Raj et al. (2021), opt to perform separation as the initial step. However, this approach poses a challenge for the separation system, as most speech separation studies utilize fully overlapped generated data, which does not correspond to natural conversations containing both overlapped and overlap-free regions (Chen et al., 2020). In an effort to address this gap, Yoshioka et al. (2018a) introduced continuous speech separation (CSS) which employs a permutation invariant training (PIT) framework tailored for multichannel scenarios. Instead of traditional time-frequency masking, mask-driven MVDR beamforming is adopted in CSS to leverage spatial information from the multichannel audio capture.

Various CSS pipeline variants exist, with the initial accomplishment of the CSS pipeline achieved in an audio-visual context by Yoshioka et al. (2019). As illustrated in Figure 2.13, the multichannel audio undergoes a dereverberation process to enhance audio quality. Subsequently, the audio is fed into the CSS module, which generates separated audio streams for different speakers. The output from CSS is then directed to an ASR module, producing a sequence of time-marked recognized words. Concurrently, both the separated audio and the dereverberated audio are routed to a sound source localization module, enhancing the efficacy of speaker diarization. Finally, the separated audio, sound source localization output, ASR results, and video features are collectively input into the diarization module to ascertain speaker identities at the word level.

In the pipeline presented by Yoshioka et al. (2019), dereverberation is achieved through Weighted Prediction Error (WPE) minimization. The separation module involves Bidirectional Long Short-Term Memory (BLSTM)-based GSS using MVDR beamforming. The sound source localization generative model is characterized by employing a complex angular central Gaussian model, as proposed by Ito et al. (2016). For speaker diarization, the process incorporates the time-marked tokens from the ASR output and an embedding-based speaker identification model, utilizing convolutional layers augmented by residual blocks (He et al., 2016). The ASR system itself follows a conventional hybrid structure,

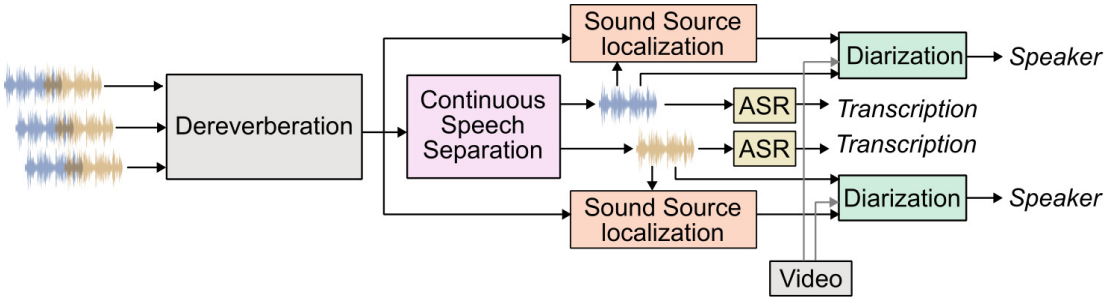


Figure 2.13: CSS pipeline of Yoshioka et al. (2019).

comprising a latency-controlled bidirectional LSTM acoustic model and a weighted finite state transducer decoder.

Subsequently, Chen et al. (2020) created the LibriCSS dataset, recorded using LibriSpeech utterances, to facilitate CSS research. They presented a CSS pipeline designed to process continuous speech through both the CSS and ASR modules. In the initial stage, the CSS pipeline employs separation, utilizing a speaker-independent CSS approach proposed by Yoshioka et al. (2018a). This CSS method generates a fixed number N of audio streams, each containing at most one active speaker at any given time. In segments without speaker overlap, the CSS algorithm routes incoming speech to one output channel, while other channels remain silent. Chen et al. (2020) set number of audio streams to two, considering that occurrences of three speakers overlapping are rare. The CSS pipeline supports streaming processing with a sliding window approach. This window comprises past, current, and future contexts, improving mask estimation accuracy by providing acoustic context. Only masks within the current subwindow are utilized, and the window shifts by the frame size for continuous processing.

The pipeline suggested by Chen et al. (2020) does not include diarization. Consequently, Raj et al. (2021) extended the pipeline by integrating a diarization module after ASR, illustrated in Figure 2.14. The inclusion of a diarization module following ASR may potentially enhance diarization performance, as reduced false alarms can be achieved when obtaining segments from the ASR output.

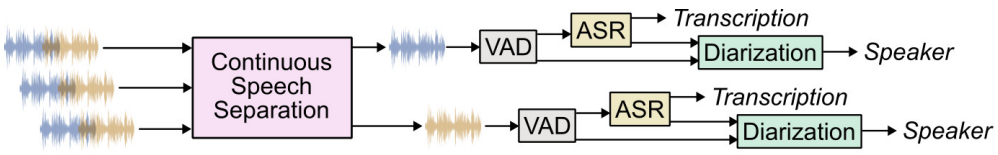


Figure 2.14: CSS pipeline of Raj et al. (2021) and von Neumann et al. (2023).

More recently, von Neumann et al. (2023) introduced a comparable pipeline with the same architecture illustrated in Figure 2.14. They implemented a syntactically informed diarization module by leveraging sentence- and word-level boundaries from the ASR module to facilitate speaker turn detection. Their pipeline employs a single-channel CSS with the TF-GridNet separation architecture (Wang et al., 2023c,d). Following the separation, an energy-based Voice Activity Detection (VAD) module is utilized to identify utterance boundaries in the separated signals. For ASR, the Whisper (Radford et al., 2023) system is employed, providing transcription along with sentence- and word-level boundaries. Finally, speaker identification is performed using d-vector-based embeddings (Cord-Landwehr et al., 2023) and k-Means++ clustering (Arthur and Vassilvitskii, 2007).

Finally, the enhancement of CSS, core module in this pipeline, has been a recent focus of study. Chen et al. (2021) introduced the Conformer architecture for the separation model, demonstrating its effectiveness in real meeting scenarios with both single-channel and multichannel settings. Additionally, Wang et al. (2021b) proposed a multi-microphone complex spectral mapping approach to address both speaker separation and dereverberation within the context of CSS. Furthermore, Yoshioka et al. (2022) introduced VarArray, a speech separation neural network model that is agnostic to array geometry, making it applicable to scenarios with any number of microphones.

2.2.1.3 Pipeline of Raj et al. (2021): Separation + Diarization + ASR

In the two pipelines above, ASR is directly followed by separation. However, in this sequential setup, separation errors, such as having two active speakers in one separated segment, are propagated to the ASR module. To address this challenge, Raj et al. (2021) proposed a novel pipeline order: separation, diarization, and ASR.

The distinctive feature of this design is that the diarization module serves not only for speaker identification but also for adjusting the relationship between the separation result and the input to ASR. As described in Figure 2.15, after a regular CSS, the output segments may contain errors. If we provide transcription and speaker IDs to these segments, as in the CSS pipeline, errors may be carried over to the final result. Therefore, Raj et al. (2021) suggested to incorporate a diarization module to extract a speaker embedding for each segment. These embeddings are then provided to the decoder of a speaker-biased ASR to enhance the generation of transcriptions in overlapping segments.

For each of the CSS, diarization, and ASR modules, Raj et al. (2021) experimented with different models. Specifically, the models for CSS include masked-based MVDR (Yoshioka et al., 2019) and sequential multi-frame separation (Wang et al., 2021a). The diariza-

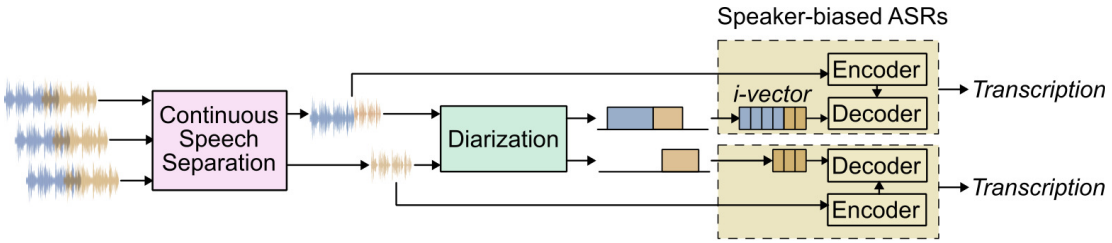


Figure 2.15: Pipeline of automatic meeting transcription proposed by Raj et al. (2021).

tion models tested are x-vector (Snyder et al., 2018) + clustering, Region Proposal Networks (RPNs) (Huang et al., 2020), and Target-Speaker Voice Activity Detection (TS-VAD) (Medennikov et al., 2020). As for ASR, TDNN-F-based (Povey et al., 2018) hybrid DNN-HMM (Manohar et al., 2019) and Transformer-based end-to-end ASR (Karita et al., 2019) models are employed. More precisely, the scheme as in Figure 2.15 used an MVDR separation model, spectral clustering-based diarization, and a hybrid DNN-HMM ASR model. The hybrid DNN-HMM model, introduced by Peddinti et al. (2015), employs a 2-pass decoding strategy. The first pass utilizes i-vectors extracted from entire utterances, focusing on reliable regions with confidence measures and duration less than 1 s. This excludes silence, fillers, and ideally overlapping speech. The second pass uses i-vectors extracted from statistics obtained solely from these reliable regions. Their experimental results proved the effectiveness of their design.

In pipeline systems encompassing separation, diarization, and ASR, there is no universally optimal combination of these three modules. Each discussed pipeline exhibits unique characteristics. The CHiME-7 DASR pipeline (Cornell et al., 2023), guided by diarization for source separation, demonstrates improved separation performance. The CSS pipeline (Chen et al., 2020), incorporating CSS, aligns the separation task with real meeting scenarios. In the pipeline proposed by Raj et al. (2021), the integration of diarization between CSS and ASR enhances ASR robustness to potential separation errors. Furthermore, the CHiME-7 DASR pipeline operates in offline mode, whereas the CSS pipeline and the pipeline of Raj et al. (2021) are capable of functioning in a streaming mode. In summary, the choice of whether to place the separation or diarization module first depends on various considerations, as outlined by Boeddeker et al. (2024). Starting with speech separation can, on the one hand, eliminate acoustic distortions and simplify sub-tasks in varying the audio conditions. On the other hand, conducting diarization on non-overlapping speech is considerably simpler than on overlapped speech. Alternatively, starting with diarization has its advantages, as the information about segment boundaries for speech absence,

single-speaker, and multi-talker speech is provided, facilitating the separation task.

2.2.2 End-to-end and joint optimization systems

In contrast to a pipeline system, where modules are trained and utilized independently, end-to-end systems incorporate modules that provide outputs with multiple functionalities, as well as modules that are jointly trained. It's important to note that while the overall transcription system might not be entirely end-to-end, this section focuses on cases where specific parts of the pipeline are implemented using an end-to-end approach.

2.2.2.1 End-to-end separation and diarization

As discussed in Section 2.2.1, the order of separation and diarization modules in the pipeline is subject to different considerations, highlighting the interdependence of these two modules and indicating the potential for a joint treatment (Boeddeker et al., 2024). This section introduces state-of-the-art models that address separation and diarization concurrently, providing separated sources with associated speaker identities. To complete the meeting transcription process, an ASR model is applied for the transcription of the separated sources, as illustrated in Figure 2.16.

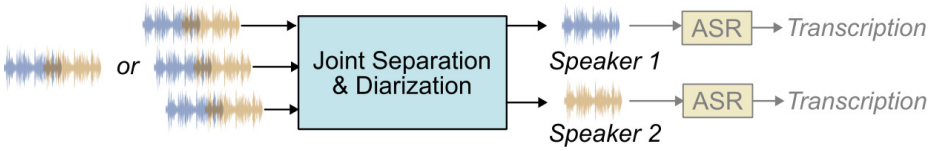


Figure 2.16: Pipeline including a joint Separation and Diarization module. In Section 2.2.2.1 and Section 2.2.2.2, the terms in gray represent the complementary steps of the systems discussed, to complete the overall meeting transcription task.

The Recurrent Selective Attention Network (RSAN), introduced by Von Neumann et al. (2019), presents a pioneering all-neural method for simultaneous speaker counting, diarization, and source separation. This Neural Network-based estimator follows a block-online approach, enabling it to monitor speakers continuously, even during periods of silence spanning multiple time blocks. Subsequently, Takahashi et al. (2019) suggested integrating the time-domain audio separation network, TasNet (Luo and Mesgarani, 2019) (refer to Section 2.1.1.3), into the RSAN framework. Furthermore, to enhance robustness in real meetings featuring spontaneously-speaking speakers, significant noise, and reverberation, Kinoshita et al. (2020) introduced a decoding scheme aimed at mitigating overestimation errors in source counting.

The RSAN framework still relies on speaker embedding for speaker diarization. However, Maiti et al. (2023) introduced a joint framework that performs diarization in end-to-end mode. This framework, called EEND-SS, incorporates end-to-end neural diarization (EEND) models (Fujita et al., 2019a,b, 2020) for speaker counting, utilizing encoder-decoder-based attractors (EDA) (Horiguchi et al., 2020), as well as speech separation through Conv-TasNet. Additionally, the framework includes separation masks corresponding to a variable number of speakers and employs a fusion technique to refine the separated speech signal with the acquired speaker diarization information. In a more recent development, Boeddeker et al. (2024) introduced a joint diarization and separation approach called TS-SEP. This method extends the target speaker voice activity detection (TS-VAD) diarization approach, which provides the initial speaker embeddings. Notably, the final combined speaker activity estimation network in TS-VAD is replaced with a network that generates speaker activity estimates at a time-frequency resolution. These estimates function as masks for source extraction, achieved either through masking or beamforming techniques.

2.2.2.2 Joint optimization of separation and ASR

Real meeting conditions, marked by heavy speech overlap, severe noise, and reverberation, present a challenge for ASR. One approach to address this challenge is to employ a front-end module, as mentioned in Section 2.1.1, to facilitate the recognition task. Several studies have delved into the joint treatment of separation and ASR modules (Kanda et al., 2019a; von Neumann et al., 2020a; Li et al., 2021b). Given the distinct nature of the tasks for these two modules—generating audio masks for separation and predicting words for ASR—it is not feasible to design a single fully end-to-end architecture that takes in mixed audio and simultaneously outputs separated audio and transcription. Instead, the joint treatment of these two modules is achieved through a form of joint optimization of their parameters. In the following, we present various methods for single- and multichannel audio to jointly optimize speech separation (possibly with enhancement) and speech recognition. For a comprehensive meeting transcription system, complementary speaker diarization is necessary to obtain speaker identity information (refer to Figure 2.17).

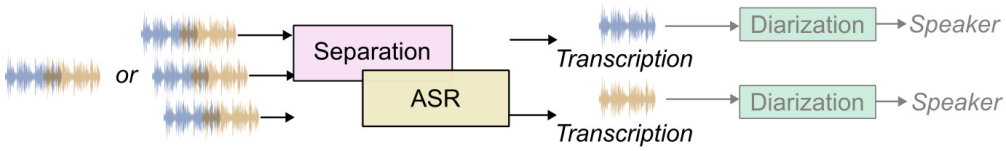


Figure 2.17: Pipeline including a joint Separation and ASR module.

In a single-channel setting, von Neumann et al. (2020a) introduced the first jointly optimized multi-talker ASR system for an unknown number of speakers. This approach combines source separation and speaker counting techniques with a single-speaker CTC / attention-based speech recognizer. The authors initially explored a Dual-Path Recurrent Neural Network (DPRNN)-TasNet (Luo et al., 2020b) separator for a fixed number of speakers as a front-end for ASR. They performed joint fine-tuning of the DPRNN with a pre-trained ASR (von Neumann et al., 2020b) system. Subsequently, the DPRNN was integrated into the One-and-Rest Permutation Invariant Training (OR-PIT) architecture, which was extended with mechanisms for speaker counting. Finally, the OR-PIT architecture was combined with an ASR system to create a new multi-speaker ASR system designed to handle an unknown number of speakers. Later, Wu et al. (2021) focused on enhancing training objective functions to establish a stronger connection between separation and ASR tasks. Specifically, after pre-training the separation model using the ideal amplitude mask (IAM) as the training target, they fine-tuned the separation model with the training criteria of ASR, which combines CTC and attention-based objective functions. Their findings demonstrate that aligning the separation objective with the ASR objective leads to improved performance in ASR.

In a multichannel setting, Hitachi and Paderborn University collaborated to address the CHiME-5 challenge (Barker et al., 2018). Specifically, in the work of Kanda et al. (2019a), they applied speech enhancement with GSS to the mixture, extracting separated audio for subsequent ASR processing. In GSS, they initially utilized time annotations provided in the dataset to estimate the number of active speakers. Additionally, fine-tuned annotations were obtained by a strong ASR system following VAD. The incorporation of ASR output to fine-tune the GSS system illustrates the principle of joint optimization. In terms of speech enhancement (Boeddeker et al., 2018), they employed (Weighted Prediction Error) WPE for dereverberation and a MVDR beamformer. The GSS system comprised a spatial mixture model using complex angular central Gaussian distributions (Ito et al., 2016). The ASR system (Kanda et al., 2018) employed an acoustic model consisting of a convolutional neural network (CNN), a time delay neural network (TDNN), and a residual bidirectional long short-term memory (RBiLSTM) network.

Lastly, recent studies, such as those conducted by Li et al. (2021b) and Shi et al. (2022), emphasize the consideration of both single- and multichannel audio inputs. The shared key concept in these studies involves back-propagating the ASR loss to the front-end separation module, enabling the joint training of the two modules.

2.2.2.3 End-to-end ASR and diarization

In all the methods discussed previously, separation plays a vital role by providing separated audio to the ASR. However, with the growing emphasis on end-to-end multi-speaker ASR (Kanda et al., 2020b), there is a possibility to bypass the separation step and obtain the transcription along with speaker identity simultaneously, as depicted in Figure 2.18.

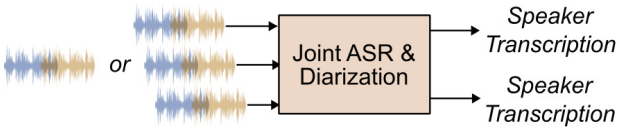


Figure 2.18: Pipeline including a joint Diarization and ASR module.

In the realm of multi-speaker ASR, earlier studies adopted PIT (Yu et al., 2017b,a; Seki et al., 2018), employing multiple output layers corresponding to different speakers. The model was trained by considering all possible speaker permutations, but this approach increased the model size and is constrained by the number of speakers. To address this in an end-to-end mode, Kanda et al. (2020b) introduced the concept of Serialized Output Training (SOT). This innovative approach involves only one output layer that generates the transcriptions of multiple speakers one after another in a first-in-first-out mode. This is realized in an attention-based encoder-decoder end-to-end ASR system, where generating the transcript of one speaker directs the attention to the region corresponding to that speaker.

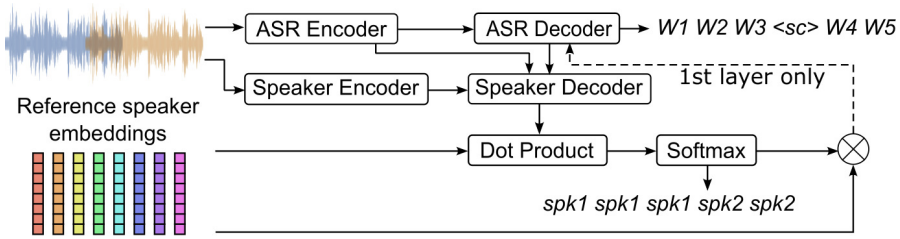


Figure 2.19: Speaker-Attributed ASR (Kanda et al., 2020a).

Employing the SOT concept, Kanda et al. (2020a) introduced an end-to-end system named Speaker-Attributed ASR (SA-ASR), which jointly addresses Speaker Counting, Speech Recognition, and Speaker Identification. Illustrated in Figure 2.19, the model comprises an ASR Encoder, an ASR Decoder, a speaker Encoder, and a Speaker Decoder. The inputs to the model consist of an acoustic feature sequence $X \in \mathbb{R}^{L \times A}$ where L is the sequence length and A the feature dimension, and a matrix $S \in \mathbb{R}^{E \times K}$ of E-dimensional

reference speaker embeddings obtained from enrollment data, each corresponding to one speaker k out of K. The feature sequence X is fed to the speaker encoder and the ASR encoder:

$$H^{\text{spk}} = \text{SpeakerEncoder}(X) \in \mathbb{R}^{T \times D}, \tag{2.6}$$

$$H^{\text{asr}} = \text{ASREncoder}(X) \in \mathbb{R}^{T \times D}. \tag{2.7}$$

where T and D are the embedding dimension and the length of the embedding sequence, respectively. The resulting embeddings H^{spk}, H^{asr} and the $n-1$ previous ASR output tokens $\hat{y}_{[1:n-1]}$ are given to the speaker decoder to obtain a speaker posterior $\hat{s}_n$ and a speaker profile $\bar{s}_n$ associated with the n-th output token as follows:

$$q_n = \text{SpeakerDecoder}(\hat{y}_{[1:n-1]}, H^{\text{asr}}, H^{\text{spk}}) \in \mathbb{R}^{E}, \tag{2.8}$$

$$\hat{s}_n = \text{Softmax}(S^{\intercal}\, q_n) \in \mathbb{R}^{K}, \tag{2.9}$$

$$\bar{s}_n = \sum S\, \hat{s}_n \in \mathbb{R}^{E}, \tag{2.10}$$

where $(\cdot)^{\intercal}$ denotes matrix transposition. The ASR embedding H^{asr} and the weighted speaker profile $\bar{s}_n$ are provided to the ASR decoder to generate the n-th ASR output token:

$$\hat{y}_n = \text{ASRDecoder}(\hat{y}_{[1:n-1]}, H^{\text{asr}}, \bar{s}_n). \tag{2.11}$$

Given the ground truth token and speaker sequences, the training objective is to maximize the joint probability

$$\begin{aligned} P(Y, S|X) = \prod_{n=1}^{N} & P(\hat{y}_n|\hat{y}_{[1:n-1]}, \hat{s}_{[1:n]}, X) \\ & \times P(\hat{s}_n|\hat{y}_{[1:n-1]}, \hat{s}_{[1:n-1]}, X). \end{aligned} \tag{2.12}$$

Subsequently, Kanda et al. (2021b) integrated SA-ASR in a Transformer architecture, enhancing the model's capabilities. In this design, the ASR Encoder adopts a modified Conformer network (Gulati et al., 2020), and the Speaker Encoder is a d-vector extractor (Zhou et al., 2019). The ASR Decoder, a Transformer-based Decoder, includes an additional layer to process the speaker profile $\bar{s}_n$. The ASR Decoder for layer l is outlined as follows:

$$z^{\text{asr}}_{[1:n-1],0} = \text{PosEnc}(\text{Embed}(y_{[1:n-1]})), \tag{2.13}$$

$$\bar{z}^{\text{asr}}_{n-1,l} = z^{\text{asr}}_{n-1,l-1} + \text{MHA}_l(z^{\text{asr}}_{n-1,l-1}, z^{\text{asr}}_{[1:n-1],l-1}, z^{\text{asr}}_{[1:n-1],l-1}), \tag{2.14}$$

$$\bar{\bar{z}}^{\text{asr}}_{n-1,l} = \bar{z}^{\text{asr}}_{n-1,l-1} + \text{MHA}_l(\bar{z}^{\text{asr}}_{n-1,l}, H^{\text{asr}}, H^{\text{asr}}), \tag{2.15}$$

$$z^{\text{asr}}_{[1:n-1],l+1} = \begin{cases} \bar{\bar{z}}^{\text{asr}}_{n-1,l} + \text{FF}_l(\bar{\bar{z}}^{\text{asr}}_{n-1,l} + W^{\text{spk}} \cdot \bar{s}_n) & (l = 1) \\ \bar{\bar{z}}^{\text{asr}}_{n-1,l} + \text{FF}_l(\bar{\bar{z}}^{\text{asr}}_{n-1,l}) & (l > 1) \end{cases}, \tag{2.16}$$

$$y_n = \text{Softmax}(W \cdot z^{\text{asr}}_{n-1,L} + b). \tag{2.17}$$

Here, Embed() and PosEnc() correspond to the embedding function and absolute positional encoding function, respectively. $\text{MHA}(Q, K, V)$ denotes the multi-head attention mechanism of the query Q, key K, and value V matrices. Additionally, FF() refers to the position-wise feed-forward network.

The Speaker Decoder takes the output of the first layer of the ASR Decoder $\bar{z}^{\text{asr}}_{n-1,1}$ and processes in the first layer of the Speaker Decoder as

$$\bar{\bar{z}}^{\text{spk}}_{n-1,1} = \bar{z}^{\text{asr}}_{n-1,1} + \text{MHA}_1(\bar{z}^{\text{asr}}_{n-1,1}, H^{\text{asr}}, H^{\text{spk}}), \tag{2.18}$$

$$\bar{\bar{z}}^{\text{spk}}_{n-1,2} = \bar{\bar{z}}^{\text{spk}}_{n-1,1} + \text{FF}_1(\bar{\bar{z}}^{\text{spk}}_{n-1,1}). \tag{2.19}$$

Then the next layers of the Speaker decoder compute

$$\bar{z}^{\text{spk}}_{n-1,l} = z^{\text{spk}}_{n-1,l-1} + \text{MHA}_l(z^{\text{spk}}_{n-1,l-1}, z^{\text{spk}}_{[1:n-1],l-1}, z^{\text{spk}}_{[1:n-1],l-1}), \tag{2.20}$$

$$\bar{\bar{z}}^{\text{spk}}_{n-1,l} = \bar{z}^{\text{spk}}_{n-1,l-1} + \text{MHA}_l(\bar{z}^{\text{spk}}_{n-1,l}, H^{\text{spk}}, H^{\text{spk}}), \tag{2.21}$$

$$\bar{\bar{z}}^{\text{spk}}_{n-1,l+1} = \bar{\bar{z}}^{\text{spk}}_{n-1,l} + \text{FF}_l(\bar{\bar{z}}^{\text{spk}}_{n-1,l}), \tag{2.22}$$

$$q_n = W \cdot z^{\text{spk}}_{n-1,L}. \tag{2.23}$$

Finally, q_n is used to calculate a speaker posterior $\hat{s}_n$ and a speaker profile $\bar{s}_n$ as in Equation (2.8).

In summary, the Transformer-based SA-ASR focuses atttention on the audio region corresponding to the current speaker by utilizing both transcription and speaker information. Thereafter, SA-ASR was extended to accommodate diverse scenarios (Kanda et al., 2021a), address an unlimited number of speakers (Kanda et al., 2022b), and cater to streaming applications (Kanda et al., 2022a). In a recent work (Li et al., 2023), SA-ASR was extended to a non-autoregressive mode by using Paraformer, an effective non-autoregressive model proposed by Gao et al. (2022) .

A variant involves the prediction of speaker-specific timestamps as an integral component of the ASR output. This innovation was initially introduced by Radford et al. (2023)

in their multitask model, Whisper, wherein one of the tasks involves generating a time-aligned transcription for the current audio segment, similar to *<|time0|> how are you doing <|time240|>*. Subsequently, Cornell et al. (2024) expanded upon this concept with the development of Sliding-Window Diarization-Augmented Recognition (SLIDAR). In SLIDAR, they integrated speaker identity for each segment through speaker embedding and clustering techniques applied to individual windows, resulting in an output akin to *<|spk0|><|time0|> how are you doing <|time240|>*.

Inspired by single-channel attention, Yu et al. (2023) developed a multichannel attention scheme called Multi-frame cross-channel attention (MFCCA). Cross-channel attention either concentrates on understanding global correlations between sequences across different channels or efficiently utilizes detailed channel-wise information at each time step. As shown in Figure 2.20, the h-th MFCCA head is computed as

$$Q_h = XW_h^q + (b_h^q)^\intercal \in \mathbb{R}^{T\times C\times D}, \tag{2.24}$$

$$K_h = X_{cc}W_h^k + (b_h^k)^\intercal \in \mathbb{R}^{T\times (2F+1)C\times D}, \tag{2.25}$$

$$V_h = X_{cc}W_h^v + (b_h^v)^\intercal \in \mathbb{R}^{T\times (2F+1)C\times D}, \tag{2.26}$$

$$H_h = \text{Softmax}\left(\frac{Q_h(K_h)^\intercal}{\sqrt{D}}\right)V_h \in \mathbb{R}^{T\times C\times D}, \tag{2.27}$$

where C represents the number of channels, W_h^q, W_h^k, b_h^q and b_h^k are learnable parameters, and $X_{cc} = [X_{cc}^0, \ldots, X_{cc}^t, \ldots, X_{cc}^T]$ with $X_{cc}^t = [X^{t-F}, \ldots, X^t, \ldots, X^{t+F}] \in \mathbb{R}^{(2F+1)C\times D}$ the concatenation of F context frames at each time step t.

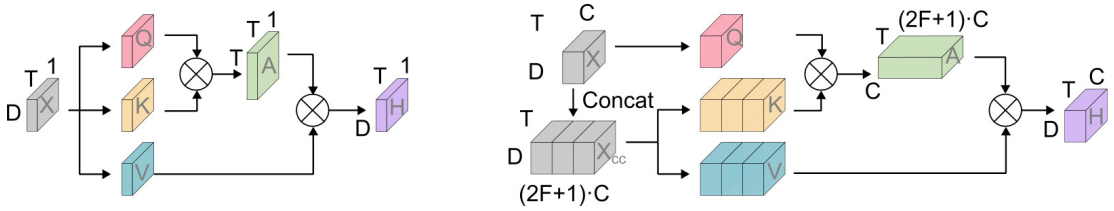


Figure 2.20: Single-channel attention (left) (Vaswani et al., 2017) and Multi-frame cross-channel attention (right) (Yu et al., 2023).

In a recent study, Shi et al. (2023b) introduced two pipelines for multichannel multi-speaker meeting transcription. As shown in Figure 2.21, in the first pipeline, referred to as MC-FD-SOT, an oracle VAD, multichannel frame-level diarization, and multichannel SOT are incorporated. The second pipeline, named MC-WD-SOT, involves multichannel word-level diarization, while the remaining modules are identical to those in MC-FD-SOT.

Similar to SA-ASR, they also need reference speaker embeddings as inputs to speaker diarization.

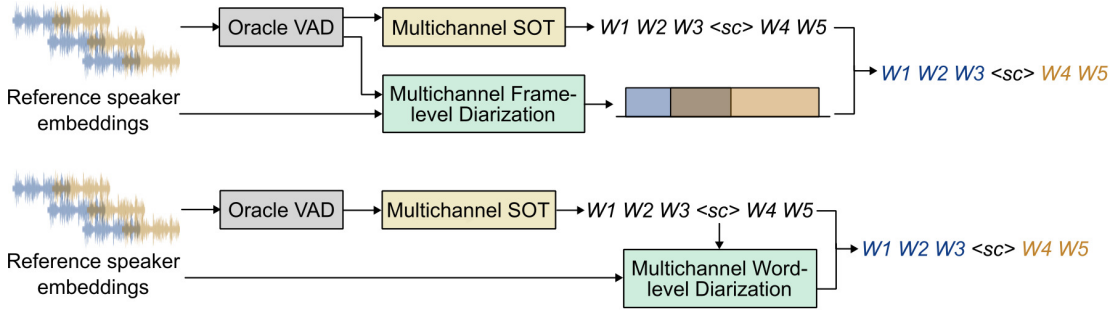


Figure 2.21: Multichannel Frame-level diarization with Multichannel SOT (MC-FD-SOT) pipeline (top) and Multichannel Word-level diarization with Multichannel SOT (MC-WD-SOT) pipeline (down) (Shi et al., 2023b).

The two pipelines utilize MFCCA for multichannel SOT. For MC-FD-SOT, they employ target speaker activity detection (Medennikov et al., 2020; He et al., 2021) to obtain the frame-level diarization result. These timestamps are then aligned with the output of SOT. For MC-WD-SOT, an attention-based architecture (Yu et al., 2022a) similar to Equation (2.20) is used to predict speaker identity. The difference is that the speaker representation is not used in the ASR decoder, as shown in Equation (2.13).

In the realm of end-to-end and joint optimization systems, existing methods include end-to-end separation and diarization, joint optimization of separation and ASR, and finally end-to-end integration of ASR and diarization. The first two approaches necessitate additional modules, such as ASR and/or diarization, to complete the comprehensive meeting transcription pipeline. Current research predominantly concentrates on end-to-end ASR and diarization, due to the absence of a separation module. This is emphasized by the smooth merging of ASR and diarization tasks, as demonstrated in studies like those carried out by researchers such as Kanda et al. (2021b), Cornell et al. (2024), and Shi et al. (2023b).

2.3 Summary

This chapter provided an overview of existing methods for automatically generating meeting transcriptions. We began by exploring the fundamental modules involved in this task, namely speech separation, speaker diarization, and ASR. Next, we delved into two major categories of methods: pipeline methods and end-to-end methods for meeting transcription. Most methods incorporate a speech separation module to enhance speech quality

and facilitate subsequent ASR and diarization tasks. However, end-to-end approaches that integrate only speaker diarization and ASR offer advantages such as a smaller system size and increased robustness to varying signal conditions. The following chapters will primarily focus on these two principles.

3 Joint speech separation, ASR and speaker identification

分事务以达其功。
Handle affairs separately to achieve success.

On far-field meeting transcription, several studies have proposed to separate individual speech signals first and then to employ a single-speaker ASR module (Yoshioka et al., 2018b; Chen et al., 2019; Kanda et al., 2019a). Choosing the right separation model is essential for the success of this approach. A multichannel separation architecture is beneficial because it exploits multichannel information. The FaSNet (Luo et al., 2019) architecture meets this need through a two-stage process that generates time-domain filters for all channels. The TAC (Luo et al., 2020a) scheme further improves performance, enhancing the FaSNet framework.

In addition, training a separation model such as FaSNet on real-life datasets is challenging due to the lack of ground truth signals. Some methods like mixture invariant training (Wisdom et al., 2020; Sivaraman et al., 2022) propose unsupervised training for single-channel mixtures, while other methods (Drude et al., 2019; Jiang et al., 2021; Saijo and Scheibler, 2022) offer solutions for multichannel settings, which rely on unsupervised training based on proposed loss functions or metrics. However, few studies have attempted to find solutions based on the real data itself. Notably, supervised learning using high-quality ground truth sources is more reliable and efficient (Love, 2002; Wang and Chen, 2018). Furthermore, although some studies (Chen et al., 2020; Taherian and Wang, 2021) have evaluated performance as a function of the speaker overlap ratio in the test set, determining the optimal overlap ratio in the training set when testing on real-life meeting data remains a challenge.

This chapter first introduces a pipeline composed of a multichannel separation model, an ASR model, and a speaker identification model. To the best of our knowledge, we are

the first to jointly fine-tune ASR and speaker embedding models in a PIT manner. Second, we propose data generation and augmentation methods for training the separation model on real meeting data. Then, we evaluate the influence of the overlap ratio for training the separation model and its impact on ASR and speaker identification fine-tuning on both synthetic and real corpora.

This chapter is organized as follows. Section 3.1 presents our pipeline system and the method for generating training data for separation. Section 3.2 presents our experimental settings, followed by the discussion of results in Section 3.3. Finally, Section 3.4 provides a summary of this chapter.

3.1 Proposed methods

3.1.1 System architecture

We propose a design similar to the CSS pipeline and the one suggested by Raj et al. (2021). Since the multichannel separation model operates in an end-to-end fashion, we do not need a specific dereverberation module before CSS, unlike in Figure 2.13. Additionally, a VAD module is unnecessary because the separated signals contain only speech, which will be directly passed to the ASR and speaker embedding models, unlike in Figure 2.14. Moreover, the diarization step in the pipeline by Raj et al. (2021) (see Figure 2.15) is no longer necessary. We choose to embed speaker information at the sentence level, leveraging more comprehensive information to perform speaker identification effectively.

As illustrated in Figure 3.1, the proposed pipeline consists of three modules: speaker separation, automatic speech recognition (ASR), and speaker identification. The inputs consist of multichannel, multi-speaker speech and reference speaker embeddings. The separation model processes this input to produce single-channel, single-speaker audio. This improves speech quality by reducing overlapping speech, noise, and reverberation, which enhances overall speech quality and facilitates subsequent ASR and speaker diarization tasks. The separation model is followed by a single-channel, single-speaker ASR and a speaker embedding model in parallel. The diarization task in this pipeline is simplified to speaker identification. As highlighted in Section 2.1.2, speaker recognition based on speaker embeddings demonstrates stability and versatility across various scenarios. We use a method based on a speaker embedding model due to its efficiency on non-overlapped speech. Instead of clustering all speaker embeddings, we assume the availability of reference speaker embeddings and identify speakers by performing a dot product operation between the extracted embeddings and the reference embeddings. The outputs of the system are the transcripts

and the speaker identities of all the speakers present in the mixture.

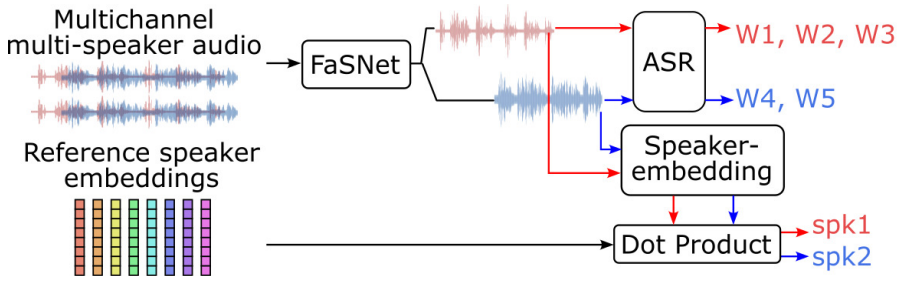


Figure 3.1: Proposed joint FaSNet, ASR and speaker identification system.

Multichannel speech separation using FaSNet

To perform multichannel speech separation, we employ the FaSNet system, which directly estimates time-domain filters. To address channel permutation and enable the model to handle varying numbers of microphones, we use a TAC design paradigm. Detailed explanations of FaSNet and TAC are provided in Section 2.1.1.3. The neural FaSNet-TAC separation model not only separates the speech sources but also improves the speech quality by reducing the noise and reverberation present in the mixture.

Conformer-Transformer ASR

Despite the high quality of FaSNet-TAC, separated speech signals can still contain interfering speech, and separation errors can propagate to the ASR model. To partly address this challenge, it's important to choose a powerful ASR system. As discussed in Section 2.1.3.2, attention-based Encoder-Decoder models represent the state-of-the-art architecture for offline ASR. Therefore, we propose to employ an ASR model with a Conformer-based Encoder (Gulati et al., 2020) and a Transformer-based Decoder (Karita et al., 2019).

Speaker embedding using ECAPA-TDNN

To encode speaker information, we use x-vectors generated by the ECAPA-TDNN model (Desplanques et al., 2020). As discussed in Section 2.1.2.1, ECAPA-TDNN employs Res2Net and Squeeze-and-Excitation blocks for enhanced channel interdependencies and temporal context expansion. The resulting embeddings are compared with reference speaker embeddings to predict the speaker ID.

3.1.2 Data alignment for training the FaSNet model

Speech separation on real-world far-field data presents two challenges. First, the amount of real meeting data is often insufficient to train a powerful separation model. Second, there is a lack of ground truth enhanced sources to be used as training targets. Some real meeting corpora, such as AMI (Carletta et al., 2005) and ICSI (Janin et al., 2003), include headset recordings for each speaker, which cannot be directly used as ground truth due to the delays between these recordings and those captured by each multiple distant microphone (MDM), which vary over time due to speaker movements.

To tackle these two challenges, we propose to generate clips of overlapping array signals and the corresponding aligned headset signals. The process is divided into three steps (see Figure 3.2): for each meeting, (a) based on data annotation, we extract all non-overlapping speech segments for each speaker; (b) we employ matched filters to align each non-overlapping headset segment with the corresponding array segment, and cut them into fixed-length clips; (c) we randomly sample array clips from different speakers, sum them to create far-field mixtures, and take the corresponding aligned headset clips as the corresponding ground-truth enhanced sources.

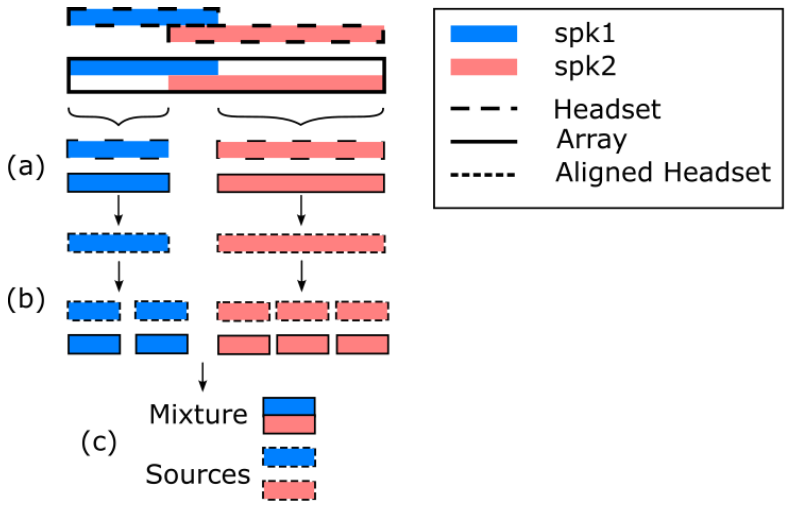


Figure 3.2: Generation of ground truth source signals for real meeting data.

The matched filters $f_{ij}(t)$ in step (b) are calculated in the least squares sense by solving

$$\min_{f_{ij}} \sum_{t} (f_{ij} \star h_j(t) - x_i(t))^2 \tag{3.1}$$

where $h_j(t)$ and $x_i(t)$ stand for the headset signal of speaker j and the array signal at micro-

phone i, respectively, and $\star$ denotes time-domain convolution. The solution is classically obtained as the finite impulse response (FIR) Wiener filter,[1] which is commonly employed for filter estimation (Vincent et al., 2006; Le Roux et al., 2019).

3.2 Experiments

3.2.1 Datasets

We validate the effectiveness of the proposed pipeline on the AMI meeting corpus (Carletta et al., 2005). The AMI corpus includes multiple distant microphone (MDM) and individual headset microphone (IHM) data. We refer to the first channel of the MDM data as single distant microphone (SDM). The MDM data consists of approximately 100 hours of 8- to 16-channel audio recordings of 3 to 5 participants in meetings. The data is annotated in terms of ASR and diarization, with the start and end timestamps for each sentence.

Mixed AMI for FaSNet training

To train the FaSNet model, we apply the method described in Section 3.1.2 to the AMI 2-channel MDM and IHM meeting data. This method creates mixtures of real single-speaker AMI segments and their corresponding ground truths. We name this dataset Mixed AMI. We only use one-quarter[2] of all meetings and fix the clip length to 4 s. The training, development and test sets contain 150 h, 17 h, and 16 h of speech, respectively.

Real AMI for fine-tuning of ASR and speaker embedding models

After separating the real AMI MDM data by FaSNet, the ASR and speaker embedding models are fine-tuned and evaluated on the resulting enhanced data. For fine-tuning, the order of the separated speech needs to be known, or a Permutation Invariant Training (PIT) method is necessary. Without prior knowledge of the speaker order, using PIT for the ASR model following a separation model is a common approach, as several studies have shown (Kolbæk et al., 2017; Yu et al., 2017b; Yoshioka et al., 2018a). To the best of our knowledge, we are the first to fine-tune ASR and speaker embedding models together using a PIT method.

In order to process the meeting files, which typically consist of approximately 1 h of content, we have to divide each meeting into smaller segments. We adopt a segmentation approach inspired by "utterance groups" (Kanda et al., 2021c) which works as follows. (a)

[1] https://en.wikipedia.org/wiki/Wiener_filter

[2] All meetings have been initially segmented into a, b, c and d audio segments by corpus providers, we only use the a partition of all meetings.

Segment each meeting using a chunk size of b seconds and a hop size of o seconds. (b) If the start/end time of a segment falls within a region involving more than one speaker, it is adjusted to be 2 s outside the overlap region. (c) If the start/end time of a segment falls within a word, it is adjusted to align with the start/end of that word. Segmenting in this manner can increase the number of training samples. Additionally, it allows for easy control over the overlap ratio.

We conducted experiments with a chunk size of 5 s, and a hope size of 2 s. Table 3.1 presents some statistics of the training, development, and test sets generated with this segmentation method. We then divided them into four different sets based on the number of speakers, which include datasets with 1, 2, 3 and 4 speakers, respectively. The number of segments for each speaker count is as follows: 5,737 segments with 1 speaker, 4,911 segments with 2 speakers, 3,986 segments with 3 speakers, and 1,936 segments with 4 speakers.

Table 3.1: AMI statistics after segmentation (chunk size of 5 s). The average duration is in seconds (s), and the total duration is in hours (h).

	Training set					
# speakers	1	2	3	4	5	total
# segments	34,719	30,304	18,492	6,900	10	9,0425
Avg. dur. (s)	5.62	6.48	7.74	8.93	9.43	6.59
Total dur. (h)	54.22	54.57	39.77	17.11	0.02	165.71
# words	444,656	592,651	520,644	252,017	434	1,810,402
	Development set					
# speakers	1	2	3	4		total
# segments	4,104	3,357	2,362	988		10,811
Avg. dur. (s)	5.55	6.38	7.45	8.45		6.49
Total dur. (h)	6.33	5.95	4.88	2.31		19.50
# words	52,730	65,072	65,575	35,803		219,180
	Test set					
# speakers	1	2	3	4		total
# segments	4,211	3,035	2,129	858		10,233
Avg. dur. (s)	5.57	6.44	7.90	9.34		6.63
Total dur. (h)	6.52	5.43	4.67	2.22		18.86
# words	52,765	59,624	64,769	33,500		210,658

Synthetic AMI for fine-tuning of ASR and speaker embedding models

To facilitate the study of varying overlap ratios for speech separation, we chose to create synthetic mixed utterances for fine-tuning ASR and speaker embedding models. This approach allows us to reorder the segmented sources by computing the separation loss with

ground truth single-speaker segments and avoid PIT during downstream training. Specifically, we create 2-speaker overlapped utterances by randomly summing 1-speaker segments from real AMI data. We refer to this corpus as synthetic AMI.

Statistics of overlap ratios for real AMI

Since multi-speaker utterances in real-life applications are not entirely overlapped, we experiment with different overlap ratios when creating the mixed AMI data. Before determining the overlap ratio for data mixing, we first studied the overlap ratios in real-life scenarios. Using data annotation, we calculated the overlap ratios for the entire AMI corpus. We divided the calculations into two different scenarios: the raw meetings without segmentation and the segmented corpus (real AMI). In the first case, we computed the overlap ratio both with and without considering silence. For the real AMI data, we only calculated the overlap ratio for 2-speaker overlaps because we conducted varying overlap ratio training on mixed AMI only for the 2-speaker case. Specifically, we took all the segments with exactly 2 speakers and calculated the overlap ratio within this subset. Figure 3.3 illustrates the calculation methods for these scenarios.

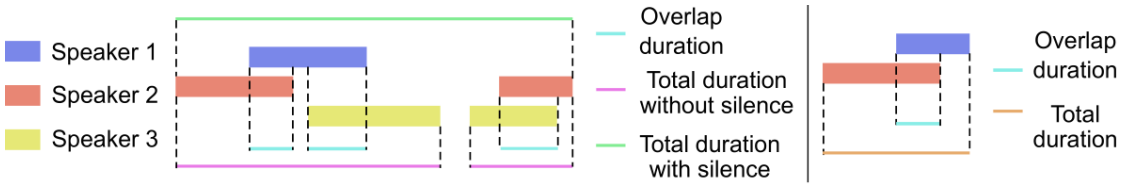


Figure 3.3: 2-speaker overlap ratio calculation for AMI raw meetings (left) and for segmented AMI data (right).

Table 3.2 presents the results. In real-life scenarios, speaker overlap is rare: 2-speaker overlap occurs in only 12.1% of the entire recorded meetings and 14.5% of all the speech regions. In our real AMI subset, the overlap ratio for 2-speaker segments is 17.9%. Based on these findings, we selected overlap ratios of 0%, 25%, 50%, and 75% for the 2-speaker mixed AMI.

Table 3.2: Statistics of overlap ratio (%) in AMI raw meetings.

# Speakers	0	1	2	3	4	5
With silence	16.6	68.6	12.1	2.3	0.5	0.004
Without silence	0	82.2	14.5	2.7	0.6	0.004

3.2.2 Model description

The implementation of FaSNet-TAC is taken from the Asteroid toolkit (Pariente et al., 2020). The frame size and context size of FaSNet are set to 4 and 16 ms, respectively. The encoder dimension and feature dimension are 64. The dual path blocks consist of a 4-layer dual model. The number of parameters of FaSNet is only 2M. For the subsequent ASR model, we use a Conformer-based single-channel ASR model pretrained [3] on the LibriSpeech dataset (Panayotov et al., 2015), implemented using the SpeechBrain toolkit (Ravanelli et al., 2021). The Conformer-based encoder and the Transformer-based decoder have 12 and 6 layers, respectively. All multi-head attention mechanisms have 8 heads, the model dimension is 512, and the size of the feedforward layer is 2,048. The ASR model has 132M parameters.

The speaker embedding model is an ECAPA-TDNN (Emphasized Channel Attention, Propagation, and Aggregation in Time-Delay Neural Network) (Desplanques et al., 2020) pre-trained[4] on the VoxCeleb (Chung et al., 2019; Nagrani et al., 2020) corpora, yielding 192 dimensional embeddings. The ECAPA-TDNN model has 21M parameters.

3.2.3 Training setup

The FaSNet model is trained on Mixed AMI for 200 epochs using the Adam optimizer with a learning rate of 10^{-3} with early stopping. Using Adam with a learning rate of 3×10^{-4}, we fine-tune the ASR model on Mixed AMI for 15 epochs and fine-tune the speaker embedding model for an additional 5 epochs. When fine-tuning on the combined Mixed AMI and real AMI, we fine-tune the ASR model for the first 7 epochs and fine-tune the speaker embedding model for the last 3 epochs.

3.2.4 Metrics

We evaluate the separation performance using the scale-invariant signal-to-distortion ratio (SI-SDR) and its improvement (SI-SDRi) in dB on the Mixed AMI test set. For ASR performance, we utilize the word error rate (WER). In the context of speaker identification in meeting scenarios, the sentence-level speaker error rate (SER) is commonly employed to measure performance (Kanda et al., 2020a, 2021b). In our case, we embed each separated source using ECAPA-TDNN and assign a speaker ID by computing the dot product between

[3] Available at https://huggingface.co/speechbrain/asr-conformer-transformerlm-librispeech

[4] Available at https://huggingface.co/speechbrain/spkrec-ecapa-voxceleb

this embedding and the reference speaker embeddings. The SER is calculated as

$$\text{SER} = \frac{E}{A} \tag{3.2}$$

where E represents the number of utterances with incorrect predictions of the speaker, and A is the total number of utterances.

3.3 Evaluation results

3.3.1 Results for different overlap ratios on synthetic AMI

We first trained and tested our system on mixed and synthetic AMI data. To examine the influence of the overlap ratio, we trained four separation models corresponding to four different overlap ratios: 0%, 25%, 50%, and 75%. We did not include a 100% overlap ratio, as completely overlapped speech is unrealistic in real-life meeting scenarios. For each separation model, we then fine-tuned the ASR and speaker identification modules on separated speech mixtures with overlap ratios equal to or lower than the one used for training the separation model. We included mismatched overlap ratios to assess the robustness of the separation models to such mismatch. We tested ratios less than those used in separation training because separating more overlapped data during inference is more challenging. Given that we have access to different overlap ratios for mixed AMI, this difficulty can be mitigated.

Table 3.3 shows the results on the test set. First, increasing the overlap rate from 0% to 75% results in a reduction of 3 dB in SI-SDRi (from 12.61 dB to 9.61 dB), and 88% and 61% relative degradation in WER (from 28.31% to 53.42%) and SER (from 6.65% to 10.73%), respectively. This considerable degradation demonstrates that the performance is highly sensitive to the overlap ratio. Specifically, the relative increases in WER are 18% (from 28.31% to 33.61%), 24% (from 33.61% to 41.69%), and 28% (from 41.69% to 53.42%) at each increment of 25% in overlap ratio. This demonstrates that the degradation speed in ASR is quicker than the increase in overlap ratio, highlighting the sensitivity of recognition performance to overlap.

In terms of the robustness to mismatched overlap ratio, better results are generally obtained when the overlap ratio used to fine-tune the ASR and speaker embedding models matches the overlap ratio used to train FaSNet. More specifically, fine-tuning and testing on 50% overlapped data using a separation model trained on 50% overlapped data results in an 11% relative lower WER (from 46.88% to 41.69%) and a 4% relative lower SER (from

Table 3.3: SI-SDR (dB), SI-SDRi (dB), WER (%), and SER (%) achieved by the FaSNet pipeline on synthetic 2-channel 2-speaker mixed AMI and synthetic AMI data as a function of the speaker overlap ratio in the FaSNet training set and the ASR and speaker embedding fine-tuning set.

Overlap ratio in training set		**Mixed AMI test set**			**Synthetic AMI test set**	
FaSNet	**ASR + speaker**	**Overlap**	**SI-SDR ↑**	**SI-SDRi ↑**	**WER ↓**	**SER ↓**
0%	0%	0%	**10.07**	**12.61**	28.31	6.65
25%	0%	0%	9.99	12.53	**26.89**	**6.60**
	25%	25%	8.92	11.46	33.61	8.05
50%	0%	0%	9.90	12.45	27.49	7.74
	25%	25%	9.01	11.56	33.42	8.00
	50%	50%	8.26	10.80	41.69	10.05
75%	0%	0%	8.71	11.25	32.64	10.33
	25%	25%	8.17	10.72	37.85	10.24
	50%	50%	7.64	10.18	46.88	10.47
	75%	75%	7.06	9.61	53.42	10.73

10.47% to 10.05%) compared to using a separation model trained on 75% overlapped data. This demonstrates that, without augmented datasets for training, matched training and testing conditions lead to better performance.

3.3.2 Results on real AMI using PIT

We then conducted experiments on 2-speaker real AMI data to evaluate the performance of our system in real-life scenarios. Initially, we fine-tuned the system only on real AMI data using a separation model trained on 50% overlapped data, achieving a WER of 41.67% and an SER of 12.67%. To further improve the robustness of the ASR and speaker identification modules, we used both synthetic AMI and real AMI data as the training set, and tested on real AMI data. The measured overlap ratio in 2-speaker scenarios on real AMI data is 21.03% in the training and development sets, and 17.09% in the test set, so we used 25% overlapped synthetic AMI data for fine-tuning. We compared different separation models trained on various overlap ratios.

Table 3.4 shows the results on real AMI data. First, adding synthetic AMI data to the training set improves performance with an 18% relative reduction in WER (from 41.67% to 34.32%) and a 17% relative reduction in SER (from 12.67% to 10.65%). This demonstrates the importance of multi-condition training for enhancing model robustness. Additionally, Table 3.3 shows that using a matched separation model can lead to better performance at a specific overlap ratio, but Table 3.4 reveals a different trend. Given that the overlap

ratio in fine-tuning ranges from 21.03% to 25%, using a separation model trained on 50% or 75% overlapped data yields better results compared to using 0% or 25% overlapped data. Specifically, the model trained on 50% overlapped data achieves a 5% relative reduction in WER (from 36.24% to 34.42%) compared to the model trained on 25% overlapped data. This indicates that real AMI data, being more complex, benefits from a separation model trained under more challenging conditions to improve robustness in real-life scenarios.

Table 3.4: SI-SDR (dB), SI-SDRi (dB), WER (%), and SER (%) achieved by the FaSNet pipeline on 2-channel 2-speaker real AMI data when trained on synthetic AMI and real AMI data using PIT as a function of the speaker overlap ratio in the FaSNet training set.

Overlap ratio in the FaSNet training set	Mixed AMI test set			Real AMI test set		
	overlap	SI-SDR	SI-SDRi	overlap	WER	SER
0%	25%	7.68	10.22	21.03%	36.84	10.87
25%	25%	8.92	11.46	21.03%	36.24	**9.23**
50%	25%	**9.01**	**11.56**	21.03%	**34.32**	10.65
75%	25%	8.17	10.72	21.03%	35.46	10.23

Note: We employed the SCTK toolkit (NIST, 2024) to conduct significance tests, specifically the Matched Pair Sentence Segment test. We highlight in bold the best WER/SER result and the results statistically equivalent to it at a 0.05 significance level.

3.3.3 Comparative analysis of FaSNet on mixed AMI and real AMI

We propose to train FaSNet on mixed AMI to perform separation on real AMI. To analyze FaSNet's adaptation capability, we compare the separation performance on mixed AMI and real AMI by visualizing the spectrograms of separated speech. Figure 3.4 illustrates the separation performance on mixed AMI. We observe that despite overlapping speech regions for two speakers, FaSNet effectively separates these regions. The speaker boundary lines are accurately predicted compared to the ground truth, and there is no interference speech in both speakers' estimated outputs. This demonstrates that FaSNet efficiently separates speech in the test set of the same type as the training set.

Figure 3.5 shows the separation output on real AMI data. By listening to the mixture, we can discern that the first speaker's speech ends before the two-second mark, and the second speaker starts speaking from the two-second mark. There is no obvious overlap region between the two speakers. However, the estimated speech for the first speaker contains residual speech from the second speaker starting at 2 seconds. By listening to the audios, we can tell that the second speaker is the dominant one in the mixture, speaking for

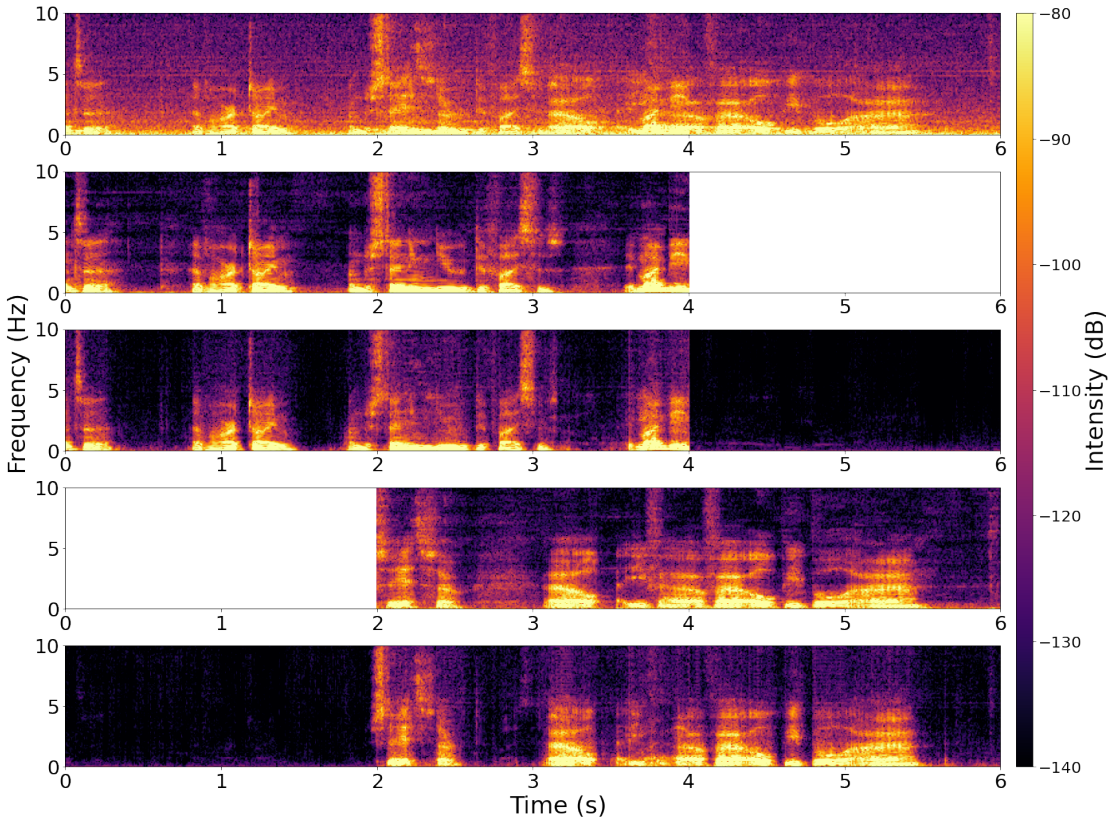


Figure 3.4: Spectrogram of 2-channel 2-speaker separation on a mixed AMI test chunk. From top to bottom: 1st channel of mixture; ground truth of speaker 1; estimated speaker 1; ground truth of speaker 2; estimated speaker 2.

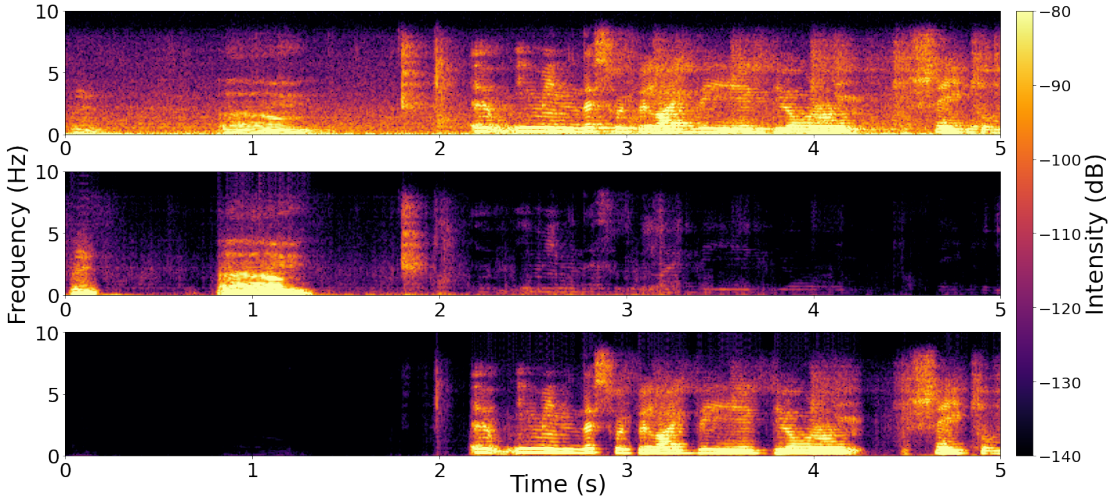


Figure 3.5: Spectrogram of 2-channel 2-speaker separation on a real AMI test chunk. From top to bottom: 1st channel of mixture; estimated speaker 1; estimated speaker 2.

a longer duration and at a higher volume. Consequently, the estimated speech for the second speaker is of quite high quality. This demonstrates a shortcoming of training on mixed AMI, where the two speakers are balanced in the mixture, and then testing on real AMI, where the mixture is more random. It is common for one speaker to be dominant during the mixture in real scenarios due to meeting dynamics or the speakers' positions relative to the microphones. This highlights the need to train the separation model on a variety of mixtures to improve the model's robustness.

Finally, we observed that the mixture from mixed AMI contains more ambient noise compared to the real AMI mixture. This is because the mixture generation involved summing MDMs from two different speakers, resulting in doubled noise. Training on this noisier data improves the model's robustness to ambient noise.

3.4 Summary

This chapter introduced a transcription pipeline that includes multichannel separation, single-channel single-speaker ASR, and speaker identification models. Experimental results demonstrate that speaker overlap ratio in the training data for the separation model and the fine-tuning data for the ASR and speaker embedding models can significantly influence the system's performance. Specifically, on synthetic AMI data, varying the overlap ratio from 0% to 75% results in an 88% relative increase in WER and a 61% relative increase in SER. Results on real AMI data reveal that increasing the variety of training data can help improve model robustness. More specifically, training on both synthetic and real AMI data can reduce the WER by 18% relative compared to training only on real AMI data. Results on real AMI data also demonstrate that separation models trained on data with higher speaker overlap are more robust in real-life conditions. Finally, FaSNet performs better on the mixed AMI test set than on the real AMI test set, primarily due to the imbalance of speaker segments in the real AMI mixtures. This finding demonstrates the necessity of multi-condition training to increase the model's robustness.

4 End-to-end multichannel speaker-attributed ASR

欲穷千里目，更上一层楼。
To see a thousand miles further, ascend one more story higher.

Chinese proverb

Chapter 3 presents a pipeline using separation model as a front-end processing. An issue with this approach is that the separation errors may propagate to the ASR module. Later studies (Chang et al., 2019; Wu et al., 2021; Shi et al., 2022) have proposed to back-propagate the ASR training losses for all speakers to the front-end separation module using a PIT criterion to optimize the two modules jointly. However, the system can often handle only a fixed number of speakers.

To address this, end-to-end multi-speaker ASR and diarization systems able to handle a variable number of speakers have been developed, and have demonstrated promising results for single-channel meeting transcription (Guo et al., 2021; Lu et al., 2021; Sklyar et al., 2021; Kanda et al., 2021b). For instance, the Transformer-based SA-ASR system of Kanda et al. (2021b) achieves joint multi-speaker ASR and diarization by computing speech and speaker representations and sharing them across these two tasks.

For multichannel settings, multiple approaches have been proposed that exploit spatial information for improved ASR and diarization (Chang et al., 2019, 2020; Scheibler et al., 2023). For instance, Yu et al. (2023) proposed to use MFCCA in the ASR Encoder to fuse the input channels. However, the ASR and the diarization modules in this approach use different types of attention for multichannel fusion, leading to an increase in the number of model parameters. Moreover, in the architecture proposed by Yu et al. (2023), the communication between ASR and speaker information is not bidirectional, as the speaker information is not reciprocally exploited by the ASR. Besides, multichannel SA-ASR (MC-SA-ASR) can exploit spatial information which is generally advantageous for localizing dif-

ferent speakers. However, the predominant approaches in end-to-end multichannel ASR use Mel filterbank features as input and discard phase information (Yu et al., 2023; Zhao et al., 2021). More recently, the "all-in-one" model of Shao et al. (2022) uses interchannel phase difference (IPD) and position features, wherein the later require video information to localize the speakers. Chang et al. (2021) pass magnitude and phase information from each channel were separately passed through linear layers and concatenate them to form the input for multichannel ASR. To our knowledge, there is currently no research comparing and discussing the impact of different input features on MC-SA-ASR.

In this chapter, we target end-to-end MC-SA-ASR, and investigate the benefit of i) leveraging speaker information for ASR and ii) exploiting phase information. To do so, we propose an MC-SA-ASR system that tightly couples a Conformer-based encoder with MFCCA, a speaker encoder and a speaker-attributed Transformer-based decoder. Unlike Yu et al. (2023), our model features a single ASR encoder, from which the same ASR embedding is passed to both the ASR decoder and the speaker decoder. We explore the use of phase information as input, comparing it to Mel filterbank features. Moreover, we evaluate the effectiveness of using data-invariant beamformed signals as input. We conduct extensive experiments on simulated mixtures of LibriSpeech data as well as on the AMI corpus.

The structure of the chapter is as follows. In Section 4.1, we introduce our proposed MC-SA-ASR system and the different features and feature encoding methods. Section 4.2 presents our experimental setup, followed by Section 4.3 presenting comparative results of using different input features. Finally, Section 4.4 provides a summary. The primary contents of this chapter have been published at ASRU 2023 (Cui et al., 2023).

4.1 Proposed methods

4.1.1 Model architecture

The original Transformer-based end-to-end single-channel SA-ASR (SC-SA-ASR) of Kanda et al. (2021b) includes an ASR block and a speaker block (see Figure 2.19). We extend it into a multichannel architecture by replacing the original ASR encoder by a multichannel ASR encoder. Our proposed multichannel SA-ASR (MC-SA-ASR) system is illustrated in Figure 4.1. The inputs to the model consist of a multichannel acoustic feature sequence $X \in \mathbb{R}^{L \times C \times A}$ where L is the sequence length, C is the number of channels, and A the feature dimension, along with a matrix of reference speaker embeddings obtained from enrollment data. The multichannel ASR encoder implements MFCCA and performs channel-wise

encoding. The details of MFCCA can be found in Equation (2.24). The ASR encoder for layer l is outlined as follows:

$$\hat{x}_l^{\text{asr}} = x_l^{\text{asr}} + \text{MFCCA}_l(x_l^{\text{asr}}) \in \mathbb{R}^{T\times C\times D}, \tag{4.1}$$
$$\bar{x}_l^{\text{asr}} = \hat{x}_l^{\text{asr}} + \text{MHSA}_l(\hat{x}_l^{\text{asr}}) \in \mathbb{R}^{T\times C\times D}, \tag{4.2}$$
$$\bar{\bar{x}}_l^{\text{asr}} = \bar{x}^{\text{asr}} + \text{Conv}_l(\bar{x}^{\text{asr}}) \in \mathbb{R}^{T\times C\times D}, \tag{4.3}$$
$$\bar{\bar{\bar{x}}}_l^{\text{asr}} = \bar{\bar{x}}_l^{\text{asr}} + \text{FF}_l(\bar{\bar{x}}_l^{\text{asr}}) \in \mathbb{R}^{T\times C\times D}, \tag{4.4}$$
$$H^{\text{asr}} = \bar{\bar{\bar{x}}}^{\text{asr}} + \text{ConvFusion}_l(\bar{\bar{\bar{x}}}^{\text{asr}}) \in \mathbb{R}^{T\times D}. \tag{4.5}$$

Here, $\text{MHSA}(X)$ denotes the time-wise multi-head self attention mechanism. $\text{Conv}(X)$ is a time-wise convolution layer, $\text{FF}(X)$ is a time-wise feed-forward layer, and $\text{ConvFusion}(X)$ is a convolution fusion layer that fuses multichannel features into single-channel.

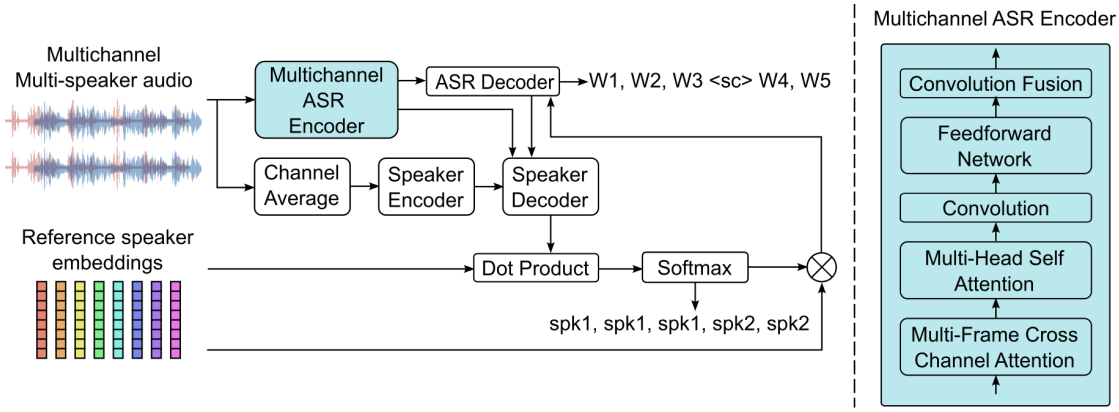


Figure 4.1: Overview of the proposed MC-SA-ASR system (left) and its encoder (right).

The rest of the MC-SA-ASR performs single-channel processing in the same way as SC-SA-ASR system. Details can be found in Equations 2.6 to 2.20. The output of MC-SA-ASR is the concatenation of all speakers' sentences, where each token is associated with one speaker ID and distinct speakers are separated by <sc> token. The next subsections describe the elements that are specific to the proposed system.

Convolution fusion

Convolution fusion serves as the output layer of the multichannel ASR encoder (see Figure 4.1). It combines the representations corresponding to the multiple input channels. The study by Yu et al. (2023) presents the mechanism for 8-channel fusion. We extend it to support 2, 3, and 4 channel input as illustrated in Figure 4.2.

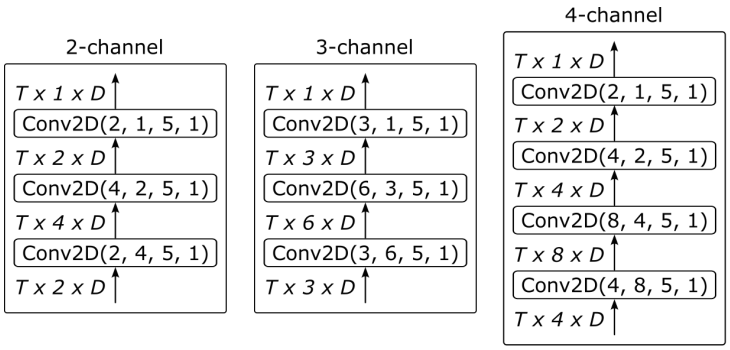


Figure 4.2: Multichannel convolution fusion extended to 2, 3 and 4 input channels.

The design where CNNs first increase the channel dimension and then reduce it has several advantages. By increasing the channel dimension, the model can capture more complex and diverse features from the input data, providing more capacity for the network to apply non-linear transformations. After enriching and integrating the features, reducing the channel dimension helps in condensing the information into a more compact and efficient representation. A similar design can be found in ResNet (He et al., 2016), EfficientNet (Duong et al., 2020), and MobileNetV2 (Srinivasu et al., 2021), among others.

Speaker encoder

As interchannel differences cannot directly contribute to improving the representativity of speaker characteristics, our speaker encoder follows a single-channel design. To address this, Mel filterbank features are first averaged across all channels and then fed to our speaker embedding model. We use an x-vector speaker embedding model based on ECAPA-TDNN (Desplanques et al., 2020). To align the dimension of the x-vectors with our model architecture, we replace the final (average pooling) layer with a linear layer that converts the dimension of the speaker embedding to the dimension of the model. The two layers of the speaker encoder are as follows:

$$\hat{x}^{\text{spk}} = \text{ECAPA-TDNN}(x^{\text{spk}}) \in \mathbb{R}^{T \times E}, \tag{4.6}$$

$$H^{\text{spk}} = \text{Linear}(\hat{x}^{\text{spk}}) \in \mathbb{R}^{T \times D}. \tag{4.7}$$

Here, ECAPA-TDNN(X) denotes the speaker embedding function and E denotes the embedding dimension. Linear(X) is a linear layer.

4.1.2 Input features

We consider two alternative sets of input features. On the one hand, we compute M-dimensional log Mel filterbank features from the STFT magnitude only. On the other hand, we concatenate the STFT magnitude and the cosine and sine of the phase, each with dimension G, into a $3 \times G$-dimensional representation, that is called the magnitude+phase feature. These features then undergo a specific processing, which is illustrated in Figure 4.3 and described hereafter.

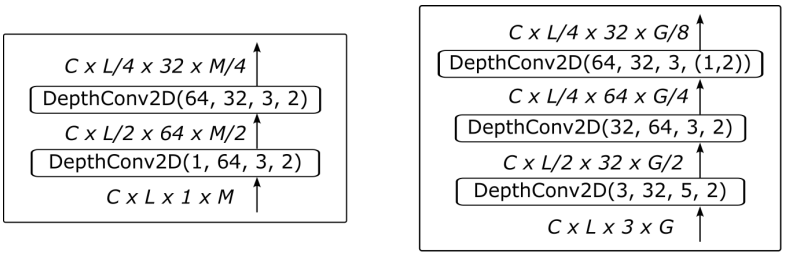


Figure 4.3: Depth-wise 2D-convolution feature extraction (input dimension, output dimension, kernel size, stride) for Mel filterbank (left) and magnitude+phase (right) features.

For the Mel filterbank, we apply depthwise separable convolution on each microphone. In the input array, C represents the number of microphones and L is the audio length. After two layers of 2-dimensional convolution, the output has a dimension of $C \times T \times A$, where $T = L/4$ since each layer performs a sub-sampling of factor 2 over the time dimension, and $A = 32 \times M/4$, which is the output feature dimension.

The magnitude+phase features are processed using three layers of depthwise separable convolution. In the first layer, convolutional operations are used to fuse information across magnitude, cosine and sine values. The next two layers are similar to the ones used for the Mel filterbank, resulting in features with a dimension of $C \times T \times A$, similar to the Mel filterbank features, but with $A = 32 \times G/8$.

Finally, for both the Mel filterbank and magnitude+phase features, the output array of dimension $C \times T \times A$ is passed to a linear layer yielding a representation array of dimension $C \times T \times D$, where D is the model dimension which is the same for both input features.

As an alternative or a complement to spatial features, we propose to preprocess the data with data-independent beamformers so that each resulting channel captures the speaker(s) from a specific angular sector. This preprocessing makes it possible to exploit the phase information corresponding to distinct speaker positions even in the situation when the MC-SA-ASR input features are Mel filterbank only.The beamformers for each angular sector is calculated from all I microphones, so the number of microphones and the number

of angular sectors are two different concepts and can vary independently. In the following, we compare three different inputs to MC-SA-ASR, as illustrated in Figure 4.4: the original 4-channel signals from 4 microphones, 4-channel beamformed signals extracted from 4 microphones, and 4-channel beamformed signals extracted from 8 microphones.

We calculate the beamformer for each angular sector by solving the Equation (2.3). The geometrical illustration of azimuth and elevation can be found in Figure 2.5. The integration over elevation ϕ is from -89.5 to 89.5 degrees, with a step size of 1 degree, and the integration over azimuth ϕ is from 0 to 359 degrees, also with a step size of 1 degree. The frequency range is from 0 to 8,000 Hz, with a step size of 31.25 Hz, resulting in a total of 256 frequency bins. For realistic consideration of speakers' possible positions with respect to the microphone array, we choose an elevation range from 10 to 60 degrees. The 4 angular sectors cover a width of 90° each, with azimuths ranging from 315 to 45°, from 45 to 135°, from 135 to 225°, and from 225 to 315°.

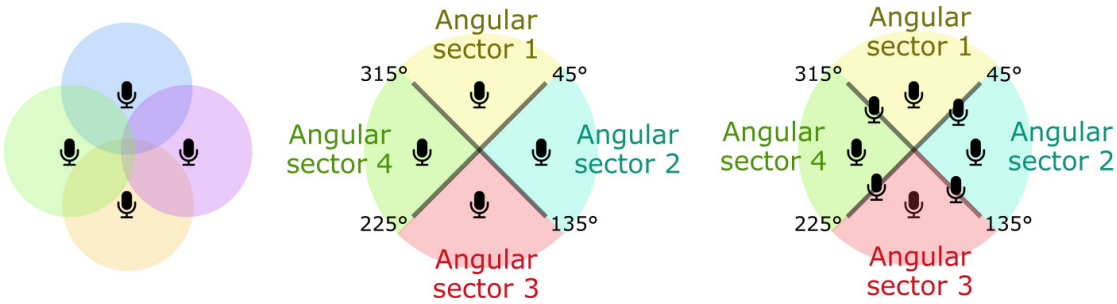
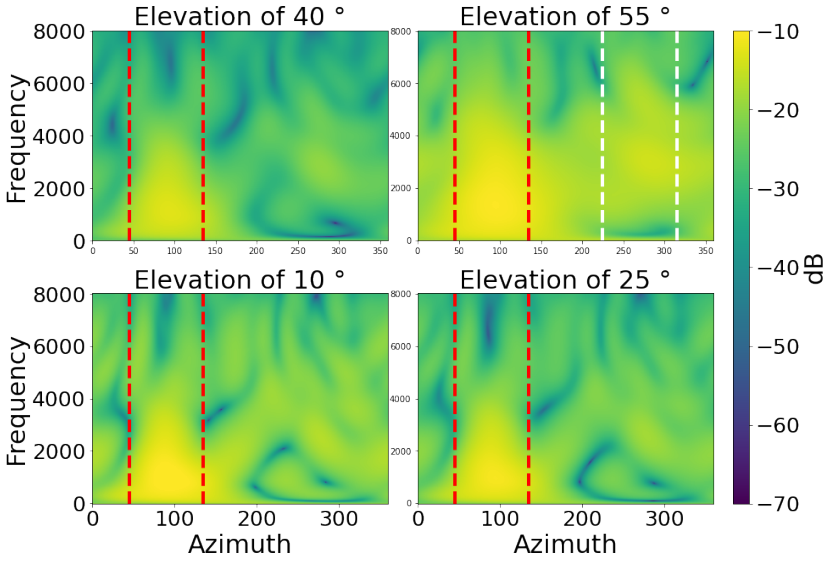
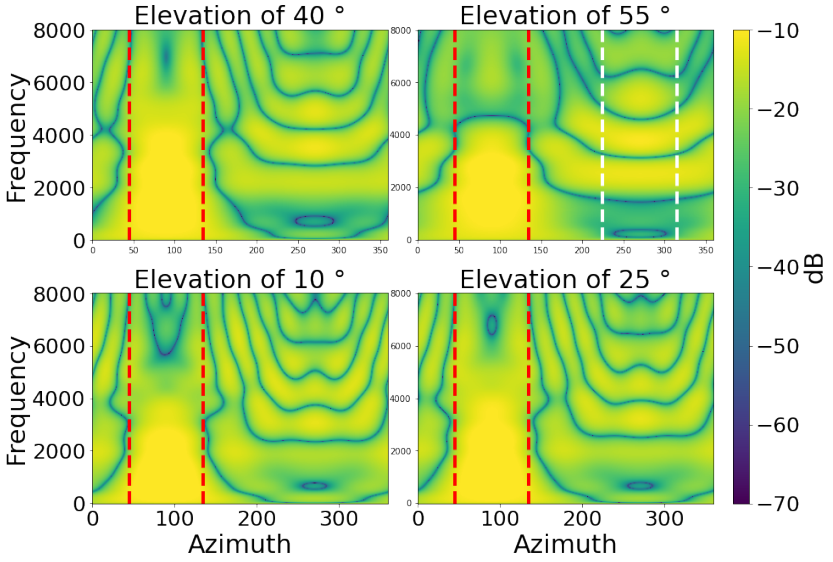


Figure 4.4: Original microphone channels (left), 4 beamformed channels (angles) from 4 microphones (center), and 4 beamformed channels from 8 microphones (right).

Figure 4.5 illustrates the response of the beamformer associated with the second angular sector. It can be observed that for both the 4- and 8-microphone settings, the responses below 2,000 Hz correspond to the desired response. The values within the target angular sector, an azimuth range from 45 to 135°, are higher than those in other sectors. For the 8-microphone setting, the quality of the response degrades above 4,000 Hz (2,000 Hz for 4-microphone), especially at elevations of 10° and 55°, because they are closer to the border of the target elevation range, which is from 10° to 60°. At an elevation of 55° for both the 4- and 8-microphone settings, higher values can be found in other sectors, such as the fourth sector with an azimuth range from 225° to 315°. Overall, the 8-microphone responses are sharper and more accurate than those of the 4-microphone setting.



(a) 4 microphones



(b) 8 microphones

Figure 4.5: Response of the fixed beamformer associated with the second angular sector. The zone between the red lines represents azimuth values from 45 to 135º. The zone between the white lines represents azimuth values from 225 to 315º.

4.2 Experimental setup

4.2.1 Datasets

Simulated multichannel multi-speaker LibriSpeech

We simulate a multi-speaker scenario using the LibriSpeech dataset (Panayotov et al., 2015). The train-960, dev-clean and test-clean subsets are used to generate our training, development and test sets, respectively. We assume a circular 2- to 8-microphone array with an aperture of 10 cm. We generate Room impulse responses (RIRs) using the gpuRIR toolkit (Diaz-Guerra et al., 2021). The length, width and the height of the room are randomly drawn in the range of 3 to 8 m and 2.4 to 3 m, respectively. The microphone and speaker positions are randomly sampled with the constraints that the center of the microphone array is at most 0.5 m away from the room center, the speakers are at least 0.5 m away from the walls, and their heights are 0.6 to 0.8 m above the middle plane. The RT60 value is randomly drawn in the range of 0.4 to 1 s. Each multi-speaker signal is generated by randomly drawing one utterance from each of 1 to 3 speakers, mixing these utterances convolved by RIRs, and concatenating the corresponding transcripts separated by the <sc> token. The start time of each utterance is shifted relative to the previous one, with a delay ranging from 0.5 seconds to the maximum duration of the previous utterance. This realistic choice also guarantees the first-in-first-out principle behind SOT. Each multi-speaker signal is linked to 8 speaker profiles[1], encompassing both the actual speakers in the signal and templates from randomly selected speakers out of the 2,484 LibriSpeech speakers. Speaker embeddings are computed as the average of two random enrollment sentences for each speaker.

AMI data

We validate the effectiveness of the proposed MC-SA-ASR model on real-world data by fine-tuning and testing our pre-trained model on the AMI meeting corpus (Carletta et al., 2005). We chose to use the segmentation method presented in Section 3.2.1 because of the following benefits. Firstly, the utilization of a hop size allows for an increased number of training samples. Secondly, segmenting outside the speaker overlap regions respects the FIFO training approach. We conducted experiments on the AMI MDM corpus with a chunk size of 5, 10 or 15 s and a hop size of 2 s. Figure 4.6 illustrates the distribution of the number of segments and speakers in the datasets generated using different chunk sizes. Table 3.1 presents some statistics of the training, development, and test sets generated with a chunk size of 5 s. It is observed that the probability of a chunk containing multiple speakers

[1]We chose a higher number to train a robust model, as real-life diarization results may identify more speakers than actually present.

increases as the chunk size increases. To evaluate the model's performance on datasets with varying numbers of speakers, we combined all the test sets segmented at 5, 10, and 15 s. We then divided them into four different test sets based on the number of speakers, which include datasets with 1, 2, 3 and 4 speakers, respectively. The number of segments for each speaker count is as follows: 5,737 segments with 1 speaker, 4,911 segments with 2 speakers, 3,986 segments with 3 speakers, and 1,936 segments with 4 speakers.

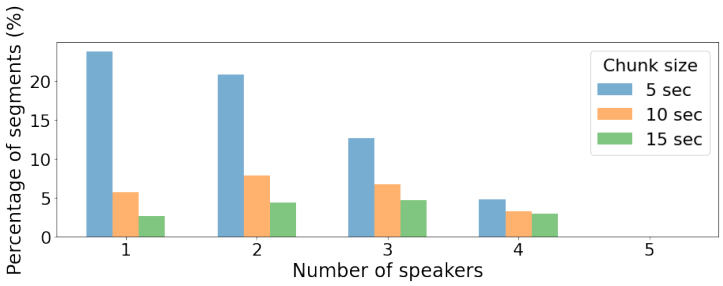


Figure 4.6: Percentage of segments containing a given number of speakers for different chunk sizes on the AMI corpus.

4.2.2 Model and training

4.2.2.1 Baseline

We choose two baseline systems to study the impact of a multichannel encoder that effectively utilizes speaker information. The first baseline is the end-to-end SC-SA-ASR system proposed by Kanda et al. (2021b). It helps us to compare the impact of a multichannel MFCCA-based ASR encoder on far-field speech. Our implementation of the SC-SA-ASR system uses ECAPA-TDNN speaker embeddings. The second baseline is the MFCCA-based multichannel ASR (MC-ASR) model proposed by Yu et al. (2023). The MC-ASR model has been extended to perform speaker identification in the MC-WD-SOT model presented by Shi et al. (2023b). However, the MC-WD-SOT model does not incorporate any speaker information in the ASR decoder. Hence, the performance of the ASR module in the MC-WD-SOT model is not expected to be better than that of the MC-ASR model. Thus, in favor of a simpler implementation, we choose the MC-ASR model as the second baseline.

4.2.2.2 Model description

We compute the STFT with a window length of 25 ms and a hop size of 10 ms. The magnitude information is used to generate 80-dimensional log Mel filterbank features. For

the magnitude+phase features, we generate 3×201-dimensional features that include the STFT magnitude and the cosine and sine of the STFT phase. According to the convolution process in Figure 4.3, the convolutional feature extractor produces features A of size 640 and 832 for the Mel filterbank features and magnitude+phase features, respectively. The speaker encoder uses 80-dimensional log Mel filterbank features averaged across all channels.

For all the models (SC-SA-ASR, MC-ASR and MC-SA-ASR) in our experiments comparing Mel filterbank and magnitude+phase features, the Conformer-based encoder has 12 layers, and the Transformer-based decoder has 1 layer. The speaker decoder in SC-MC-ASR and MC-SA-ASR has 2 layers. For the experiments comparing original channels with data-invariant beamformed channels, we use an MC-ASR model featuring 12 layers in the encoder and 6 layers in the decoder. We chose to use one decoder layer for model comparison to reduce computing time and memory consumption. For the experiments comparing the data-invariant beamformed input, we redefined the number of layers as 6. All multi-head attention mechanisms have 4 heads, the model dimension D is set to 256, and the size of the feedforward layer is 2,048. Following Yu et al. (2023), the context frame length F of MFCCA in MC-ASR and SA-MC-ASR is set to 2. Our text tokenizer is a SentencePiece model (Kudo and Richardson, 2018) with a vocabulary of 5,000 tokens. The speaker embedding model is an ECAPA-TDNN model pre-trained[2] on the VoxCeleb1 (Chung et al., 2019) and VoxCeleb2 (Nagrani et al., 2020) training data, generating 192 dimensional embeddings. In each reference speaker embedding matrix S, the number of speakers K is set to 8, and the embedding for each speaker is derived from two random enrollment sentences.

4.2.2.3 Training setup

Our experiments were implemented using the SpeechBrain toolkit (Ravanelli et al., 2021). On the simulated multi-speaker LibriSpeech data, all the models were trained until convergence. The ASR modules in SC-SA-ASR and MC-SA-ASR were pre-trained for 80 epochs by setting S and H^{spk} to zero. We utilized the Adam optimizer with a learning rate of 5×10^{-4} during the pre-training process. Subsequently, the ASR and speaker modules of the SC-SA-ASR model were fine-tuned for 60 epochs using a learning rate of 2.5×10^{-4}. The MC-SA-ASR model was fine-tuned for 120 epochs using a learning rate of 1.5×10^{-5}, after the initial training of 60 epochs. The weight of the speaker loss was set to to 0.1, following prior work (Kanda et al., 2020a). The MC-ASR model with 1-layer decoder was trained for 140 epochs, and the MC-ASR model with 6-layer decoder was trained for 120 epochs, both with a learning rate of 5×10^{-4}. For all the experiments, the global batch size

[2] Available at https://huggingface.co/speechbrain/spkrec-ecapa-voxceleb

(batch size × number of GPUs × gradient accumulation factor) is fixed to 160.

We fine-tune the pre-trained SC-SA-ASR and MC-SA-ASR models on the AMI MDM datasets using the Full-corpus-ASR[3] partition. SC-SA-ASR utilizes only the 1st channel of Array 1[4] of the train-dev-test splits; while MC-SA-ASR utilizes the 1st and 5th channels for the 2-channel model. The datasets with chunk sizes of 5, 10, and 15 s undergo fine-tuning for 40 and 90 epochs, respectively. In each case, the first half of all training epochs updates the ASR module, while the second half jointly updates the ASR and speaker modules. All fine-tuning steps employ the Adam optimizer with a learning rate of 1×10^{-4}, and a global batch size of 160.

4.2.3 Metrics

We use the word error rate (WER) to evaluate the ASR task, and the token-level speaker error rate (T-SER) and sentence-level speaker error rate (S-SER) (Kanda et al., 2020a) to evaluate the speaker prediction task. In order to calculate the S-SER, a single speaker is assigned to each hypothesized sentence by taking the maximum over token-level speaker prediction counts within the respective sentences segmented by <sc>. Furthermore, we evaluate the speaker counting accuracy (Kanda et al., 2020a) via the confusion scores

$$\text{Confusion\_score}(i, k) = \frac{N_k^i}{N_k}, \tag{4.8}$$

where N_k^i is the number of test signals where the estimate indicates i speakers when the ground truth has k speakers, and N_k represents the number of test signals where the ground truth has k speakers.

We employed the SCTK toolkit (NIST, 2024) to conduct significance tests, specifically the Matched Pair Sentence Segment test. We highlight in bold the best WER/SER result and the results statistically equivalent to it at a 0.05 significance level.

4.3 Evaluation results

4.3.1 Results of MC-SA-ASR using Mel filterbank vs. magnitude+phase features

On simulated multichannel multi-speaker LibriSpeech data

[3] All partitions can be found at https://groups.inf.ed.ac.uk/ami/corpus/datasets.shtml.
[4] Few meetings have two arrays, each array consisting of 8 microphones.

Table 4.1 presents the results achieved by the baseline models (SC-SA-ASR and MC-ASR) and the proposed model (MC-SA-ASR) on simulated multi-speaker Librispeech data. First of all, by comparing the results of SC-SA-ASR and MC-SA-ASR, we can conclude that using a multichannel encoder to process multi-microphone speech information improves the speech recognition performance. Specifically, the MC-SA-ASR model with Mel filterbank input features achieves a WER of 14.77% in the 4-channel scenario, that is a 12% relative reduction compared to the SC-SA-ASR model (16.81%). On average, the WER of the MC-SA-ASR models with 2, 3, and 4 channels (15.14%) is reduced by 10% relative compared to the SC-SA-ASR model. Secondly, the proposed MC-SA-ASR model, which incorporates speaker information into the ASR decoder, obtains a 16% relative reduction in WER compared to MC-ASR (18.03%). This suggests that leveraging speaker information can improve the performance of multi-speaker ASR in a multichannel setting.

Table 4.1: WER (%), S-SER (%), and T-SER (%) of different systems on the simulated multichannel multi-speaker LibriSpeech test set. Results are grouped by the number of speakers in the simulated mixture. The '1,2,3-spk' column shows the results obtained on a test set containing mixtures of 1, 2, and 3 speakers.

System	Feat	#Chn	1-spk			2-spk			3-spk			1,2,3-spk			#Prm
			WER	S-SER	T-SER	WER	S-SER	T-SER	WER	S-SER	T-SER	WER	S-SER	T-SER	
Baseline															
SC-SA-ASR	Mel	1	7.15	1.64	3.13	14.68	3.23	5.42	20.39	**5.59**	7.67	16.81	**3.95**	6.18	61.1M
MC-ASR	Mel	2	8.52	-	-	16.76	-	-	22.87	-	-	18.21	-	-	27.8M
		3	8.29	-	-	16.54	-	-	22.01	-	-	18.03	-	-	
		4	8.11	-	-	16.43	-	-	21.76	-	-	17.84	-	-	
Proposed															
MC-SA-ASR	Mel	2	**6.64**	**1.15**	2.84	14.21	3.19	**5.14**	19.03	5.92	7.34	15.41	4.34	6.46	51.6M
		3	**6.62**	**1.11**	**2.73**	14.24	3.15	5.56	18.92	6.11	7.29	15.25	4.12	6.16	
		4	7.17	1.41	2.95	14.09	**3.06**	**5.10**	18.70	6.06	**6.93**	**14.77**	**4.05**	6.40	
	Mag+Phase	2	7.24	1.53	3.30	15.42	3.85	6.22	20.34	7.06	7.98	16.76	4.87	7.29	
		3	6.86	1.78	3.39	**13.69**	3.72	5.39	18.14	6.86	7.83	15.04	4.08	6.28	
		4	6.91	1.86	3.43	**13.97**	**3.10**	5.85	**18.03**	6.69	7.26	**14.69**	4.12	**5.85**	

The test WERs obtained by the proposed MC-SA-ASR model with Mel filterbank input features and with magnitude+phase input features do not exhibit significant differences. However, interestingly, in the 2-channel scenarios, Mel filterbank features (15.41%) outperformed magnitude+phase (16.76%) with a 8% relative lower WER. Conversely, in the test results of the 3- and 4-channel scenarios, magnitude+phase features achieved a slight relative reduction of 1.4% and 0.5% in WER, respectively. Moreover, in the 3-channel scenario, magnitude+phase features exhibited a relative reduction of 4% in WER on both 2-

and 3-speaker datasets compared to Mel filterbank features. From this, we can conclude that in multi-speaker multichannel ASR, phase features perform better on chunks with a larger number of speakers. This can be explained by the fact that as the number of speakers in a room increases, the positional information of the speakers has a greater impact on ASR performance.

On AMI data

Table 4.2 presents the test results of SC-SA-ASR and MC-SA-ASR on datasets segmented using different chunk sizes. MC-SA-ASR consistently outperforms SC-SA-ASR across all datasets. Particularly, with a chunk size of 5 s, 2-channel MC-SA-ASR achieves a relative 2% reduction in WER compared to SC-SA-ASR. We would also like to highlight that our proposed MC-SA-ASR model exhibits a 16% relative reduction in model size compared to the SC-SA-ASR model which has 61M paramters. This reduction comes by replacing the first (feedforward) layer of the original Conformer by a smaller-sized MFCCA.

Table 4.2: WER (%), sentence-level SER (S-SER) (%) and token-level SER (T-SER) (%) of SC-SA-ASR and MC-SA-ASR models fine-tuned on AMI with a consistent chunk size (5, 10 or 15 s) across train, dev and test splits. Our proposed MC-SA-ASR uses 2 channels.

System	5 s			10 s			15 s		
	WER	**S-SER**	**T-SER**	**WER**	**S-SER**	**T-SER**	**WER**	**S-SER**	**T-SER**
SC-SA-ASR	46.03	**33.86**	**28.54**	46.54	40.88	28.95	48.78	45.93	30.87
MC-SA-ASR	**45.15**	34.14	29.08	**45.97**	**39.93**	**28.00**	**48.49**	**44.58**	**28.56**

We also compare the performance of models trained on different chunk sizes on all the test sets. Table 4.3 presents the results, divided into 4 subsets based on the number of speakers. We can observe that 2-channel MC-SA-ASR consistently achieves a lower WER than SC-SA-ASR. Particularly, the 5-second model demonstrates a 6% relative reduction in WER compared to SC-SA-ASR on the 2-speaker test set. Moreover, we observe that models trained on smaller chunk sizes perform better. In MC-SA-ASR, the 5-second model exhibits relative reductions in WER of 43%, 34%, 17%, and 11% on the test sets with 1, 2, 3 and 4 speakers compared to the 15-second model, respectively (note that the SERs exhibit a similar trend). This might be explained by the fact that, according to Figure 4.6, when decreasing the number of speakers, both the total number and the proportion of 5-second segments in the training set increase compared to 15-second segments.

The comparison of S-SER and T-SER in Table 4.2 shows that the S-SER is much higher than the T-SER on the AMI test set, for both SC-SA-ASR and MC-SA-ASR, as opposed to lower S-SER and higher T-SER values on the LibriSpeech test set in Table 4.1.

Table 4.3: WER (%) and token-level SER (T-SER) (%) of SC-SA-ASR and MC-SA-ASR models fine-tuned on AMI with different chunk sizes across train and test splits. Rows 5 s, 10 s and 15 s refer to the chunk size in train-dev splits. The test set consists of 5, 10 and 15 s chunks. MC-SA-ASR uses 2 channels.

System	1-spk		2-spk mix		3-spk mix		4-spk mix	
	WER	T-SER	WER	T-SER	WER	T-SER	WER	T-SER
SC-SA-ASR								
5 s	28.10	**13.37**	42.03	25.53	54.04	35.65	67.27	43.17
10 s	31.84	17.25	43.93	27.88	54.20	35.82	67.92	43.49
15 s	48.38	32.38	60.73	43.12	64.39	44.53	73.68	49.40
MC-SA-ASR								
5 s	**27.68**	**13.43**	**39.54**	**24.83**	**52.76**	35.70	**64.72**	**41.24**
10 s	31.81	17.52	43.77	27.18	53.92	**34.99**	66.54	42.80
15 s	48.69	31.79	60.34	42.48	63.66	44.19	72.69	48.36

The reason is that the prediction of speaker change markers <sc> is a more challenging task on the AMI test set. To quantify this issue, we evaluate the performance of the two systems in terms of speaker counting task on the AMI test set.

Table 4.4 presents the speaker counting accuracy obtained by counting the occurrences of <sc> tokens in the ASR output, on datasets with different numbers of speakers. We observe that the MC-SA-ASR system is consistently outperforming the SC-SA-ASR system. However, the accuracy decreases by 60% relative from the 3-speaker test set to the 4-speaker set for MC-SA-ASR (61% for SC-SA-ASR). Furthermore, for scenarios involving 2, 3, and 4 speakers, the majority of errors originate from underestimating the number of speakers.

Table 4.4: Speaker counting accuracy (%) on the 5-, 10- and 15-second AMI test chunks for models trained on 5-second chunks.

System	# Speakers	Estimated # speakers				
		1	2	3	4	>4
SC-SA-ASR	1	91.75	7.53	0.64	0.06	0.00
	2	11.26	76.82	10.73	1.03	0.14
	3	1.75	36.67	50.12	8.83	2.60
	4	0.36	21.02	51.65	19.31	7.64
MC-SA-ASR	1	**92.40**	6.98	0.48	0.06	0.05
	2	11.57	**77.05**	10.71	0.54	0.10
	3	1.80	37.18	**50.74**	8.50	1.75
	4	0.82	18.59	53.87	**20.04**	6.66

4.3.2 Results of MC-ASR using data-invariant beamforming

We first compare the speech quality of different types of signals by visualizing the spectrogram of an AMI meeting chunk before and after beamforming. As shown in Figure 4.7, both beamformed signals significantly reduce ambient noise and reverberation compared to the original far-field channel. This demonstrates that beamforming signals from the direction of speaker(s) can effectively enhance the speech region. By comparing the signal extracted from 4 microphones with the one extracted from 8 microphones, particularly in the zone between the white lines, we observe that the latter provides better dereverberation in the speech region by reducing interfering speech and enhancing the target speech. Similar phenomena can be observed in other speech segments, where the signal from 8 microphones shows clearer speech formants. This demonstrates that using more microphones for data-invariant beamforming can leverage phase information to better extract speech regions and enhance the target speech.

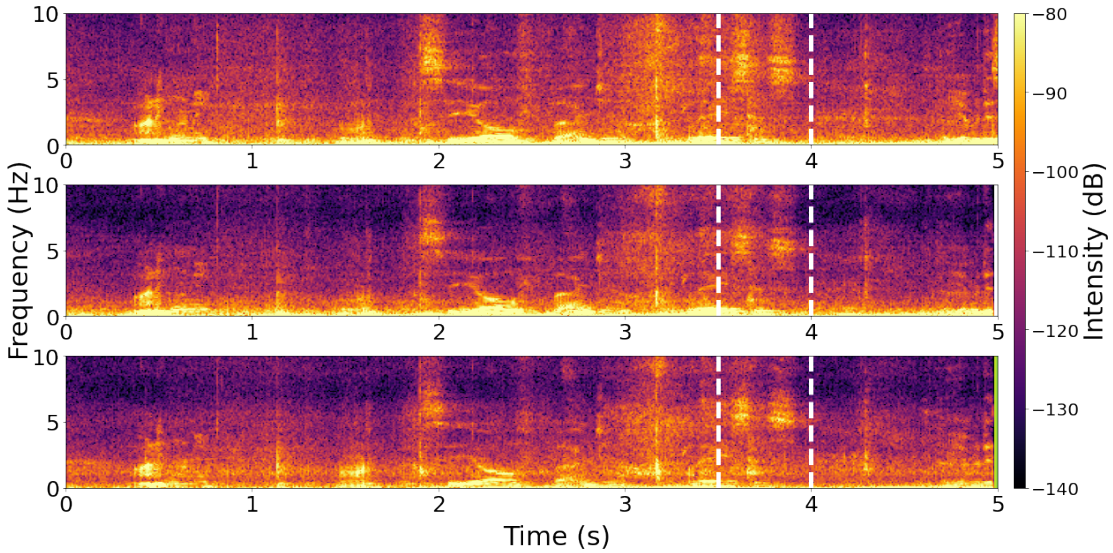


Figure 4.7: Spectrogram of one AMI test chunk using data-invariant beamformers. From top to bottom: 1st array channel; beamformed signal of 1st sector from 4 microphones; beamformed signal of 1st sector from 8 microphones.

We then compare the results of MC-ASR trained on original channels versus data-invariant beamformed channels. Table 4.5 shows the test results on the LibriSpeech test set using a pretrained model and on the AMI test set using a fine-tuned model based on the corresponding pretrained model. On the LibriSpeech test set, the model trained on 4-channel beamformed signals from 4 microphones (noted as 4-4 model) does not show

improvement compared to the model trained on the original 4-channel signals (noted as 4- model), with WERs of 17.64% vs. 16.30% on the 1,2,3-speaker mixture test set, respectively. However, the model trained on 4-channel beamformed signals from 8 microphones (noted as 8-4 model) demonstrates a 13% relative improvement in WER compared to the 4- model, reducing the WER from 16.30% to 14.26% on the 1,2,3-speaker mixture test set. For the 2-speaker mixture test set, the 4-4 model and 8-4 model show better performance compared to the 4- model, with a 14% relative reduction in WER (from 14.82% to 12.27%) for the 4-4 model and a 29% relative reduction in WER (from 14.82% to 10.53%) for the 8-4 model.

Table 4.5: WER (%) on original vs. beamformed channels for MC-ASR (27.8M) trained on synthetic LibriSpeech, and fine-tuned on AMI.

Input type	# Micro	# Sectors	LibriSpeeh test set				AMI test set			
			1-spk	2-spk	3-spk	1,2,3-spk	1-spk	2-spk	3-spk	1,2,3,4-spk
Original	4	-	7.21	14.82	19.76	16.30	25.89	41.70	54.68	45.25
Beamformed	4	4	7.46	12.27	22.99	17.64	24.52	**39.61**	52.19	43.14
	8	4	**6.73**	**10.53**	**19.06**	**14.26**	**22.96**	40.01	**49.59**	**41.64**

On the AMI test set, both the 4-4 and 8-4 models show significant improvement compared to the 4- model. Specifically, the 8-4 model achieves an 8% relative reduction in WER (from 45.25% to 41.64%) on the 1, 2, 3-speaker mixture test set compared to the 4- model. In more detail, the 8-4 model demonstrates a reduction in WER of 11% relative (from 25.89% to 22.96%), 4% relative (from 41.70% to 40.01%), and 9% relative (from 54.68% to 49.59%) for the 1, 2, and 3-speaker groups, respectively. These improvements highlight the efficiency of fine-tuning on beamformed signals to better exploit phase information.

Moreover, the 8-4 model consistently performs better than the 4-4 model on both the LibriSpeech and AMI test sets. This demonstrates that using beamformed signals from a larger number of microphones results in better performance for the MC-ASR model. This is because more microphones allow the filter for a certain sector to incorporate more information, thereby providing better enhancement (see Figure 4.5) of signals from that sector.

4.4 Summary

In this chapter, we have introduced an end-to-end MC-SA-ASR system that combines a Conformer-based encoder with multi-frame cross-channel attention and a speaker-attributed Transformer-based decoder. Experimental results demonstrate that, on simulated data, our approach achieves relative reductions in WER of up to 12% and 16% com-

pared to existing single-channel and multichannel methods, respectively. We also studied the impact of using Mel filterbank vs. magnitude+phase features on MC-SA-ASR. On real-world data, our model achieves a relative reduction in WER of up to 6% compared to SC-SA-ASR. However, it still has limitations in accurately determining the number of speakers in scenarios involving three or more participants. Future research can focus on improving this aspect. Finally, we have demonstrated that models trained on data-invariant beamformed data can improve MC-ASR performance with up to a 29% relative in WER. This opens up new possibilities for enhancing input features for multichannel input.

5 Joint beamforming and speaker-attributed ASR

磨刀不误砍柴工。
Sharpening the axe will not delay the cutting of firewood.

Chinese proverb

In Chapter 4, we integrated the multi-frame cross-channel attention (MFCCA) module to the traditional SA-ASR in order to explore features across channels and time. Such multichannel attention schemes are often believed to outperform classical beamforming, yielding state-of-the-art performance (Yu et al., 2023). Yet, in contrast to a frequency-dependent complex-valued beamformer, they rely on frequency-independent real-valued weights, which results in limited noise and reverberation reduction. The data-independent beamformer explored in Chapter 4 address this limitation, but they result in limited enhancement performance, hence do not always outperform MFCCA.

In this chapter, we propose to combine SA-ASR with a data-dependent beamforming front-end. The beamformer reduces noise and reverberation and fuses the mixture channels into a single-channel enhanced mixture that is fed to SA-ASR. While such a front-end is common in single-speaker scenarios, extending it to real multi-speaker scenarios is nontrivial. First, the beamformer must vary its spatial response over time according to the speakers' positions and activity patterns. This is why few examples of multi-speaker beamforming front-ends are found in the literature. BeamformIt (Anguera et al., 2007) was used as a front-end to PIT based multi-speaker ASR (Chang et al., 2019) as well as to single-speaker ASR baselines for multi-speaker ASR tasks (Ochiai et al., 2017; Lee et al., 2020; Watanabe et al., 2020), while minimum variance distortion-less response (MVDR) beamforming was used as a front-end to SA-ASR by Kanda et al. (2023) without comparison to MFCCA. To the best of our knowledge, no full-neural beamforming front-end was used for SA-ASR so far. Second, as already mentioned in Chapter 3, the pretraining of a neural beamformer on

real meeting data is challenging due to the absence of ground truth noiseless dry mixture signals as pretraining targets. For instance, the AMI corpus involves a multiple distant microphone (MDM) array and headset microphones for the individual speakers, but using the sum of headset microphones as the ground truth is not directly feasible due to the delay between the audio captured by each headset and that captured by each MDM channel, which varies over time depending on the speaker's position.

In this chapter we address four goals. First, we utilise the data alignment and augmentation method presented in Section 3.1.2 to pretrain a multi-speaker neural beamformer on a real meeting corpus. Note that the beamformer is employed to reduce noise and reverberation, but not to extract the individual speakers. Second, we propose a pipeline integrating beamforming with SA-ASR, aiming to improve both speech and speaker recognition. Third, we evaluate the differences in performance between statistical, hybrid, and neural beamformers. Finally, we jointly optimize the neural beamformer and the SA-ASR model.

The chapter is organized as follows. Section 5.1 presents the considered beamformers and SA-ASR model. Section 5.2 introduces experimental settings including the AMI data alignement and augmentation pipeline. Section 5.3 describes our experimental setup and results. We conclude in Section 5.4.

5.1 System architecture

We propose a joint system integrating beamforming and SA-ASR for multichannel, distant-microphone meeting transcription. This system aims to tackle the complexities associated with recognizing and transcribing speech captured in environments where multiple speakers and background noise are present. As illustrated in Figure 5.1, the multichannel audio input undergoes a two-stage processing pipeline. In the first stage, the audio is processed by a beamformer to generate enhanced single-channel audio. The beamforming stage employs spatial filtering techniques to focus on the desired speech source while reducing noise and reverberation, thus significantly improving the clarity of the speech signals. The second stage involves feeding the beamformed audio into the SA-ASR model to obtain both speech and speaker recognition results. We compare the performance of the fixed DAS beamformer, the hybrid DNN-MVDR (noted as MVDR) beamformer, and the fully neural FaSNet beamformer. Our experiments include fine-tuning the SA-ASR model on training data enhanced with each respective beamformer. Furthermore, we employ backpropagation from the SA-ASR loss to FaSNet, enabling the neural beamformer to be fine-tuned according to the SA-ASR training objective. This integrated approach aims to optimize the entire

system for improved transcription accuracy in challenging acoustic environments.

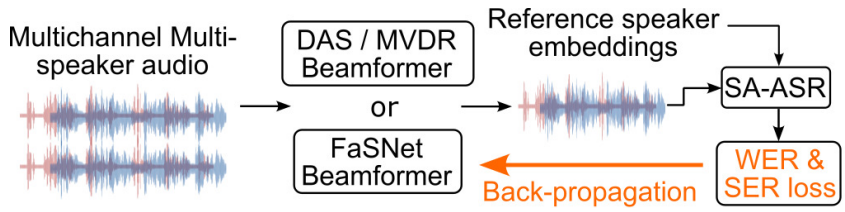


Figure 5.1: Proposed joint system of beamformer and SA-ASR.

5.1.1 DAS beamforming

The delay-and-sum (DAS) beamformer (Johnson and Dudgeon, 1993) is a fixed beamformer, which depends only on the delays between the microphone signals and a reference microphone. Equation details can be found in Equation (2.2). The DAS beamformer was employed as front-end in many single-speaker ASR studies, e.g., by Xiao et al. (2016). It involves computing the delays using a time difference of arrival (TDOA) estimator such as the generalized cross-correlation with phase transform (GCC-PHAT) (Knapp and Carter, 1976), shifting the phase of the microphone signals accordingly in the complex short-time Fourier transform (STFT) domain, and summing them. The DAS beamformer, which consists of a basic fixed beamforming technique, serves as our baseline method.

5.1.2 MVDR Beamforming

Deep Neural Network (DNN)-based Minimum Variance Distortionless Response (MVDR) beamforming (Lu et al., 2022; Kim et al., 2023) combines neural networks with traditional beamforming methods. The DNN is trained to estimate masks in the time-frequency domain that enhance desired signals and suppress interference. This information is then used to compute the MVDR beamformer weights, which minimize output power while preserving target signals. The MVDR beamformer leverages the spatial covariance matrices of both the speech signal and noise to calculate these weights, ensuring effective noise reduction and signal enhancement. Specifically, the beamforming weights are derived by normalizing the product of the inverse noise covariance matrix and the steering vector, which corresponds to the largest eigenvalue of the speech covariance matrix. This method can be seen as a transition between traditional mathematical beamforming and fully neural network-based approaches.

5.1.3 FaSNet with TAC

The Filter-and-Sum Network (FaSNet) system (Luo et al., 2019) employs a fully neural architecture and aims to directly estimate time-domain beamforming filters. It employs a two-stage design: the first stage estimates filters for a reference microphone, and the second stage estimates filters for the remaining microphones based on pairwise cross-channel features between the pre-separated output and each microphone. The FaSNet architecture utilizes dual-path recurrent neural networks (DPRNNs) (Luo et al., 2020b) to extract information from both the channel and frame levels. The Transform-average-concatenate (TAC) (Luo et al., 2020a) design paradigm addresses channel permutation and is capable of handling various numbers of microphones. Furthermore, the FaSNet model's fewer training parameters contribute to its ability to perform low-latency processing effectively.

5.1.4 Dereverberation with WPE

Multichannel dereverberation using Weighted Prediction Error (WPE) (Nakatani et al., 2010) reduces reverberation by modeling and subtracting late reverberant components from the observed audio signal using long-term linear prediction. This technique optimizes prediction coefficients and error weights to enhance speech clarity and intelligibility in reverberant environments. We employ a specific implementation known as Generalized WPE, as proposed by Yoshioka and Nakatani (2012). This approach integrates the WPE method with a conditional separation and dereverberation method (Yoshioka et al., 2010), thereby revealing the common mathematical foundation that unites these established methods. Multiple studies (Yang and Chang, 2020; Dowerah et al., 2022) have shown that using WPE in conjunction with beamforming can enhance audio quality, thereby improving the performance of speech recognition or speaker identification.

5.1.5 SA-ASR

The Transformer-based end-to-end speaker-attributed ASR (SA-ASR) system (Kanda et al., 2021b) is employed as back-end system of ASR and diarization. Details can be found in Section 2.2.2.3. Following the Serialized Output Training principle (Kanda et al., 2020b), the output concatenates all speakers' sentences in first-in-first-out order, and each token is associated with one speaker ID.

5.2 Experiments on AMI

5.2.1 Datasets

Mixed AMI

We applied a similar alignment method as described in Section 3.1.2 to pretrain a multi-speaker neural beamformer on a real meeting corpus. The difference is that we sum the corresponding aligned headset clips to obtain the corresponding ground-truth enhanced mixtures, as shown in Figure 5.2.

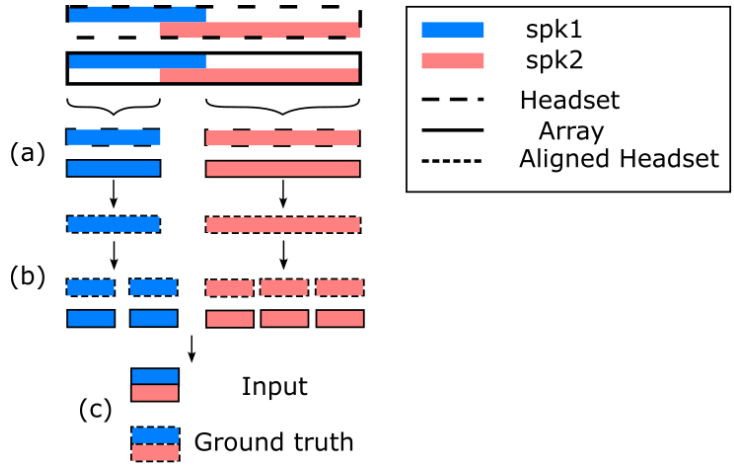


Figure 5.2: Mixture generation from real meeting data.

To train the MVDR and FaSNet beamformer, we apply this method to the AMI meeting corpus to create mixtures of real single-speaker AMI segments and the corresponding ground truths. The preparation of the data is similar to Section 3.2.1.

Simulated multichannel multi-speaker LibriSpeech data

To achieve good performance on AMI, the SA-ASR model should be pretrained on a larger simulated distant-microphone multi-speaker dataset (Yang et al., 2023). We created a 960 h training set and a 20 h development set from the LibriSpeech train-960 and dev-clean sets (Panayotov et al., 2015). We employed the method described in Section 4.2.1 to generate 2-, 4- and 8-microphone data containing 1 to 3 speakers.

Real AMI

After it has been pretrained on multi-speaker LibriSpeech, the SA-ASR model is fine-tuned and evaluated on real AMI MDM data. We utilize the segmentation method in Section 4.2.1 to partition the MDM data into 5 s chunks and adjust the chunk start/end

times to non-overlapped word boundaries. The resulting Real AMI dataset contains respectively 165 h, 19 h, and 19 h for training, development, and test. For both Mixed AMI and Real AMI, we consider 2-, 4- and 8-channel settings. For Real AMI, the channels are taken from Array1,[1] namely channels 1 and 5 in the 2-channel setting, channels 1, 3, 5, and 7 in the 4-channel setting, and all 8 channels in the 8-channel setting.

5.2.2 Model description

We utilize the implementation of DAS from the SpeechBrain toolkit (Ravanelli et al., 2021). For the MVDR model, we utilize the implementation in TorchAudio (Yang et al., 2022; Lu et al., 2022), which employs the Conv-TasNet (Luo and Mesgarani, 2019) mask generator as the DNN module. The number of filterbank output channels and the number of bins in the estimated masks are both 513. The implementation of FaSNet with TAC is from the Asteroid toolkit (Pariente et al., 2020). The frame size and context size of FaSNet are set to 4 and 16 ms, respectively. The encoder dimension and feature dimension are 64. The dual path blocks consist of a 4-layer dual model. We implemented SA-ASR and the MFCCA-based MC-SA-ASR system in Chapter 4 as a baseline using SpeechBrain. In SA-ASR and MC-SA-ASR, the Conformer-based encoder, the Transformer-based decoder and the speaker decoder have 12, 6 and 2 layers, respectively. All multi-head attention mechanisms have 4 heads, the model dimension is 256, and the size of the feedforward layer is 2,048. MC-SA-ASR has fewer parameters than SA-ASR due to replacing the first (feedforward) layer of the original Conformer by smaller-sized MFCCA. The speaker embedding model is an ECAPA-TDNN (Emphasized Channel Attention, Propagation, and Aggregation in Time-Delay Neural Network) (Desplanques et al., 2020) pre-trained[2] on the VoxCeleb (Chung et al., 2019; Nagrani et al., 2020) corpora, yielding 192 dimensional embeddings.

Additionally, we test the performance obtained with WPE, implemented by Drude et al. (2018), during the evaluation of the MC-SA-ASR model. We also compare the performance of beamforming with and without WPE during the fine-tuning of SA-ASR.

5.2.3 Training setup

The MVDR and FaSNet beamformer are trained on Mixed AMI for 80 epochs using the Adam optimizer with a learning rate of 10^{-3}.

[1] A few meetings have two arrays, each consisting of 8 microphones.
[2] Available at `https://huggingface.co/speechbrain/spkrec-ecapa-voxceleb`

The ASR modules in SA-ASR and MC-SA-ASR are pretrained on multi-speaker LibriSpeech for 80 epochs using Adam with a learning rate of 5×10^{-4}. The ASR and speaker modules are then further pretrained on multi-speaker LibriSpeech for 60 epochs using a learning rate of 2.5×10^{-4}.

After this, SA-ASR and MC-SA-ASR are fine-tuned on either unprocessed (baseline) or beamformed Real AMI data for 15 epochs, using Adam with a learning rate of 3×10^{-4}. We fine-tune the ASR module for the first 8 epochs and jointly fine-tune the ASR and speaker modules in the last 7 epochs.

When jointly optimizing FaSNet and SA-ASR, the FaSNet model is not trained to convergence. Instead, to assess the impact of different pretraining levels, we pretrain it for 0, 5, 10, or 50 epochs. After this, the FaSNet and SA-ASR models are jointly fine-tuned on Real AMI for 15 epochs.

5.2.4 Metrics

We assess the beamforming performance via the scale-invariant signal-to-distortion ratio (SI-SDR) and its improvement (SI-SDRi) in dB on the Mixed AMI test set. We calculate the baseline SI-SDR for SA-ASR by defining the array mixture signal as the estimated signal, ensuring that the SI-SDRi is 0 dB without beamforming. For all beamforming methods, we calculate the SI-SDR by defining the beamformed signal as the estimated signal and compute SI-SDRi by subtracting the corresponding baseline SI-SDR. The performance of SA-ASR is evaluated by the word error rate (WER) and the sentence-level speaker error rate (SER) (Kanda et al., 2020a) in % on the Real AMI test set.

For all the results in this chapter, we employed the SCTK toolkit (NIST, 2024) to conduct significance tests, specifically the Matched Pair Sentence Segment test. For the mixture of 1,2,3 and/or 4 speakers, the best WER/SER and the results statistically equivalent to it at a 0.05 significance level are highlighted.

5.3 Evaluation results

5.3.1 Fine-tuning SA-ASR with DAS vs. with frozen MVDR and FaSNet

We initially evaluated an SA-ASR model fine-tuned on the first channel of Real AMI. The resulting WER and SER on mixtures of 1, 2, 3, or 4 speakers were 44.54% and 34.73%, respectively. However, when testing the same model on FaSNet beamformed 2-channel

Real AMI, the WER and SER increased to 64.32% and 46.30%. This discrepancy can be attributed to the distinct acoustic characteristics between far-field and beamformed scenarios. Models trained on far-field data may lack robustness to the cleaner, less reverberant conditions of beamformed audio. This observation contrasts with one of the conclusions in Chapter 3 and findings from the CHiME challenge studies, such as those reported by Cornell et al. (2023). In the CHiME challenges, it was demonstrated that using noisy training data can indeed enhance the robustness of models, particularly when large datasets are available for training. However, when training data is limited, as is the case with the 80 hours of AMI training data, simply using noisy data may not be as effective. This discrepancy arises because, with a large training dataset, incorporating noisy examples can help the model generalize better to real-world conditions. However, for smaller datasets, the combination of both clean and noisy data, rather than using noisy data alone, is more beneficial for training, as proposed by Ko et al. (2017). The limited AMI training data implies that using only noisy data might compromise the model's ability to adapt to different acoustic conditions, thus potentially reducing the adaptation capacity of the SA-ASR model. Therefore, in all following experiments, we fine-tune the SA-ASR model on real AMI training data enhanced using the same beamformer as the test data, which is essential to adapt it to the specific conditions of the test set.

Table 5.1 shows the test results of the baseline models (SA-ASR and MC-SA-ASR) and the combination of SA-ASR with three beamformers, where the parameters of MVDR and FaSNet are frozen during fine-tuning. Without WPE, the WER comparison between SA-ASR (44.54%) and MC-SA-ASR (44.99%) demonstrates that, while MFCCA had achieved a 16% relative WER reduction on simulated data in Section 4.3, it is inefficient on real meeting data. In general, fine-tuning the SA-ASR model on beamformed audio improves the ASR performance. Particularly in the 8-channel setting, using the DAS beamformer leads to a 6% relative reduction in WER without WPE (41.71%) and 8% with WPE (40.96%) compared to SA-ASR. It is also interesting to note that, despite FaSNet's superior denoising and dereverberation performance in terms of SI-SDRi, the SA-ASR model trained on speech beamformed by FaSNet performs less effectively than the one trained on speech beamformed by DAS. In the 8-channel setting, without WPE, using DAS results in a 5% relative reduction in WER compared to using FaSNet (from 44.11% to 41.71%), and a reduction of 4% relative in SER is observed for 4-channel (from 35.90% to 34.29%). The WER relative reduction is up to 6% (from 44.23% to 41.71%) compared to the MVDR-SA-ASR system. The latter system has a similar performance to the FaSNet-SA-ASR system.

To find the reason for the difference between DAS-SA-ASR and FaSNet-SA-ASR, we

Table 5.1: SI-SDR(%), SI-SDRi (%), WER (%) and SER (%) for models fine-tuned and tested on unprocessed (SA-ASR and MC-SA-ASR) or beamformed (DAS-SA-ASR, MVDR-SA-ASR, FaSNet-SA-ASR) data. For convenience, we denote SA-ASR, SI-SDR, and SI-SDRi as SA, SDR, and SDRi, respectively.

System	#Prm	#Chn	Mixed AMI test set								Real AMI test set							
			1-spk		2-spk		3-spk		1,2,3,4-spk		1-spk		2-spk		3-spk		1,2,3,4-spk	
			SDR	SDRi	SDR	SDRi	SDR	SDRi	SDR	SDRi	WER	SER	WER	SER	WER	SER	WER	SER
SA	69M	1	5.41	0	5.75	0	5.79	0	5.66	0	26.76	11.92	40.23	32.66	52.31	45.11	44.54	34.73
MC-SA	59M	2	5.41	0	5.75	0	5.79	0	5.66	0	26.41	11.73	40.79	32.64	52.59	43.82	44.99	34.43
+WPE (test)		2	5.72	0.31	5.93	0.18	5.92	0.13	5.87	0.21	26.43	12.13	40.80	32.54	52.32	44.14	44.72	34.65
DAS-SA	69M	2	5.62	0.21	5.42	-0.33	5.23	-0.56	5.39	-0.27	25.59	12.82	40.36	33.87	52.04	45.55	44.03	35.56
		4	5.66	0.25	5.47	-0.28	5.25	-0.54	5.35	-0.31	24.43	12.23	39.51	33.26	50.25	42.98	42.34	34.29
		8	5.66	0.25	5.48	-0.27	5.30	-0.49	5.38	-0.28	23.51	12.13	38.41	33.12	50.43	43.44	41.71	34.40
+WPE		2	6.08	0.67	5.70	-0.05	5.40	-0.39	5.66	0.00	24.65	11.59	38.64	32.14	50.29	43.87	42.39	34.05
		4	6.33	0.92	5.82	0.07	5.40	-0.39	5.67	0.01	23.50	11.26	37.22	32.00	49.34	42.34	**40.96**	**33.33**
		8	6.18	0.77	5.54	-0.21	5.04	-0.75	5.35	-0.31	23.49	11.43	37.89	32.67	50.12	43.59	41.37	**33.84**
MVDR-SA	74M	2	7.40	1.98	7.42	1.67	7.46	1.80	7.44	1.78	26.54	12.94	41.07	34.47	52.81	45.18	44.39	35.92
		4	7.94	2.52	7.98	2.23	7.99	2.19	7.99	2.33	27.15	12.63	41.35	34.92	52.72	44.94	44.76	35.78
		8	8.14	2.72	8.10	2.34	8.10	2.30	8.11	2.44	27.31	12.42	41.27	34.52	52.63	45.07	44.23	35.21
+WPE		2	7.75	2.34	7.48	1.73	7.48	1.69	7.55	1.89	26.40	12.78	41.09	33.66	52.00	44.89	44.29	35.54
		4	8.25	2.83	8.01	2.26	7.97	2.18	8.05	2.39	26.70	13.04	41.28	34.57	52.34	44.19	44.35	35.54
		8	8.40	2.99	8.09	2.34	8.03	2.24	8.14	2.48	26.35	12.93	41.03	33.72	52.12	44.26	44.12	35.37
FaSNet-SA	72M	2	10.21	4.79	9.85	4.09	9.56	3.76	**9.76**	**4.10**	26.86	11.33	40.91	35.67	52.57	47.12	44.57	36.24
		4	10.22	4.80	9.89	4.13	9.63	3.83	**9.82**	**4.15**	26.82	12.23	40.29	34.87	51.52	45.73	43.85	35.90
		8	10.41	4.99	10.01	4.25	9.72	3.92	**9.96**	**4.29**	26.53	10.73	39.93	34.78	51.70	45.28	44.11	35.51
+WPE		2	7.35	1.93	7.18	1.42	7.04	1.24	7.16	1.50	26.75	12.40	41.07	34.75	52.38	46.16	44.48	36.01
		4	6.48	1.07	6.26	0.51	6.13	0.34	6.24	0.58	25.86	12.63	39.27	33.15	51.78	44.16	43.29	34.97
		8	5.88	0.47	5.88	0.47	5.66	-0.13	5.85	0.19	26.16	10.78	39.35	34.89	51.22	45.25	43.39	35.41

visualize the spectrogram of one 8-channel Mixed AMI test chunk before and after beamforming (see Figure 5.3). It can be seen that, although FaSNet exhibits effective denoising, it also removes a portion of the speech signal, as highlighted by the white columns in the figure. On the contrary, DAS can preserve a significant portion of all speech signals while providing some denoising, which results in better speech and speaker recognition results.

5.3.2 Effectiveness of adding WPE in frozen beamformers

Table 5.1 also shows the performance differences for each system with or without WPE for dereverberation. First, even without beamforming, using WPE only during the inference phase for MC-SA-ASR results in a 0.21 dB improvement in SI-SDR and a slight absolute WER reduction of 0.27% (from 44.99% to 44.72%). For systems using beamformed signals, integrating WPE during the beamforming phase improves the SI-SDRi for DAS and

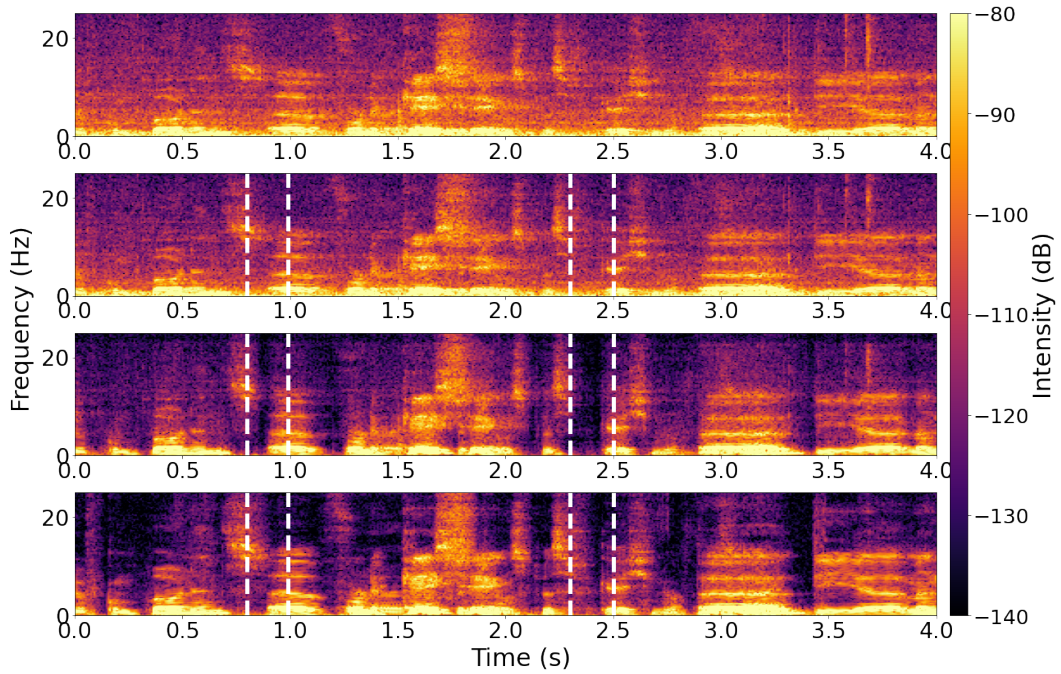


Figure 5.3: Spectrogram of one 8-channel Mixed AMI test chunk for comparison of beamformers. From top to bottom: 1st array channel; DAS beamformed signal; FaSNet beamformed signal; ground truth.

MVDR but not for FaSNet. However, using WPE during beamforming to fine-tune the SA-ASR model systematically improves ASR and speaker identification performance. Specifically, the DAS-SA-ASR system achieves a 3% relative reduction in WER (from 42.39% to 40.96%) and SER (from 34.29% to 33.33%) with WPE in the 4-channel setting. This demonstrates that WPE aids in reverberation reduction for fixed beamformers. However, the improvement in recognition performance for neural beamformers (MVDR and FaSNet) is less pronounced, likely because these beamformers have already learned to reduce reverberation during their training process.

5.3.3 Joint optimization of FaSNet and SA-ASR

Since fine-tuning SA-ASR with a frozen MVDR and FaSNet does not significantly improve SA-ASR performance, we conduct the joint optimization of SA-ASR and FaSNet, since FaSNet has relatively better performance and fewer parameters than MVDR. More specifically, we pretrain FaSNet for 0, 5, 10, or 50 epochs and subsequently fine-tune it for 15 epochs jointly with SA-ASR by backpropagating the SA-ASR loss.

The results in Table 5.2 show that, without WPE, joint optimization of FaSNet and SA-ASR (40.52%) reduces the WER by 9% relative to the frozen FaSNet (44.57%) and to SA-ASR (44.54%). We also observed a 7% relatively lower SER (33.68%) than using the frozen FaSNet (36.24%). However, the fine-tuned FaSNet exhibits a smaller SI-SDRi than the pretrained one. This indicates that joint training optimizes FaSNet for ASR performance rather than maximum noise and reverberation reduction at the cost of greater speech distortion. Furthermore, while the number of FaSNet pretraining epochs significantly impacts the SI-SDRi, it does not significantly impact the result of joint optimization, provided it's nonzero. This demonstrates the insensitivity of the joint optimization to the pretraining level of FaSNet.

Table 5.2: SI-SDR(%), SI-SDRi (%), WER (%) and SER (%) for jointly trained 2-channel FaSNet and SA-ASR models.

WPE usage	Mixed AMI test set					Real AMI test set							
	Pretrained			Fine-tuned		1-spk		2-spk		3-spk		1,2,3,4-spk	
	# Epo	SDR	SDRi	SDR	SDRi	WER	SER	WER	SER	WER	SER	WER	SER
None	0	5.66	0	-16.21	-21.87	25.31	11.37	38.04	32.28	48.63	43.07	41.71	**33.68**
	5	9.27	3.61	5.13	-0.53	24.91	13.06	36.84	33.54	47.49	45.00	**40.60**	34.87
	50	9.69	4.02	7.05	1.39	24.54	13.51	36.93	33.36	47.71	43.87	**40.52**	34.43
Fine-tune	0	5.87	0.21	-14.37	-20.03	25.51	11.75	38.58	32.03	50.23	42.53	43.41	**33.68**
	5	7.63	1.97	4.37	-1.29	25.04	12.59	37.46	31.16	48.05	41.59	41.00	**32.84**
	50	7.29	1.63	5.82	0.16	24.75	12.59	37.87	30.89	48.01	42.23	**40.92**	**33.14**
Test	50	7.29	1.63	7.29	1.63	25.11	11.90	38.03	31.73	48.61	42.52	41.46	33.33

We also visualize the beamformed signal before and after fine-tuning in Figure 5.4. FaSNet was pretrained for 5 epochs and then fine-tuned with SA-ASR. After fine-tuning, noise reduction was less effective compared to the pretrained FaSNet beamformed signal, which explains the SI-SDR decrease from 9.27 dB to 5.13 dB. However, examining the zone between the first two white lines, we find that the fine-tuned FaSNet beamformed signal repairs the over-suppression caused by the pretrained FaSNet while maintaining dereverberation compared to the 1st array channel. In the zone between the second set of white lines, the fine-tuned speech signal becomes more concentrated and enhanced compared to both the original and pretrained states. This demonstrates that fine-tuning FaSNet improves speech quality not by reducing noise, but by enhancing the intelligibility of the speech region.

Moreover, we tested different uses of WPE during the joint optimization phase. First, incorporating WPE during the fine-tuning optimization of beamforming does not result in better recognition performance. On the contrary, using WPE led to a 4% relative increase

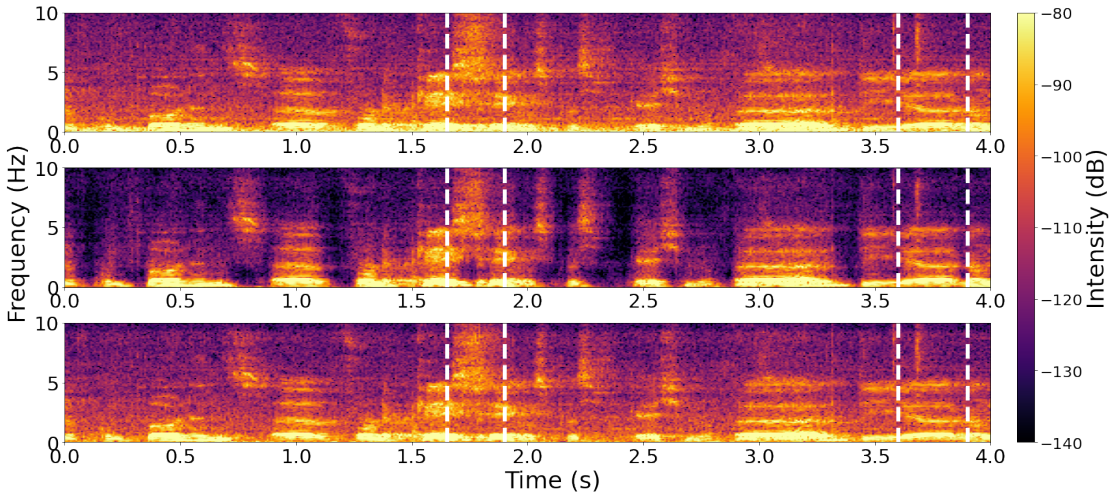


Figure 5.4: Spectrogram of one 8-channel Mixed AMI test chunk for comparison of pretrained and fine-tuned FaSNet beamformers. From top to bottom: 1st array channel; pretrained FaSNet beamformed signal; fine-tuned FaSNet beamformed signal.

in WER (from 41.71% to 43.41%) when training FaSNet from scratch with the training objectives of SA-ASR. Second, applying WPE to FaSNet after the joint optimization is completed also does not bring any benefits (41.46%). This indicates that the joint optimization of FaSNet and SA-ASR does not benefit from the dereverberation provided by WPE.

5.4 Summary

This chapter explored the integration of beamforming with SA-ASR for joint speech and speaker recognition of far-field meeting audio. We evaluated the impact of fine-tuning SA-ASR on the outputs of DAS, MVDR, or FaSNet beamformers and jointly fine-tuning SA-ASR with the FaSNet beamformer, and compared it with state-of-the-art MFCCA-based channel fusion. Experiments reveal that, in contrast to previously published results on simulated data, MFCCA is inefficient on real AMI data. This highlights the importance of systematically evaluating SA-ASR on real meeting data. Utilizing a DAS beamformer and jointly optimizing SA-ASR with the FaSNet beamformer lead to relative WER reductions of 8% and 9%, respectively. The use of WPE in the DAS-SA-ASR system can bring a 3% relative reduction in both WER and SER.

6 Real-life meeting transcription pipeline and its optimization

纸上谈兵终觉浅，
绝知此事要躬行。
In ink, we scheme, but wisdom's stream runs shallow,
To truly grasp, action alone we must follow.

The previous chapters introduced different architectures for processing synthetic or real meeting corpora that are already segmented.

While the model architecture is undoubtedly crucial, the application of the end-to-end model to real-life data has not received much attention. Due to the lack of significant amounts of real meeting-style training data, training of multi-speaker ASR models relies on simulated multi-speaker overlapped distant speech data. To improve performance on real test data, these models are adapted or fine-tuned on real training data. A few studies (Kanda et al., 2021c; Yang et al., 2023) have discussed training and evaluation on real recorded multi-speaker datasets, such as the AMI meeting corpus (Carletta et al., 2005) which still remains a challenging multi-speaker speech transcription task.

Training and inference on real long-length multi-speaker audio recordings is a non-trivial issue due to computational and memory requirements. One approach to address this issue is to segment the long recording at silence positions into disjoint segments (Kanda et al., 2021c). Another approach is to segment it into fixed-sized chunks and adjust the start/end times to non-overlapped word boundaries (Section 4.2.1). Both approaches rely on ground-truth annotations of the test set. In real-life applications, this information is not available and a Voice Activity Detection (VAD) system becomes essential for obtaining speech segments.

A second problem with training and inference on real multi-speaker audio is how to obtain the reference speaker embeddings (a.k.a. speaker profiles) used as inputs by the

SA-ASR system. In real-life applications, front-end Speaker Diarization (SD) becomes essential for this purpose. While studies on VAD (Medennikov et al., 2020; Yang et al., 2010) and SD (Park et al., 2022; Xiao et al., 2021) exist independently, there is little discussion on how to optimally integrate them into an SA-ASR pipeline to create a ready-to-use meeting transcription system. Speaker assignment in SA-ASR is typically performed using an x-vector-based speaker embedding model (Snyder et al., 2018). Preparation of a representative template for each speaker typically involves averaging the embeddings of candidate segments (Kanda et al., 2021b; Yu et al., 2022a). However, the selection of segments for better representing each speaker remains a question.

Another overlooked aspect is the synergy between the ASR and speaker modules within the SA-ASR system, which enhances both text and speaker prediction, particularly the former by leveraging the speaker information to assist ASR (Kanda et al., 2021b). However, there is a lack of research addressing the significance of speaker information in the ASR decoding process.

The contributions of this chapter include: (a) a VAD-SD-SA-ASR pipeline for real meeting transcription; (b) fine-tuning of SA-ASR on VAD output segments instead of ground-truth or fixed-sized segments to better fit the test conditions; (c) a discussion on strategies to improve speaker assignment in SA-ASR by leveraging various attributes related to VAD and SD, such as the number and length of segments used to obtain speaker profiles; (c) and a discussion on the role of the speaker module for aiding ASR decoding.

This chapter is organized as follows. Section 6.1 presents our VAD-SD-SA-ASR pipeline approach and individual modules. Section 6.2 presents our experimental settings, including the preparation of the training, development, and test data, followed by the discussion of results in Section 6.3. Finally, Section 6.4 provides a summary of the chapter. The major contents of this chapter have been published at Speaker Odyssey 2024 (Cui et al., 2024).

6.1 Proposed methods

6.1.1 System architecture

In real-life applications, extracting the speech parts and removing the silence segments is an essential step to ensure the quality of subsequent processing. Our proposed system consists of VAD, SD, and SA-ASR modules. Figure 6.1 illustrates the overall pipeline, which starts with VAD to divide the entire meeting into speech and non-speech segments. In a far-field multichannel setting, different speakers may be closer to different micro-

phones, resulting in varying volume levels across channels. VAD performed on single-channel speech can cause problems due to these imbalances, leading to inaccurate detection of speech segments as closer speakers are more reliably detected than those farther away. To address this, our system first applies a beamforming step to the AMI 8-channel MDM to produce a single-channel enhanced signal for VAD input.

Since SA-ASR requires speaker profiles as inputs, we need an SD model to determine the number of speakers in the entire meeting and identify the speech segments for each speaker. Since the performance of SD and SA-ASR is less sensitive to multichannel input compared to VAD, and because our experiments aim to validate the enrollment system and study other factors influencing performance, we chose to use SDM data for SD and SA-ASR to save running time. It should be noted that better performance can be achieved by using beamformed MDM. We use the results of VAD to perform SD on speech segments. Subsequently, based on the SD results, we use the non-overlapped portions of speech from each speaker to compute an average speaker embedding for that speaker. Finally, the segments obtained by VAD and the speaker profiles obtained through SD serve as input to SA-ASR, allowing us to retrieve the speech content of all different speakers in a first-in-first-out order.

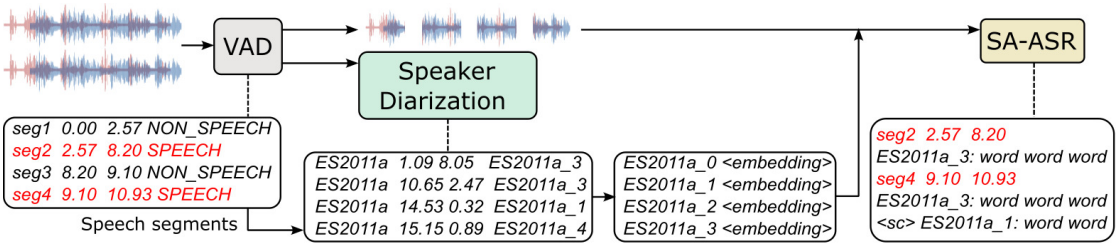


Figure 6.1: Proposed VAD-SD-SA-ASR Pipeline.

Voice activity detection

A VAD system identifies speech and non-speech segments in an audio signal, which is crucial for applications like speech recognition. The VAD system takes as input the single-channel signal obtained by delay-and-sum (DAS) beamforming applied to the AMI 8-channel MDM signal. The DAS beamformer is the same as the one used in Section 5.1.1. Neural network-based models for VAD often include architectures like Convolutional Neural Networks (Asif et al., 2022), Recurrent Neural Networks (Lin et al., 2023b), Long Short-Term Memory networks (Kim et al., 2016), etc. One of the common architectures is CRDNN (Convolutional, Recurrent, and Dense Neural Network) (Sainath et al., 2015b; Xiang et al., 2021). It combines convolutional layers for frequency variations, recurrent layers for tem-

poral modeling, and dense layers for feature mapping, which have succeeded in various speech-related tasks.

Extraction of speaker directory

As discussed in Section 2.1.2, classical SD based on speaker embeddings has a superior ability to handle various situations. Therefore, we chose to use the SD system based on x-vectors generated by ECAPA-TDNN (Emphasized Channel Attention, Propagation, and Aggregation in Time-Delay Neural Network) (Desplanques et al., 2020). Our SD system utilizes Spectral Clustering (Ning et al., 2006) as the clustering algorithm, and the number of speakers is determined by iterating over the AMI Dev set with speaker counts ranging from 1 to 9, selecting the number that results in the minimum loss. The diarization output provides the regions of speakers for generating embeddings.

Single-channel SA-ASR system

For a proof-of-concept, we employ the classical single-channel Transformer-based end-to-end Speaker-Attributed ASR (SA-ASR) (Kanda et al., 2021b) system. Details about this model can be found in Section 2.2.2.3. Following the SOT principle (Kanda et al., 2020b), the output is the concatenation of all speakers' sentences in first-in-first-out order, and each token is associated with one speaker ID.

6.1.2 Real data preparation

In real-life applications, ground-truth silence positions and utterance/word boundaries are not available at test time, so testing requires the use of VAD for segmentation. In Section 4.2.1, the SA-ASR model fine-tuned on fixed-sized segments and tested on VAD segments was found to yield poor results, which can be attributed to mismatched fine-tuning and test segment lengths. To adapt the model to the length of VAD segments during test, we also utilize VAD segments during the training phase. The following outlines the preparation of training, development, and test data on the AMI corpus. A similar approach can be used on other real corpora with sentence-level transcripts and timestamp annotations.

The process of obtaining speech segments, speaker, and text labels is the same for training, development, and test data. As shown in Figure 6.2, all meeting data is segmented into variable-sized speech segments using VAD. We merge adjacent speech segments separated by a silence shorter than a given duration threshold, resulting in an average segment length which depends on the specified threshold. Using AMI's annotation files with information about speaker segments and word boundaries, we assign text and speaker sequences

to each segment in a first-in-first-out order.

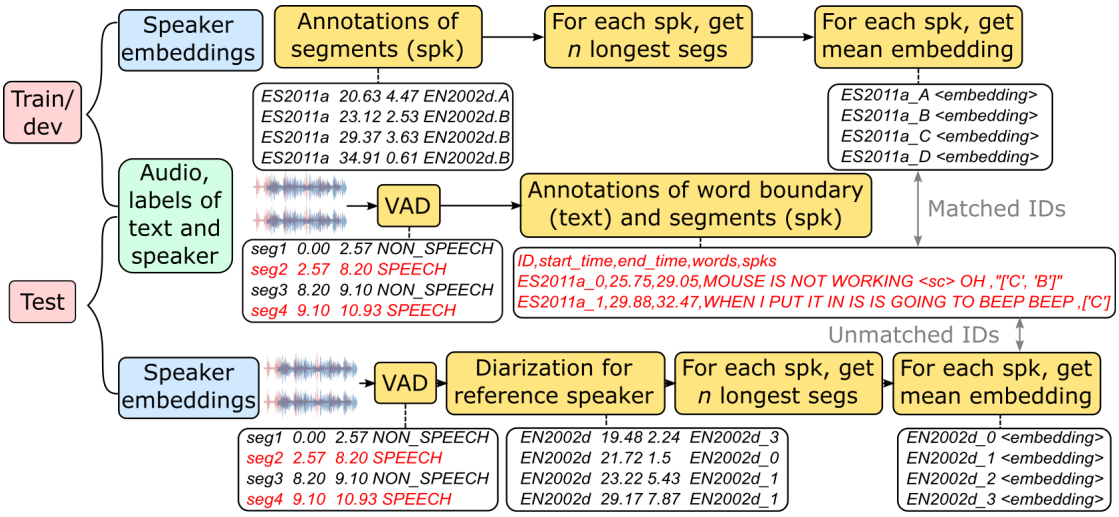


Figure 6.2: Preparation of the AMI corpus for the training, development, and test sets.

The process of obtaining speaker profiles differs for training and development vs. test. For the training and development sets, we extract all speech segments for each speaker in each meeting based on the ground-truth annotated speaker segments, and select the n longest segments for each speaker. We use a pretrained speaker embedding model to embed all n segments and calculate the average embedding as the template for the associated speaker. The process for the test set is different because, in real test scenarios, the number of speakers is not known a priori. To solve this issue, we apply the SD model to the VAD segments to estimate the number of speakers and the speech segments for each speaker. The speaker profiles for all speakers are then derived in the same way as for the training and development sets.

When computing speaker profiles for the training and development sets, the speaker IDs align with the labels used for speaker identification, as they both rely on the same speaker segment annotation file. However, during test, the SD results may predict a different number of speakers and assign new IDs to the recognized speakers. This necessitates remapping SD speaker IDs to ground-truth speaker IDs for evaluation purposes. This remapping process involves using the Hungarian algorithm Kuhn (1955), as implemented by Ravanelli et al. (2021), during the calculation of the diarization error rate.[1]

[1] https://github.com/speechbrain/speechbrain/blob/develop/tools/der_eval/md-eval.pl

6.2 Experiments on AMI

6.2.1 Datasets

AMI

We conducted fine-tuning and testing on the AMI corpus, which consists of approximately 100 h of multiple distant microphone (MDM) recordings with 3 to 5 speakers. The VAD module takes the DAS beamformed signal obtained from 8 MDM channels as input, while the SA-ASR module takes only the first MDM channel. We assessed three alternative segmentation methods. The first method involves fixed-sized segmentation with a chunk size of 5 s and a 2 s hop size, followed by adjusting the start/end times to non-overlapped word boundaries. The second method, following Kanda et al. (2021c) and Yang et al. (2023), involves segmenting based on ground-truth silence/non-speech positions between utterances: we extracted segments for all speakers from the ground-truth speaker and utterance boundary annotations, merged overlapped segments, and discarded the resulting segments which were longer than 100 s.

The third segmentation method uses VAD to extract the segments. We fused adjacent segments based on silence duration thresholds of 0.1, 0.3, 0.5, 0.7, and 0.9 s, resulting in datasets with different numbers of segments and different average segment lengths. Table 6.1 provides the statistics and average lengths for the training set under all segmentation methods.

Simulated multi-speaker LibriSpeech

To achieve optimal performance in AMI, it is imperative to undergo pretraining with a larger and cleaner dataset (Kanda et al., 2021c; Yang et al., 2023). We pretrained the SA-ASR model on distant-microphone multi-speaker data simulated using the LibriSpeech corpus (Panayotov et al., 2015). We employed the method described in Section 4.2.1 to generate single microphone signals containing 1 to 3 speakers. We utilized the train-960 and dev-clean subsets to construct our training and development datasets, creating a far-field microphone and multi-speaker set with 960 hours for training and 20 hours for development.

6.2.2 Model description

Our experiments were implemented using the SpeechBrain toolkit (Ravanelli et al., 2021). For VAD, we used the CRDNN model[2] pretrained on Libriparty (Ravanelli, 2023).

[2]Available at https://huggingface.co/speechbrain/vad-crdnn-libriparty

For SD, we use x-vectors as speaker embeddings and the Spectral Clustering (Ning et al., 2006) method. The SD and SA-ASR systems share a common speaker embedding model, ECAPA-TDNN[3], which is pretrained on the VoxCeleb1 (Chung et al., 2019) and VoxCeleb2 (Nagrani et al., 2020) training datasets, yielding 192-dimensional embeddings. Our text tokenizer is a SentencePiece model (Kudo and Richardson, 2018) with a vocabulary of 5,000 tokens. In SA-ASR, the Conformer-based encoder, the Transformer-based decoder, and the speaker decoder have 12, 6, and 2 layers, respectively. All multi-head attention mechanisms have 4 heads, the model dimension is set to 256, and the size of the feedforward layer is 2,048. The number of parameters of the employed CRDNN, ECAPA-TDNN, and SA-ASR models is 0.1M, 22.15M and 61.1M, respectively.

6.2.3 Training setup

The ASR module in SA-ASR was pretrained on simulated multi-speaker Librispeech data for 360k iterations using the Adam optimizer with a learning rate of 5×10^{-4}. The ASR and speaker modules in SA-ASR were then further pretrained for 300k iterations using a learning rate of 2.5×10^{-4}. Finally, we fine-tuned the model on real AMI single distant microphone (SDM) data for 20k iterations, using the Adam optimizer with a learning rate of 3×10^{-4}. We tuned only the ASR module for the first 10k iterations and performed joint tuning of the ASR and speaker modules in the last 15k iterations. Those numbers of iterations were fixed in preliminary experiments based on the development set WER.

6.2.4 Metrics

We evaluate the SA-ASR system's ASR and speaker assignment performance using the word error rate (WER), the token-level speaker error rate (SER) and speaker accuracy as defined in Section 4.2.3.

For all the tables of results in this chapter, we employed the SCTK toolkit (NIST, 2024) to conduct significance tests, specifically the Matched Pair Sentence Segment test. We highlight in bold the best WER/SER result and the results statistically equivalent to it at a 0.05 significance level.

[3] Available at https://huggingface.co/speechbrain/spkrec-ecapa-voxceleb

6.3 Evaluation results

6.3.1 Efficiency of fine-tuning on VAD segments

In practical applications, it is necessary to divide long continuous meeting audio into smaller-sized segments. As detailed in Section 6.2.1, our experiments involve three alternative segmentation methods. When fine-tuning the SA-ASR system, we employed different training data segmented using these three distinct methods. To emulate real-life test conditions, the primary focus is on testing all models using VAD output segments. Table 6.1 showcases the results on VAD segments with 0.5 s silence threshold, consistent with the subsequent tables. To gain a more comprehensive understanding of the model's capabilities across various speaker counts, we report the results on subsets comprising 1 to 3 speakers. For comparison, we also report the performance of each model on a matched test segmented in the same way as the training set.

Table 6.1: WER and SER (%) on the AMI test set using different segmentation methods for SA-ASR fine-tuning and testing. The speaker profiles are computed by averaging the embeddings obtained from the 3 longest annotated segments. Number of segment and average value of test VAD 1,2,3,4-spk are 3,586 and 6.96 s. The Dur indicates the average duration in seconds.

Train/ dev seg method	Train set statistics			Test (matched) 1,2,3,4-spk				Test (VAD seg with 0.5 s silence threshold) 1-spk		2-spk		3-spk		1-4-spk	
	Sil. thr.	#Seg	Dur	#Seg	Dur	WER	SER	WER	SER	WER	SER	WER	SER	WER	SER
Fixed	-	90,426	6.59 s	10,234	6.63 s	44.54	28.54	49.41	28.92	50.91	29.78	57.48	36.29	55.91	34.94
Oracle	-	21,917	10.19 s	2,504	8.67 s	**43.42**	**21.81**	44.47	23.08	50.74	28.21	56.77	31.47	54.00	31.02
VAD	0.1 s	93,150	1.59 s	10,605	1.59 s	44.62	29.03	**30.53**	17.12	45.67	29.15	53.04	38.07	47.02	30.19
	0.3 s	57,589	3.40 s	6,095	3.71 s	44.50	28.85	31.95	16.17	43.54	28.99	52.38	35.69	46.33	28.99
	0.5 s	35,091	6.18 s	3,586	6.96 s	45.09	27.46	31.25	15.53	42.05	23.21	50.57	30.54	**45.09**	27.46
	0.7 s	22,027	10.33 s	2,265	10.24 s	45.80	24.82	32.13	**13.51**	**41.74**	**22.17**	**49.82**	**27.00**	46.13	27.02
	0.9 s	14,460	15.65 s	1,519	13.15 s	47.08	24.95	33.05	**13.35**	42.52	**22.31**	51.39	**27.30**	46.27	**25.04**

Looking at the last column in Table 6.1, models fine-tuned on VAD segments consistently exhibit superior performance when tested on VAD segments compared to models fine-tuned on fixed-sized or ground-truth segments. The relative WER reduction can be up to 19% (from 55.91% to 45.09%), and the relative SER reduction can be as high as 28% (from 34.94% to 25.04%). However, this difference is not as pronounced when models are tested on their respective matched test sets. This underscores the significance of training on data that closely aligns with the test data type. For real-life applications relying on VAD segments, it is advisable to fine-tune on VAD segments, even when speaker segment annotations are

available in the training set. It is noteworthy to highlight that, even with comparable average segment lengths (10.19 s and 10.33 s), fine-tuning on VAD segments can result in a 15% (from 54% to 46.13%) lower relative WER and a 13% (from 31.02% to 27.02%) lower relative SER compared to the model fine-tuned on ground-truth segments.

Focusing on the models fine-tuned on VAD segments in Table 6.1, it can be seen that the model fine-tuned on segments with 0.5 s VAD silence threshold exhibits a lower WER, which can be attributed to the matched fine-tuning and test segment lengths. However, the SER shows a different trend. Specifically, as the average fine-tuning segment length becomes longer, the corresponding SER decreases. The SER exhibits a relative reduction of 22% (from 17.12% to 13.35%), 23% (from 29.15% to 22.31%), and 28% (from 38.07% to 27.30%) for 1, 2, and 3 speakers, respectively, when transitioning from an average fine-tuning length of 1.59 s to 15.65 s. This indicates that, as the training segments become longer, SA-ASR demonstrates improved capability in assigning speaker identities.

Table 6.2 reports the speaker counting accuracy of the three systems. First of all, the assignment of speaker labels solely based on SD is not accurate, particularly for multi-speaker scenarios. This underscores the critical role of SA-ASR systems. For the comparison of SA-ASR models, on the 1-speaker test subset, the model fine-tuned on VAD segments exhibits the highest accuracy at 94.76%, compared 80.01% for the model fine-tuned on fixed-sized segments. However, the latter system demonstrates slightly higher accuracy when the number of speakers is 2 (69.01%) or 3 (50.04%). The model fine-tuned on ground-truth segments performs better when the number of speakers is as high as 4 (29.50%). Future studies can explore how to employ different training techniques for datasets with specific numbers of speakers.

So far, we have used annotated speaker segments to extract the speaker profiles in the training, development, and test sets. In a real-life situation, using SD segments to extract the speaker profiles in the test set is desirable. However, this leads to statistical differences between the fine-tuning and test phases. To understand how this affects the SER, we evaluate both types of templates in the test set. The resulting performance when fine-tuning and testing on VAD output segments with 0.5 s silence threshold is presented in Table 6.3.

Surprisingly, while the model is fine-tuned on speaker profiles extracted from annotated speaker segments, extracting speaker profiles from SD segments during test reduces the SER by up to 16% relative (from 15.53% to 12.97%). We attribute this to the fact that human segmentation is partly inaccurate, especially when marking the boundaries of each speaker's speech segments. By contrast, SD can offer more precise speech boundaries, especially for the longest non-overlapped segments which are utilized to compute the speaker

Table 6.2: Speaker counting accuracy (%) on the AMI test set with 0.5 s VAD silence threshold. We mapped the speaker number from SD output to have an evaluation without SA-ASR.

	Method	# Speakers	Estimated # speakers				
			1	2	3	4	>4
Without SA-ASR	SD	1	89.20	10.27	0.53	0.00	0.00
		2	59.18	35.37	5.34	0.11	0.00
		3	29.73	48.65	19.46	2.16	0.00
		4	10.24	39.37	33.07	14.57	2.76
SA-ASR Train/dev seg method	Fixed-size (5 s)	1	80.04	17.28	2.29	0.32	0.05
		2	14.09	**69.01**	16.28	0.54	0.00
		3	0.73	38.93	**50.04**	9.18	0.91
		4	0.40	15.98	56.96	20.90	4.50
	Ground-truth	1	89.00	9.66	1.22	0.10	0.00
		2	20.41	63.53	15.28	0.54	0.10
		3	2.54	35.39	48.63	12.15	0.90
		4	0.00	11.47	54.50	**29.50**	3.27
	VAD (0.5 s sil. thresh)	1	**94.76**	4.91	0.26	0.05	0.00
		2	21.24	67.68	10.07	0.87	0.00
		3	2.72	40.29	47.00	8.89	0.72
		4	0.00	11.57	59.91	23.55	3.71

Table 6.3: SER (%) on the AMI test set with 0.5 s VAD silence threshold as a function of the speaker profiles used for fine-tuning and testing. Ref: templates extracted from annotated speaker segments. Test: templates extracted from SD output. The WER is unaffected and equal to that in Table 6.1 (1-spk 31.25%, 2-spk 42.05%, 3-spk 50.57%, 1,2,3,4-spk 45.09%).

Train/dev	Test	1-spk	2-spk	3-spk	1,2,3,4-spk
Ref	Ref	15.53	23.21	30.54	27.46
Ref	Test	**12.97**	**20.76**	**27.43**	**26.54**

profiles.

Surprisingly too, these improved speaker profiles do not affect the WER, despite the fact that the derived token-level weighted speaker profiles are improved and fed back to the ASR Decoder (see Figure 6.1). This suggests that this feedback mechanism proposed in Kanda et al. (2021b) may not fulfil its goal of enabling the Speaker Decoder to assist the ASR Decoder.

6.3.2 Improving speaker assignments by better creating speaker profiles

Additionally, we explored the extraction of speaker profiles from varying segment lengths and with different numbers of candidate segments. To begin with, the impact of using VAD with 0.1 s to 0.9 s silence threshold as input segments to the SD system is analyzed in Table 6.4. For each SD result, we filtered candidate segments for each speaker by considering different length intervals. The mean embedding for each speaker was then calculated from all segments in that length interval. The results showcased in Table 6.4 reveal several good solutions with either short or long candidate segment lengths. When candidate segment lengths fall within the range of 2 to 5 s, VAD silence thresholds below 0.5 s are preferable to longer ones, with a relative SER reduction of up to 12% (12.79% compared to 14.58%). On the contrary, when using longer candidate segments ranging from 6 to 50 s, short VAD silence thresholds are inadequate and VAD silence thresholds above 0.5 s are preferable.

Table 6.4: SER (%) on the AMI test set with 0.5 s VAD silence threshold by varying the length of candidate segments used to extract non-overlapped speaker profiles. The WER is unaffected and equal to that in Table 6.1.

VAD segments		2–5 s			5–10 s			6–50 s		
Sil. thresh	Avg dur	1-spk	2-spk	3-spk	1-spk	2-spk	3-spk	1-spk	2-spk	3-spk
0.1 s	1.59 s	**12.79**	**20.62**	**27.44**	60.53 †	54.96 †	†	†	†	†
0.3 s	3.40 s	13.09	20.85	**27.04**	14.16	21.76	28.08	19.03 †	26.18 †	32.48 †
0.5 s	6.18 s	**12.83**	20.93	27.59	13.88	21.82	27.97	**12.97**	**20.76**	**27.43**
0.7 s	10.33 s	14.26	22.46	27.71	14.17	22.53	27.83	**12.76**	20.98	27.71
0.9 s	15.65 s	14.58	22.16	27.84	**12.94**	21.53	**27.49**	14.11	21.75	27.81

†: The VAD output segments with 0.1 s or 0.3 s silence threshold are inadequate for obtaining enough candidate segments with the specified length, resulting in unavailable or abnormal speaker assignment results.

We further examined how different configurations, including the presence of overlapping segments, the number of candidate segments, and the length of candidate segments, impact the speaker profiles and the resulting SER. Table 6.5 illustrates the influence of these attributes using VAD segments with 0.5 s silence threshold as inputs to SD. Using all candidate segments instead of the 2 or 10 longest leads to an enhanced speaker representation irrespective of the segment length, as evidenced by a lower SER up to 15% relative (from 15.02% to 12.83%). Moreover, shorter candidate segments exhibit a greater sensitivity to the number of segments. This can be attributed to the fact that a larger number of segments

enables a more robust and comprehensive representation of each speaker. The presence of overlapping segments among the candidate segments does not appear to have a significant impact.

Table 6.5: SER (%) on the AMI test set with 0.5 s VAD silence threshold by varying the number and the length of candidate segments used to extract speaker profiles. The WER is unaffected and equal to that in Table 6.1.

Candidate segments			SER		
Overlap	# Seg	Dur	1-spk	2-spk mix	3-spk mix
no	2	2–5 s	15.02	23.72	29.75
no	10	2–5 s	14.44	21.92	27.82
no	all	2–5 s	**12.83**	**20.93**	27.59
yes	all	2–5 s	**12.80**	**20.81**	**27.28**
no	2	5–10 s	13.09	21.29	28.00
no	10	5–10 s	14.03	21.69	27.76
no	all	5–10 s	13.88	21.82	27.97
yes	all	5–10 s	13.93	21.72	27.92
no	2	6–50 s	13.04	21.42	27.53
no	10	6–50 s	13.33	21.08	**27.13**
no	all	6–50 s	**12.97**	**20.76**	**27.43**
yes	all	6–50 s	13.06	**20.81**	**27.41**

Using segment annotations, we compute the number of overlapping clips involving varying numbers of speakers. Subsequently, for each speaker count, we calculate the quantity of overlapping clips across different durations of overlap. Figure 6.3 presents statistical findings across the complete AMI corpus, revealing a minimal occurrence of 2, 3, and 4 overlapping speaker regions.

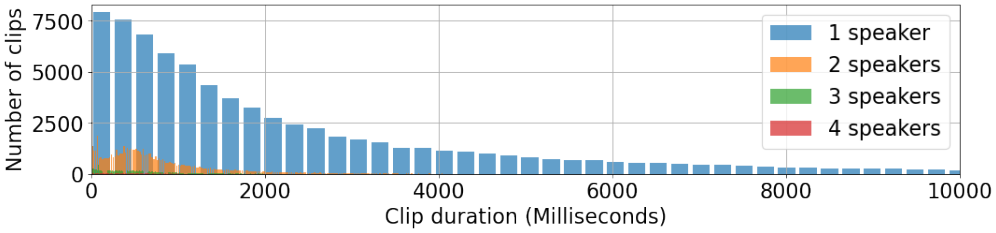


Figure 6.3: Histograms of speaker overlap on the AMI corpus. Only clips shorter than 10 s are shown.

Figure 6.4 depicts the cosine similarity between the speaker embeddings of the candidate segments in one meeting for the three specified segment lengths. Segments in the range of 6 to 50 s exhibit a higher similarity compared to those in the range of 2 to 5 s, leading to a slightly lower SER (19.62% vs. 19.88%).

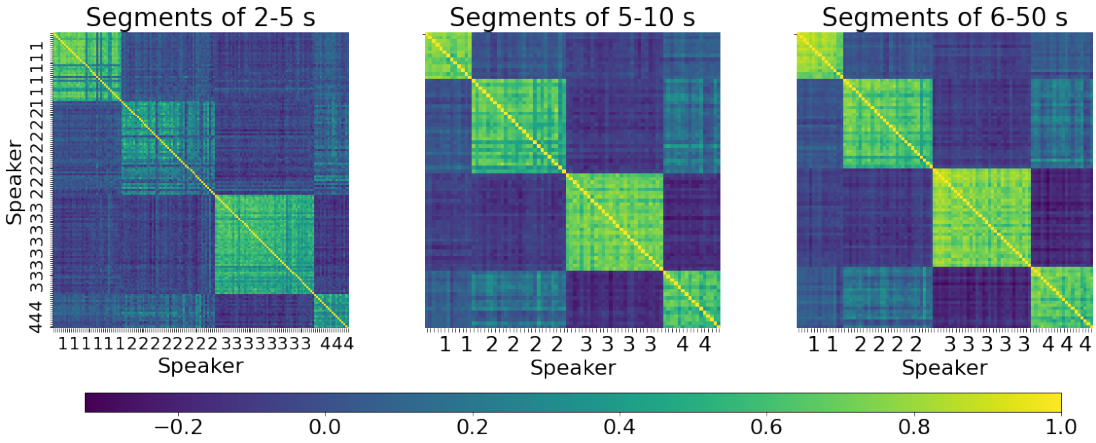


Figure 6.4: Similarity matrices between the speaker embeddings of the candidate segments in meeting ES2004c. With the three specified segment lengths, the number of candidate segments is 154, 81, and 96, and the resulting SER for that meeting is 19.88%, 19.65%, and 19.62%, respectively.

6.3.3 Impact of the speaker module in SA-ASR

Through the findings presented in Section 6.3.1 and Section 6.3.2, we have observed a consistent pattern: regardless of modifications made to the speaker profile for a given SA-ASR model, the WERs remain unchanged. This contradicts the expected behavior of the SA-ASR architecture. As illustrated in Equation (2.8), the speaker decoder's output, denoted as q_n, is utilized to derive a posterior probability $\hat{s}_n$. Subsequently, a weighted speaker profile is computed by weighting S with $\hat{s}_n$. This resulting embedding, denoted as $\bar{s}_n$, is then fed into the first layer of the ASR decoder's feed-forward network, as depicted in Equation (2.13). Hence, in theory, variations in the speaker profile S should influence the output of the ASR decoder.

We conducted further analysis to investigate the impact of speaker profiles on ASR output. Specifically, we compared the results obtained from the same pretrained SA-ASR system while varying the speaker probability $\hat{s}_n$. Three types of $\hat{s}_n$ were compared: (a) utilizing the original probability calculated by the model $\hat{s} \in \mathbb{R}^{N \times K}$ where N is the decoder state and K the speaker number, (b) setting it as a baseline with all values set to $1/K$ [4], and (c) employing an upper bound approach by using an oracle $\hat{s}_n$. The original speaker profile was generated based on segments ranging from 6 to 50 seconds of SD output using a 0.5-second silence threshold VAD. In the oracle approach, the active speaker region is assigned

[4]Note that only the probabilities used for the ASR decoder are set to $1/K$; the original probabilities for speaker identification remain unchanged.

a value of 1, while the rest are set to 0:

$$s_n^{\text{oracle}} = \begin{cases} 1 & \text{if } k_n = \text{oracle\_label}_n \\ 0 & \text{otherwise.} \end{cases} \tag{6.1}$$

Table 6.6 presents the results obtained by employing three different speaker probability settings. When setting this value to 0, the ASR module disregards speaker information during decoding, resulting in a marginal increase in WER by 3% relative across the entire test dataset (from 45.09% to 46.63%). However, utilizing the oracle value yields no discernible change in WER across subsets with varying speaker counts. This observation suggests that, within this architecture, the ASR decoder has already been optimized to effectively utilize speaker information for text decoding based on the tested speaker probabilities. Nevertheless, the slight difference between using a value of $1/K$ and the oracle value also indicates that the assistance provided by the speaker decoder to the ASR decoder is quite limited. Thus, a more optimal architecture is required to enhance the interaction between ASR and speaker identification.

Table 6.6: WER (%) and SER (%) on AMI by using SA-ASR trained on VAD 0.5 s.

Speaker probability	1-spk		2-spk mix		3-spk mix		1,2,3,4 -spk mix	
	WER	SER	WER	SER	WER	SER	WER	SER
Original	**31.25**	12.97	**42.05**	20.76	**50.57**	27.43	**45.09**	26.54
$1/K$	31.26	12.87	43.57	21.53	51.63	27.64	46.63	25.12
Oracle	**31.25**	**10.85**	**42.05**	**13.84**	**50.57**	**18.45**	**45.09**	**17.22**
Oracle (SPK eos)	**31.25**	**5.55**	**42.05**	**9.06**	**50.57**	**13.85**	**45.09**	**12.39**

Concerning the SER, while the SERs of oracle results exhibit notable decreases compared to the original results, they remain quite large. Upon analyzing the SA-ASR output, we observed that this improvement comes from providing the oracle speaker probability for identification. However, the end-of-sentence token is often recognized too early, causing both the ASR and speaker decoding processes to stop once this token has been recognized. This underscores the necessity of utilizing speaker decoding information to better assist ASR, particularly in predicting sentence endings more accurately.

Subsequently, we compared different decoding methods: since both the ASR and speaker outputs are at the token level and are mapped accordingly, we can opt to terminate the decoding process upon the appearance of either the ASR end-of-sentence token or the end of speaker prediction. When using the original speaker probability value, accurately predicting the end of the speaker label is often challenging, so we typically rely on the

ASR end-of-sentence token as a termination indicator for decoding. However, with the oracle value, it becomes feasible to leverage the speaker label length information to make better decisions about when to halt the decoding process. We conducted additional tests using both the ASR end-of-sentence token and the end of speaker prediction from the oracle label as a double-check threshold to terminate the decoding. While this reduced the SER to 12.39%, the SER did not reach 0%, primarily due to the additional wrong tokens recognized by ASR, necessitating additional wrong speaker labels, which is caused by the late prediction of the ASR end-of-sentence token. The comparison between the two oracle SERs (17.22% and 12.39%) highlights that in ASR decoding, early predictions of the end-of-sentence token occur more frequently than late predictions (see Table 6.7), underscoring the importance of enhancing ASR by compensating for overlooked aspects in the transcription process. Another potential avenue is to train the model to predict speaker end-of-sentence labels to aid the decoding process further.

Table 6.7: Examples of early and late prediction of ASR end-of-sentence token and its influence on speaker prediction

	Ground-truth	**Early eos prediction**	**Late eos prediction**
ASR (word-level)	according to (eos)	according (eos)	according to to (eos)
speaker (token-level)	1,1,1,1	1,1,1	1,1,1,1,1

6.4 Summary

In this chapter, we focused on enhancing speaker assignment in SA-ASR for real-life meetings. We proposed a VAD-SD-SA-ASR pipeline and advocated for fine-tuning on VAD segments instead of fixed-size segments. We showed that this can lead to a relative SER reduction up to 28%. We then explored strategies for extracting speaker profiles by varying the VAD and SD configurations. Our results reveal that extracting them from SD output rather than annotated speaker segments results reduces the SER by up to 16% relative. Moreover, while extracting speaker profiles from short or long segments can yield a similar SER, the latter results in improved speaker representation and speaker similarity across segments and is less sensitive to the number of segments averaged to obtain the template. Finally, we have found that improved speaker profiles do not affect the WER. Especially, not utilizing speaker information for ASR decoding results in only a 3% relative increase in WER. This suggests that an effective feedback mechanism enabling the Speaker Decoder to assist the ASR Decoder is still to be found.

7 Conclusion and perspectives

长风破浪会有时，
直挂云帆济沧海。
There will come a time when the long winds break the waves. And I will set my cloud-like sails to cross the vast seas.

Chinese proverb

This thesis tackled the challenges of multichannel meeting transcription, aiming to determine who said what and when. This involved addressing several tasks such as speech separation, speaker diarization, and ASR. The main challenge lies in how to integrate multiple modules to achieve better performance. In this chapter, we conclude the thesis with a summary of the proposed methods in Section 7.1 and explore future research perspectives in Section 7.2.

7.1 Conclusion

In Chapter 3, we introduced a transcription pipeline that includes multichannel separation, single-channel single-speaker ASR, and speaker identification models. We controlled the overlap ratio to study its influence. To train the separation model on real meeting data, we proposed a data alignment and generation method for the AMI meeting corpus. Experimental results demonstrate that the speaker overlap ratio in the training data for the separation model and the fine-tuning data for the ASR and speaker embedding models significantly influences system performance. On synthetic AMI data, varying the overlap ratio from 0% to 75% results in an 88% relative increase in WER and a 61% relative increase in SER. This considerable degradation demonstrates that the ASR and speaker identification performance is highly sensitive to the overlap ratio. Results on real AMI data show that increasing the variety of training data can improve model robustness. Specifically, training on both synthetic and real AMI data reduces WER by 18% relative compared to training only

on real AMI data. Additionally, separation models trained on data with higher speaker overlap are more robust in real-life conditions. This indicates that real AMI data, being more complex, benefits from a separation model trained under more challenging conditions to improve robustness in real-life scenarios.

In Chapter 4, we proposed an end-to-end MC-SA-ASR system that combines a Conformer-based encoder with multi-frame cross-channel attention and a speaker-attributed Transformer-based decoder. We conducted two studies on input features for our MC-SA-ASR: the first compared the impact of using Mel filterbank features vs. magnitude+phase features, and the second examined the influence of using a data-invariant beamformer to enhance original channels from specific angular sectors. Experimental results demonstrate that, on simulated data, our proposed MC-SA-ASR system achieves relative reductions in WER of up to 12% and 16% compared to existing single-channel and multichannel methods, respectively. On real-world data, our model achieves a relative reduction in WER of up to 6% compared to SC-SA-ASR. This suggests that leveraging speaker information can improve the performance of multi-speaker ASR in a multichannel setting. We also demonstrated that using magnitude+phase features in settings with a greater number of channels and/or speakers results in better performance compared to solely using Mel filterbank features. This can be explained by the fact that, as the number of speakers in a room increases, the positional information of the speakers has a greater impact on ASR performance. Finally, we have demonstrated that models trained on data-invariant beamformed data can improve MC-ASR performance up to 29% relative in WER. This opens up new possibilities for enhancing input features for multichannel input. Moreover, using beamformed signals from a larger number of microphones results in better performance for the MC-ASR model. This is because more microphones allow the filter for a certain sector to incorporate more information, thereby providing better enhancement of signals from that sector.

In Chapter 5, we explored the integration of beamforming with SA-ASR for joint speech and speaker recognition of far-field meeting audio. We evaluated the impact of fine-tuning SA-ASR on the outputs of DAS, MVDR, and FaSNet beamformers, which represent typical fixed, hybrid (neural and fixed), and fully neural beamforming methods, respectively. For each pipeline, we also examined the influence of WPE for its dereverberation effect. Additionally, we jointly fine-tuned SA-ASR with the FaSNet beamformer using the training objectives of SA-ASR. These pipelines were compared with state-of-the-art MFCCA-based channel fusion. Experiments reveal that, contrary to previously published results on simulated data, MFCCA is inefficient on real AMI data. Specifically, utilizing a

DAS beamformer led to relative WER reductions of 8% compared to MC-SA-ASR, highlighting the importance of systematically evaluating SA-ASR on real meeting data. We discovered that using frozen FaSNet as a front-end beamformer does not improve the SA-ASR performance compared to a DAS beamformer. The reason is that, although FaSNet exhibits effective denoising, it also removes a portion of the speech signal. On the contrary, DAS beamforming preserves a significant portion of all speech signals while providing some denoising, which results in better speech and speaker recognition results. Experiments with WPE demonstrate that WPE aids in reverberation reduction for fixed beamformers. However, the improvement in recognition performance for neural beamformers (MVDR and FaSNet) is less pronounced, likely because these beamformers have already learned to reduce reverberation during their training process. Finally, we demonstrated that jointly optimizing SA-ASR with the FaSNet beamformer led to relative WER reductions of 9%, and that joint training optimizes FaSNet for ASR performance rather than maximum noise and reverberation reduction at the cost of greater speech distortion.

The previous chapters presented methods on segmented data using previously collected speaker profiles. Note that all our methods require a directory of reference speaker embeddings to conduct the speaker identification task. In Chapter 6, we focused on the overall meeting transcription pipeline by proposing a VAD-SD-SA-ASR pipeline and enhancing speaker assignment in SA-ASR for real-life meetings. For the fine-tuning process, we advocated for fine-tuning on VAD segments instead of fixed-size segments, demonstrating that this approach can lead to a relative SER reduction up to 28%. This highlighted the necessity of aligning the training conditions with the test conditions, especially when the training data is limited. We then explored strategies for extracting speaker profiles by varying the VAD and SD configurations. Our results reveal that extracting speaker profiles from SD output rather than annotated speaker segments reduces the SER up to 16% relative. Moreover, while extracting speaker profiles from short or long segments can yield similar SERs, the latter results in improved speaker representation and speaker similarity across segments and is less sensitive to the number of segments averaged to obtain the template.

7.2 Future perspectives

7.2.1 Integration of speech separation and MC-SA-ASR

Comparing the experimental results in Chapters 3 and in 4 for the 2-speaker 2-channel test set on the AMI corpus using comparable fine-tuning settings shows that the

ASR performance of the method using separation (41.67%) does not improve compared to the MC-SA-ASR model (39.54%). However, the method using separation has better speaker identification performance, with 12.67% SER compared to 24.83%. This demonstrates that, on the one hand, the errors caused by the separation model can propagate to the ASR stage and degrade its performance. Hence, using an end-to-end SA-ASR method can yield better performance for overlapped speech recognition. On the other hand, predicting speaker identity using the enrolled separated speech segments results in better outcomes than using the speaker decoder, which provides a token-level prediction using the Transformer architecture with a larger number of parameters. This leads us to consider how to integrate these two methods for improved recognition performance.

Along this line, Kanda et al. (2023) proposed an architecture that first performs MVDR-based separation on the input mixture, separating it into two channels, each containing an individual speaker's speech. This approach is based on the assumption that scenarios where three speakers talk simultaneously are extremely rare. Following separation, the channels are fed to a two-channel Transformer transducer architecture that produces an output similar to SA-ASR: transcriptions of each speaker in a first-in-first-out order. Although this system lacks a speaker embedding module to predict speaker identity, it provides a new perspective on combining a speech separation module with our MC-SA-ASR. In this setup, speaker prediction shifts from the token level to the channel level, as each channel contains separated speech.

Kanda et al. (2023) performed online recognition. The limitation of this approach is the inability to identify different speaker IDs at different instances and the inability to process scenarios with more than two speakers. Moreover, to realize this idea in an offline setting, several questions need to be considered. First, it is possible for three speakers to appear in the same segment, which necessitates a method for determining the number of channels required. Second, the separation model must then be able to handle a variable number of speakers. On this topic, several studies have been proposed, such as those by Isik et al. (2016), Takahashi et al. (2019), and Zeghidour and Grangier (2021).

7.2.2 Improving speaker change token prediction in SA-ASR

In Chapter 3, on the AMI test set, the speaker counting accuracy for 1, 2, 3, and 4 speakers is 92.40%, 77.05%, 50.74%, and 20.04%, respectively. This accuracy degrades rapidly as the number of speakers in the segments increases, indicating that while MC-SA-ASR performs better than SC-SA-ASR and MC-ASR, it still has limitations in accurately determining the number of speakers in scenarios involving three or more participants.

Kanda et al. (2019b) proposed a joint decoding framework for overlapped speech recognition and speaker diarization, where speaker embedding estimation and target-speaker ASR were performed alternately. More recently, Shi et al. (2023a) proposed a context-aware SA-ASR system that utilizes a context-dependent scorer to model speaker discriminability by contrasting with speakers in the context.

Several perspectives are possible in this regard. First, the assignment of speaker labels to an utterance can be based on the speaker change labels in the ASR output. We can jointly consider speaker output for the decision of adding speaker change labels. Specifically, if the first part of an utterance has most of its tokens with speaker ID A, and the rest of the sentence has speaker ID B, it is reasonable to add a speaker change label at the switching position. Second, as discussed in Chapter 6, the collection of speaker profiles for SA-ASR necessitates an SD model. Since the SD output includes a report on speaker activity, we could enhance the SA-ASR output by leveraging the diarization output. This can be done by weighting the predictions from both the SA-ASR and the SD model, thus incorporating the diarization results into the final transcription.

7.2.3 Advanced end-to-end architecture with stronger integration of ASR and speaker information

in Chapter 6, we found that using oracle speaker probabilities does not improve the WER compared to using real probabilities. Additionally, not utilizing speaker information for ASR decoding results in only a 3% relative increase in WER. This suggests that, within the current architecture, the ASR decoder has already been optimized to effectively utilize speaker information for text decoding based on the tested speaker probabilities. The next step would be to design an effective feedback mechanism that enables the Speaker Decoder to assist the ASR Decoder. This perspective highlights the need to explore new strategies for integrating speaker information more effectively to enhance ASR performance further.

One of the possible directions would be to implement a joint encoder for both ASR and speaker feature extraction. Currently, the SA-ASR system employs a Conformer-based ASR encoder and an ECAPA-TDNN-based speaker encoder. While these encoders are efficient for their specific tasks, they both rely on Mel filterbank features. This suggests the possibility of a joint feature extraction mechanism for both ASR and speaker tasks. Such a design could enable tighter information fusion between the ASR and speaker blocks.

Another approach could be multi-modal feature extraction. Different speakers may have different intents, so integrating a joint language model within the system to predict speaker identity based on textual information could be beneficial. This could lead to more

accurate and contextually aware predictions, enhancing the overall performance of the system.

7.2.4 Adaptation to other languages

In this thesis, the SA-ASR system is trained on English data, but such a pretrained model can be adapted to another language with limited labeled data. This is particularly useful because not every language has sufficient real meeting data for training purposes. In the SA-ASR model, the speaker block primarily relies on a speaker embedding model, which is mostly independent of the language. However, the ASR component of the model is language-dependent.

To adapt a pretrained English ASR model to another language, transfer learning is an effective method. Transfer learning for data adaptation from one language to another has been extensively studied in ASR. For instance, Kunze et al. (2017) proposed adapting a pretrained ASR model to the German language by freezing several parameters. Another solution is to develop a multilingual model. Inaguma et al. (2019) proposed a language-independent ASR model using an external language model, which is integrated into the decoder of the sequence-to-sequence ASR model.

Despite these advancements, there has been limited research on transfer learning in the context of multi-speaker ASR systems. This presents a promising area for future study. Exploring transfer learning techniques in multi-speaker ASR could lead to significant improvements in adapting such systems to various languages, thereby expanding their applicability and performance across different linguistic contexts.

8 Résumé étendu

8.1 Introduction

Dans le monde d'aujourd'hui, où tout va très vite, prendre des notes manuellement pendant une réunion peut être chronophage et laborieux. Imaginez une solution où l'enregistrement de la réunion est automatiquement transcrit en notes complètes, éliminant ainsi le besoin de prendre des notes manuelles. Les outils de transcription de réunions sont devenus populaires dans les réunions en ligne, en utilisant la technologie de reconnaissance automatique de la parole (ASR) pour convertir le dialogue parlé en texte écrit.

Sur le plan académique, les chercheurs travaillent à améliorer la performance de l'ASR dans les conditions acoustiques difficiles des réunions en face-à-face qui comportent du bruit, de la parole superposée et de la réverbération, afin de rendre les modèles d'ASR plus robustes. Les conférences sur le traitement de la parole proposent fréquemment des défis sur ce sujet, tels que le *Multi-Channel Multi-Party Meeting Transcription Challenge* (M2MeT) (Yu et al., 2022b) à ICASSP 2022 et la série de défis CHiME (Barker et al., 2013; Vincent et al., 2013; Barker et al., 2017; Vincent et al., 2016; Barker et al., 2018; Watanabe et al., 2020; Cornell et al., 2023). L'intérêt combiné des milieux industriel et académique entraîne des avancées significatives (Boeddeker et al., 2018; Chang et al., 2019; Sklyar et al., 2021; Kanda et al., 2021b).

Notre objectif dans cette thèse est de générer des transcriptions de réunions qui indiquent avec précision qui a dit quoi et quand, en se basant sur les signaux enregistrés par une antenne de microphones. Les principaux défis incluent la reconnaissance de la parole et des locuteurs en présence de bruit ambiant, de parole superposée et de réverbération.

8.2 Contexte

La transcription de réunions utilisant des microphones distants est complexe. Pour améliorer les résultats, de nombreuses études ont utilisé des modules de séparation multicanale ou de formation de voies pour extraire le signal de chaque locuteur et le traiter par un module d'ASR (Yoshioka et al., 2018b; Chen et al., 2019; Kanda et al., 2019a). Cepen-

dant, les erreurs de séparation affectent le module d'ASR. Des études plus récentes (Chang et al., 2019; Li et al., 2021b; Wu et al., 2021; Zhang et al., 2021; Shi et al., 2022) ont proposé de rétropropager la fonction de coût du module d'ASR vers le module de séparation en utilisant un critère d'apprentissage invariant à la permutation (PIT) pour optimiser les deux modules conjointement, mais ces systèmes sont souvent limités à un nombre fixe de locuteurs.

Pour résoudre ce problème, des systèmes d'ASR multi-locuteurs et de diarisation de bout-en-bout ont émergé pour les enregistrements mono et multicanaux (Guo et al., 2021; Lu et al., 2021; Sklyar et al., 2021; Kanda et al., 2021b; Chang et al., 2019, 2020; Scheibler et al., 2023; Shi et al., 2023b). Kanda et al. (2021b) ont notamment proposé un système d'ASR attribuée aux locuteurs (SA-ASR) invariant au nombre de locuteurs et basé sur un modèle Transformer utilisant l'apprentissage de sortie sérialisée (SOT) (Kanda et al., 2020b), qui partage les représentations de la parole et des locuteurs entre l'ASR et la diarisation. Contrairement aux approches monocanales, les approches multicanales (Chang et al., 2019, 2020; Scheibler et al., 2023) ne partagent pas ces représentations. L'approche MC-WD-SOT de Shi et al. (2023b) fusionne les informations d'ASR et de locuteur et utilise l'attention multicanale multi-trame (MFCCA) (Yu et al., 2023) dans l'encodeur d'ASR pour assister l'identification des locuteurs.

En outre, les études antérieures sur la transcription de bout-en-bout des réunions ont principalement exploré l'architecture des modèles et ont été évaluées sur des données simulées. L'apprentissage et l'inférence sur des longs enregistrements multi-locuteurs réels posent des défis en termes d'empreinte calculatoire et exploitent souvent la vérité terrain pour segmenter les enregistrements aux instants de silence (Kanda et al., 2021c). En situation réelle, cette vérité terrain n'est pas disponible, ce qui rend un système de détection d'activité vocale (VAD) indispensable pour obtenir les segments de parole. De plus, obtenir les plongements de locuteur de référence pour le système SA-ASR nécessite une diarisation des locuteurs (SD), un sujet peu exploré en termes d'intégration optimale avec la VAD dans un pipeline de SA-ASR (Medennikov et al., 2020; Yang et al., 2010; Park et al., 2022; Xiao et al., 2021).

8.3 Séparation de la parole, ASR et identification des locuteurs conjointes

Dans la transcription de réunions multi-locuteurs, il est courant d'utiliser un modèle de séparation pour extraire les signaux de parole individuels afin d'améliorer la qualité de la

parole, puis d'employer un module d'ASR (Yoshioka et al., 2018b; Chen et al., 2019; Kanda et al., 2019a). Cependant, apprendre un modèle de séparation tel que *Filter-and-Sum Network* (FaSNet) (Luo et al., 2019) sur des données réelles est difficile en raison du manque de signaux de vérité terrain. De plus, déterminer le taux de superposition optimal de la parole dans l'ensemble d'apprentissage reste un défi. Nous présentons ici un pipeline avec séparation multicanale, ASR, et identification des locuteurs. Notre système constitue la toute première approche conjointe pour fine-tuner ASR et plongements de locuteur d'une manière PIT. Nous proposons des méthodes pour générer et augmenter les données d'apprentissage pour des réunions réelles, et évaluons l'impact du taux de superposition sur l'apprentissage du modèle de séparation et le fine-tuning des modèles ASR et d'identification des locuteurs sur des corpus synthétiques et réels.

8.3.1 Méthodes proposées

Architecture du système

Comme illustré dans la Figure 8.1, le pipeline proposé se compose de trois modules : séparation des locuteurs, ASR, et identification des locuteurs. Les entrées du système sont constituées des données audio multicanales et multi-locuteurs ainsi que des plongements des locuteurs de référence. Le modèle de séparation traite ces entrées pour produire des signaux mono-canaux avec un seul locuteur, ce qui améliore la qualité de la parole en réduisant la parole superposée, le bruit et la réverbération. Le modèle de séparation est suivi d'un modèle d'ASR monocanal et d'un modèle de plongement de locuteur opérant en parallèle. Nous identifions les locuteurs en effectuant le produit scalaire entre les plongements extraits et les plongements de référence. Les sorties du système sont les transcriptions et les identités des locuteurs présents dans le mélange.

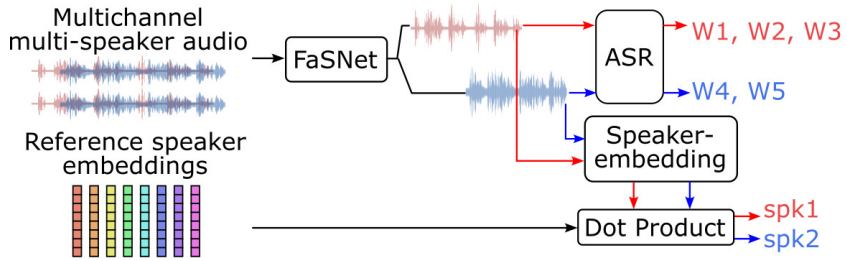


Figure 8.1: Système proposé : FaSNet, ASR et identification des locuteurs conjointes.

Nous utilisons le système FaSNet pour le modèle de séparation, qui estime directement les filtres dans le domaine temporel. Pour gérer la permutation des canaux et permettre au modèle de traiter un nombre variable de microphones, nous adoptons le paradigme

Transform-average-concatenate (TAC) (Luo et al., 2020a). Nous employons un modèle d'ASR avec un encodeur basé sur Conformer (Gulati et al., 2020) et un décodeur basé sur Transformer (Karita et al., 2019). Pour encoder les informations relatives aux locuteurs, nous utilisons des x-vectors générés par le modèle ECAPA-TDNN (Desplanques et al., 2020).

Alignement des données pour apprendre le modèle FaSNet

Nous générons des signaux de microphones en champ lointain et les signaux de microphones-casques alignés correspondants en trois étapes (voir la Figure 8.2) : (a) extraction des segments de parole non superposée ; (b) alignement par filtrage optimal et découpage en clips ; (c) création de mélanges en champ lointain et utilisation des clips de microphones-casques alignés comme vérité terrain.

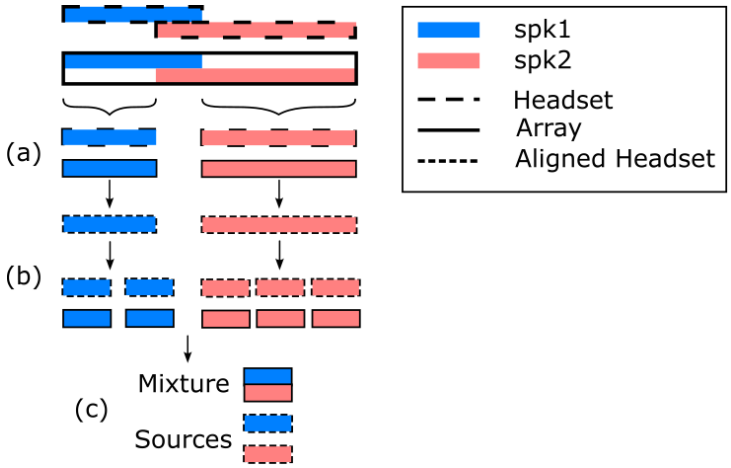


Figure 8.2: Génération des signaux de vérité terrain pour des données de réunions réelles.

Les filtres optimaux $f_{ij}(t)$ dans l'étape (b) sont calculés au sens des moindres carrés en résolvant l'équation

$$\min_{f_{ij}} \sum_{t} (f_{ij} \star h_j(t) - x_i(t))^2 \tag{8.1}$$

où $h_j(t)$ et $x_i(t)$ représentent respectivement le signal de microphone-casque du locuteur j et le signal capté par le microphone i, et $\star$ désigne la convolution dans le domaine temporel. La solution est obtenue classiquement en utilisant le filtre de Wiener à réponse impulsionnelle finie (FIR)[1].

[1] https://en.wikipedia.org/wiki/Wiener_filter

8.3.2 Protocole expérimental

Jeu de données

Nous validons l'efficacité du pipeline proposé sur le corpus de réunions AMI (Carletta et al., 2005). Pour apprendre le modèle FaSNet, nous appliquons la méthode de la Section 8.3.1 pour générer des données de réunions AMI à deux canaux et deux locuteurs. Après avoir séparé les données avec FaSNet, les modèles d'ASR et de plongement de locuteur sont fine-tunés et évalués sur les données séparées. Nous adoptons une approche de segmentation inspirée des « groupes de parole » (Kanda et al., 2021c) qui fonctionne comme suit : (a) Segmenter chaque réunion en utilisant une taille de segment de b secondes et une taille de pas de o secondes. (b) Si l'instant de début/fin d'un segment tombe dans une région contenant plus d'un locuteur, il est ajusté pour être 2 s en dehors de la région de superposition. (c) Si l'instant de début/fin d'un segment tombe au milieu d'un mot, il est ajusté pour s'aligner avec le début/la fin de ce mot. Nous avons réalisé des expériences avec une taille de segment de 5 s et une taille de pas de 2 s. Afin de faciliter l'étude de l'influence du taux de superposition sur la performance, nous avons sélectionné des taux de superposition de 0%, 25%, 50% et 75% pour générer des données AMI mélangées et synthétiques à deux locuteurs.

Modèles et apprentissage

L'implémentation de FaSNet-TAC est tirée de l'outil Asteroid (Pariente et al., 2020). Pour le modèle d'ASR, nous utilisons un modèle pré-entraîné basé sur Conformer mono-canal[2]. Le modèle de plongement de locuteur est un ECAPA-TDNN également pré-entraîné[3]. Le modèle FaSNet est appris sur AMI mélangé pendant 200 époques avec l'optimiseur Adam, un pas d'apprentissage de 10^{-3} et *early-stopping*. Le modèle d'ASR est fine-tuné sur AMI mélangé pendant 15 époques avec Adam à 3×10^{-4} puis avec le modèle de plongement de locuteur pendant 5 époques. Pour le fine-tuning sur AMI mélangé et AMI réel combinés, le modèle d'ASR est fine-tuné pendant 7 époques et le modèle de plongement de locuteur pendant 3 époques.

Métriques

Nous évaluons la performance de séparation en utilisant le *scale-invariant signal-to-distortion ratio* (SI-SDR) et son amélioration (SI-SDRi) en dB sur le jeu de test AMI mélangé. Pour la performance d'ASR, nous utilisons le taux d'erreur de mots (WER), et pour l'identification des locuteurs, le taux d'erreur de locuteurs (SER), ce dernier étant calculé en divisant

[2]Disponible sur `https://huggingface.co/speechbrain/asr-conformer-transformerlm-librispeech`

[3]Disponible sur `https://huggingface.co/speechbrain/spkrec-ecapa-voxceleb`

le nombre de phrases erronées par le nombre total de phrases.

8.3.3 Résultats

Résultats avec différents taux de superposition sur AMI synthétique

La Table 8.1 montre les résultats sur le jeu de test. Tout d'abord, l'augmentation du taux de superposition de 0% à 75% entraîne une réduction de 3 dB du SI-SDRi (de 12,61 dB à 9,61 dB), et une dégradation relative de 88% du WER (de 28,31% à 53,42%) et de 61% du SER (de 6,65% à 10,73%). Cette dégradation considérable montre que la performance est très sensible au taux de superposition. En particulier, les augmentations relatives en WER sont de 18% (de 28,31% à 33,61%), 24% (de 33,61% à 41,69%) et 28% (de 41,69% à 53,42%) à chaque incrément de 25% du taux de superposition. Cela montre que la dégradation de l'ASR est plus rapide que l'augmentation du taux de superposition.

Table 8.1: SI-SDR (dB), SI-SDRi (dB), WER (%) et SER (%) obtenus par le pipeline FaSNet sur AMI mélangé / synthétique avec deux canaux et deux locuteurs mélangés en fonction du taux de superposition des locuteurs dans l'ensemble de données d'apprentissage de FaSNet et de fine-tuning de l'ASR et du plongement de locuteur.

Taux superposition apprentissage		**Test AMI mélangé**			**Test AMI synthétique**	
FaSNet	**ASR + locuteur**	**Superposition**	**SI-SDR ↑**	**SI-SDRi ↑**	**WER ↓**	**SER ↓**
0%	0%	0%	**10,07**	**12,61**	28,31	6,65
25%	0%	0%	9,99	12,53	**26,89**	**6,60**
	25%	25%	8,92	11,46	33,61	8,05
50%	0%	0%	9,90	12,45	27,49	7,74
	25%	25%	9,01	11,56	33,42	8,00
	50%	50%	8,26	10,80	41,69	10,05
75%	0%	0%	8,71	11,25	32,64	10,33
	25%	25%	8,17	10,72	37,85	10,24
	50%	50%	7,64	10,18	46,88	10,47
	75%	75%	7,06	9,61	53,42	10,73

Résultats sur AMI réel avec PIT

La Table 8.2 montre les résultats sur AMI réel. Tandis que la Table 8.1 indique que l'utilisation d'un modèle de séparation entraîné sur un taux de superposition spécifique peut améliorer la performance de test d'un tel taux, la Table 8.2 révèle une tendance différente. Avec un taux de superposition en fine-tuning allant de 21,03% à 25%, l'utilisation d'un modèle de séparation entraîné avec 50% ou 75% de superposition donne de meilleurs résultats que l'utilisation de données avec 0% ou 25% de superposition. En particulier, le modèle entraîné avec 50% de superposition obtient une réduction relative de 5% du WER (de 36,24%

à 34,42%) par rapport au modèle entraîné avec 25% de superposition. Cela montre que les données AMI réelles, étant plus complexes, bénéficient d'un modèle de séparation entraîné dans des conditions plus difficiles pour améliorer la robustesse dans des scénarios réels.

Table 8.2: SI-SDR (dB), SI-SDRi (dB), WER (%) et SER (%) obtenus par le pipeline FaSNet sur AMI réel 2 canaux 2 locuteurs, lorsqu'il est entraîné sur AMI synthétique et réel en utilisant PIT, en fonction du taux de superposition des locuteurs dans l'ensemble de données d'apprentissage de FaSNet.

Taux superposition pour apprentissage de FaSNet	Test AMI mélangé			Test AMI réel		
	Superposition	SI-SDR	SI-SDRi	Superposition	WER	SER
0%	25%	7,68	10,22	21,03%	36,84	10,87
25%	25%	8,92	11,46	21,03%	36,24	**9,23**
50%	25%	**9,01**	**11,56**	21,03%	**34,32**	10,65
75%	25%	8,17	10,72	21,03%	35,46	10,23

8.3.4 Conclusion

Nous avons développé un pipeline de transcription de réunions composé de séparation multicanale, ASR, et identification des locuteurs. Les résultats montrent que le taux de superposition des locuteurs à l'apprentissage impacte fortement la performance : sur AMI synthétique, il augmente le WER de 88% et le SER de 61% relatifs lorsque la superposition augmente de 0% à 75%. Sur AMI réel, combiner des données synthétiques et réelles réduit le WER de 18% par rapport à un apprentissage uniquement sur AMI réel. Les modèles appris avec plus de superposition montrent une meilleure robustesse aux conditions réelles.

8.4 Reconnaissance automatique de la parole multicanale attribuée aux locuteurs de bout-en-bout

Un inconvénient de l'approche décrite dans la Section 8.3 est que les erreurs de séparation peuvent se propager au module ASR. Des solutions (Chang et al., 2019; Wu et al., 2021; Shi et al., 2022) ont été proposées pour rétropropager le coût d'apprentissage de l'ASR pour tous les locuteurs vers le module de séparation en utilisant un critère PIT pour optimiser conjointement les deux modules. Cependant, le système ne peut souvent gérer qu'un nombre fixe de locuteurs. Pour relever ce défi, nous proposons un système SA-ASR multicanal (MC-SA-ASR) composé d'un encodeur basé sur Conformer avec MFCCA, un encodeur de locuteur et un décodeur Transformer attribué aux locuteurs.

8.4.1 Méthodes proposées

Architecture du système

Le système MC-SA-ASR proposé est illustré dans la Figure 8.3. Il s'inspire du modèle SA-ASR monocanal (SC-SA-ASR) de Kanda et al. (2021b) et utilise des plongements de locuteur pour guider le décodeur d'ASR. Au lieu de l'encodeur monocanal, notre système utilise une version modifiée de l'encodeur d'ASR multicanal basé sur Conformer avec MFCCA. Les décodeurs d'ASR et de locuteur restent les mêmes que dans le modèle SC-SA-ASR.

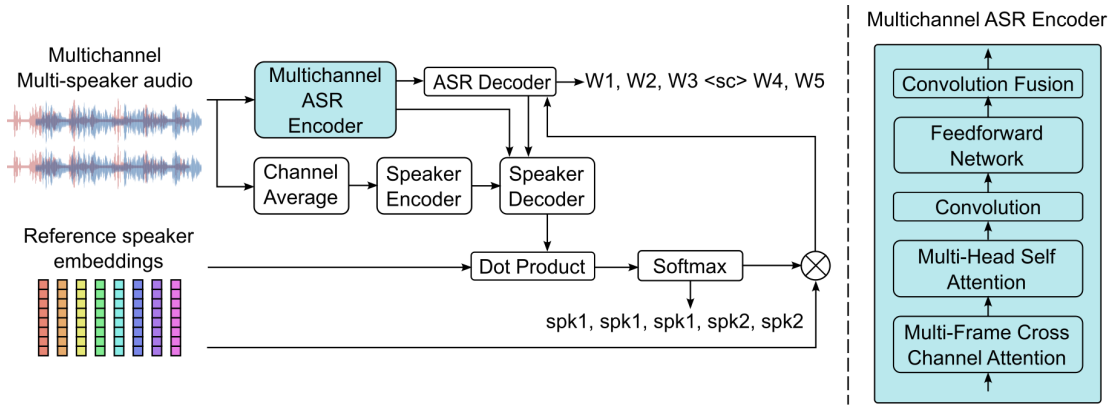


Figure 8.3: Aperçu du système MC-SA-ASR proposé (gauche) et de son encodeur (droite).

La dernière couche de l'encodeur est une couche de fusion par convolution. Elle sert de couche de sortie de l'encodeur d'ASR multicanal (voir la Figure 8.3). Cette couche combine les représentations correspondant aux multiples canaux d'entrée. L'étude de Yu et al. (2023) présente le mécanisme de fusion pour 8 canaux. Nous l'étendons pour prendre en charge des entrées à 2, 3 ou 4 canaux comme illustré dans la Figure 8.4.

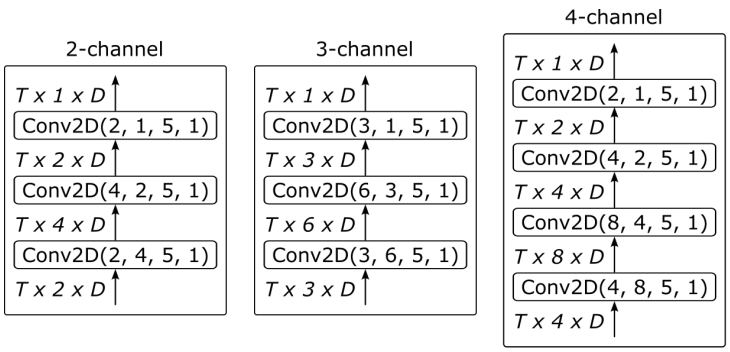


Figure 8.4: Fusion par convolution multicanal pour des entrées de 2, 3 et 4 canaux.

Les caractéristiques de banc de filtres Mel sont d'abord moyennées sur tous les canaux, puis transmises à notre modèle de plongement de locuteur. Nous utilisons un modèle de plongement de locuteur x-vector basé sur ECAPA-TDNN. Pour adapter les dimensions des x-vectors à notre architecture, nous remplaçons la couche finale de moyennage par une couche linéaire.

Caractéristiques d'entrée

Nous considérons deux ensembles alternatifs de caractéristiques d'entrée. D'une part, nous calculons des caractéristiques de banc de filtres Mel de M dimensions à partir uniquement de l'amplitude de la transformé de Fourier à court terme (STFT). D'autre part, nous concaténons l'amplitude de la STFT et le cosinus et le sinus de sa phase, chacun ayant une dimension G, pour créer une représentation de dimension $3 \times G$ appelée caractéristique *magnitude+phase*. Ces caractéristiques sont ensuite traitées spécifiquement, comme illustré dans la Figure 8.5 et décrit ci-après.

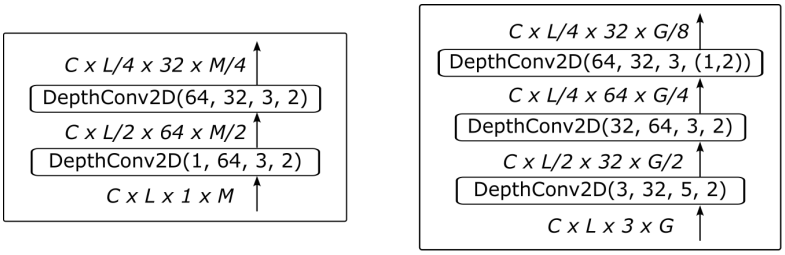


Figure 8.5: Extraction des caractéristiques par convolution 2D en profondeur (dimension d'entrée, dimension de sortie, taille du noyau, pas) pour les caractéristiques de banc de filtres Mel (gauche) et magnitude+phase (droite).

Comme alternative ou complément aux caractéristiques spatiales, nous proposons de prétraiter les données par formation de voies invariante aux données, de manière à ce que chaque canal résultant capte les signaux d'un secteur angulaire spécifique. Le filtre de formation de voies pour chaque secteur angulaire est calculé à partir des I microphones, de sorte que le nombre de microphones et le nombre de secteurs angulaires sont deux concepts différents et peuvent varier de manière indépendante. Par la suite, nous comparons trois entrées différentes pour le MC-SA-ASR, comme illustré dans la Figure 8.6 : les signaux originaux à 4 canaux provenant de 4 microphones, les signaux issus de la formation de voies à 4 canaux extraits de 4 microphones, et les signaux issus de la formation de voies à 4 canaux extraits de 8 microphones.

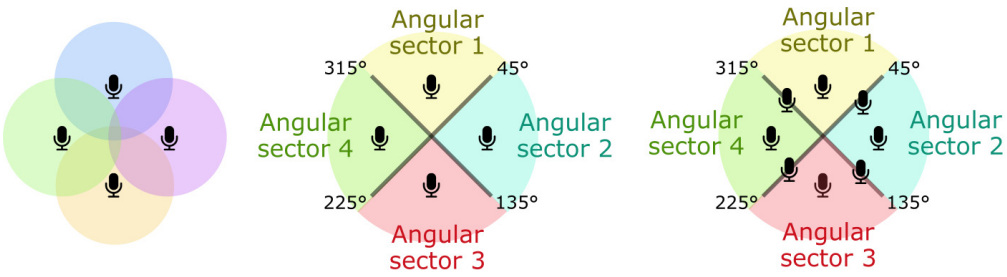


Figure 8.6: Canaux de microphones d'origine (gauche), 4 canaux issus de la formation de voies à partir de 4 microphones (centre), et 4 canaux issus de la formation de voies à partir de 8 microphones (droite).

8.4.2 Protocole expérimental

Jeu de données

Nous simulons des scénarios multi-locuteurs en utilisant LibriSpeech (Panayotov et al., 2015). Les sous-ensembles train-960, dev-clean et test-clean sont utilisés pour générer respectivement nos jeux de données d'apprentissage, de développement et de test. Nous considérons une antenne circulaire de 2 à 8 microphones. Chaque signal multi-locuteurs est généré en tirant aléatoirement une phrase de 1 à 3 locuteurs et associé à 8 plongements de locuteurs, englobant à la fois les locuteurs présents et des locuteurs sélectionnés aléatoirement.

En utilisant la méthode de segmentation décrite dans la Section 8.3.2, nous avons mené des expériences sur le corpus AMI avec des tailles de segment de 5, 10 ou 15 s et un pas de 2 s. Afin d'évaluer la performance du modèle sur des ensembles de données avec un nombre variable de locuteurs, nous avons combiné tous les jeux de test segmentés en 5, 10 et 15 s, puis les avons divisés en quatre jeux de test en fonction du nombre de locuteurs. Le nombre de segments pour chaque nombre de locuteurs est le suivant : 5 737 segments avec 1 locuteur, 4 911 segments avec 2 locuteurs, 3 986 segments avec 3 locuteurs et 1 936 segments avec 4 locuteurs.

Modèles et apprentissage

Pour tous les modèles (SC-SA-ASR, MC-ASR et MC-SA-ASR) dans nos expériences comparant les caractéristiques banc de filtres Mel et magnitude+phase, l'encodeur a 12 couches et le décodeur a 1 couche. Le décodeur de locuteur dans SC-MC-ASR et MC-SA-ASR comporte 2 couches. Pour les expériences comparant les canaux originaux avec les canaux issus de la formation de voies, nous utilisons un modèle MC-ASR avec 12 couches dans l'encodeur et 6 couches dans le décodeur. L'ECAPA-TDNN est le même modèle pré-

entraîné que dans la Section 8.3.

Tous les modèles ont été appris jusqu'à convergence sur LibriSpeech multi-locuteurs simulé. Les modules ASR de SC-SA-ASR et MC-SA-ASR ont été pré-entraînés pendant 80 époques, suivis du fine-tuning de SC-SA-ASR pour 60 époques et du modèle MC-ASR pour 140 époques avec un décodeur à 1 couche et 120 époques avec un décodeur à 6 couches.

Métriques

Nous utilisons le WER pour évaluer la tâche d'ASR, ainsi que le taux d'erreur de locuteur au niveau des tokens (T-SER) et le taux d'erreur de locuteur au niveau des phrases (S-SER) Kanda et al. (2020a) pour évaluer la tâche d'identification des locuteurs.

8.4.3 Résultats

Résultats de MC-SA-ASR utilisant les caractéristiques par banc de filtres Mel ou amplitude+phase

La Table 8.3 présente les résultats obtenus par les modèles de base (SC-SA-ASR et MC-ASR) et le modèle proposé (MC-SA-ASR) sur les données simulées de LibriSpeech multi-locuteurs. Tout d'abord, en comparant les résultats de SC-SA-ASR et MC-SA-ASR, nous pouvons conclure que l'utilisation d'un encodeur multicanal pour traiter les informations de parole à partir de plusieurs microphones améliore la performance de reconnaissance vocale. Plus précisément, le modèle MC-SA-ASR utilisant des caractéristiques de banc de filtres Mel atteint un WER de 14,77% dans les scénarios à 4 canaux, soit une réduction relative de 12% par rapport au modèle SC-SA-ASR (16,81%). Deuxièmement, le modèle MC-SA-ASR proposé, qui intègre les informations de locuteur dans le décodeur d'ASR, obtient une réduction relative de 16% du WER par rapport à MC-ASR (18,03%). Cela suggère que l'exploitation des informations spatiales peut améliorer la performance de l'ASR multi-locuteurs dans un enregistrement multicanal.

Les WER de test obtenus par le modèle MC-SA-ASR proposé avec des caractéristiques d'entrée de banc de filtres Mel et avec des caractéristiques d'entrée amplitude+phase ne présentent pas de différences significatives. Cependant, nous pouvons constater que dans le cas de l'ASR multicanale multi-locuteurs, les caractéristiques de phase atteignent de meilleurs résultats sur les segments contenant un plus grand nombre de locuteurs. Cela peut s'expliquer par le fait qu'à mesure que le nombre de locuteurs dans une pièce augmente, l'information spatiale des locuteurs a un impact plus important sur la performance de l'ASR.

Résultats de MC-ASR avec formation de voies invariante aux données

Table 8.3: WER (%), S-SER (%) et T-SER (%) des différents systèmes sur le jeu de test LibriSpeech multi-locuteurs multicanal simulé. Les résultats sont groupés par nombre de locuteurs dans le mélange simulé. La colonne 1,2,3-loc' montre les résultats obtenus sur un jeu de test contenant des mélanges de 1, 2 et 3 locuteurs.

Système	Carac	#Cn	1-loc			2-loc			3-loc			1,2,3-loc			#Prm
			WER	S-SER	T-SER	WER	S-SER	T-SER	WER	S-SER	T-SER	WER	S-SER	T-SER	
Bases															
SC-SA-ASR	Mel	1	7,15	1,64	3,13	14,68	3,23	5,42	20,39	**5,59**	7,67	16,81	**3,95**	6,18	61,1M
MC-ASR	Mel	2	8,52	-	-	16,76	-	-	22,87	-	-	18,21	-	-	27,8M
		3	8,29	-	-	16,54	-	-	22,01	-	-	18,03	-	-	
		4	8,11	-	-	16,43	-	-	21,76	-	-	17,84	-	-	
Proposés															
MC-SA-ASR	Mel	2	**6,64**	**1,15**	2,84	14,21	3,19	**5,14**	19,03	5,92	7,34	15,41	4,34	6,46	51.6M
		3	**6,62**	**1,11**	**2,73**	14,24	3,15	5,56	18,92	6,11	7,29	15,25	4,12	6,16	
		4	7,17	1,41	2,95	14,09	**3,06**	**5,10**	18,70	6,06	**6,93**	**14,77**	**4,05**	6,40	
	Mag+Phase	2	7,24	1,53	3,30	15,42	3,85	6,22	20,34	7,06	7,98	16,76	4,87	7,29	
		3	6,86	1,78	3,39	**13,69**	3,72	5,39	18,14	6,86	7,83	15,04	4,08	6,28	
		4	6,91	1,86	3,43	**13,97**	**3,10**	5,85	**18,03**	6,69	7,26	**14,69**	4,12	**5,85**	

Nous comparons ensuite les résultats du modèle MC-ASR entraîné sur les canaux originaux à celui entraîné sur les canaux issus de la formation de voies. La Table 8.4 présente les résultats sur le jeu de données de test LibriSpeech en utilisant un modèle pré-entraîné et sur le jeu de données de test AMI en utilisant un modèle fine-tuné à partir du modèle pré-entraîné correspondant. Sur le jeu de données de test AMI, les modèles 4-4 (entraîné sur 4 canaux issus de la formation de voies à partir de 4 microphones) et 8-4 (entraîné sur 4 canaux issus de la formation de voies à partir de 8 microphones) montrent une amélioration significative par rapport au modèle 4- (entraîné sur 4 canaux originaux). Ces améliorations soulignent l'efficacité du fine-tuning sur les signaux issus de la formation de voies pour mieux exploiter l'information de phase. De plus, le modèle 8-4 surpasse systématiquement le modèle 4-4 à la fois sur les jeux de test LibriSpeech et AMI. Cela démontre que l'utilisation de signaux issus de la formation de voies provenant d'un plus grand nombre de microphones améliore la performance du modèle MC-ASR.

8.4.4 Conclusion

Dans cette section, nous avons présenté un système de d'ASR multicanale attribuée aux locuteurs de bout-en-bout. Les résultats expérimentaux montrent que, sur des données simulées, notre approche réalise des réductions relatives du WER allant jusqu'à 12% et 16% par rapport aux méthodes monocanale et multicanale existantes. Enfin, nous avons

Table 8.4: WER (%) sur les canaux originaux ou sur les canaux issus de la formation de voies du modèle MC-ASR (27.8M) entraîné sur LibriSpeech synthétique et fine-tuné sur AMI réel.

Type d'entrée	# Micro	# Secteurs	Test LibriSpeeh				Test AMI			
			1-loc	2-loc	3-loc	1,2,3-loc	1-loc	2-loc	3-loc	1,2,3,4-loc
Original	4		7,21	14,82	19,76	16,30	25,89	41,70	54,68	45,25
Formation de voies	4	4	7,46	12,27	22,99	17,64	24,52	**39,61**	52,19	43,14
	8	4	**6,73**	**10,53**	**19,06**	**14,26**	**22,96**	40,01	**49,59**	**41,64**

démontré que les modèles entraînés sur des données issues de la formation de voies peuvent améliorer la performance de MC-ASR avec une réduction relative du WER allant jusqu'à 29%. Cela ouvre de nouvelles perspectives pour améliorer les caractéristiques d'entrée pour les enregistrements multicanaux.

8.5 Formation de voies et SA-ASR conjoints

Dans la Section 8.4, l'amélioration de MC-SA-ASR par rapport à SC-SA-ASR est limitée. Dans cette section, nous proposons de combiner SA-ASR avec une formation de voies dépendante des données. Cette méthode réduit le bruit et la réverbération et fusionne les canaux de mélange en un mélange rehaussé monocanal qui alimente ensuite SA-ASR.

8.5.1 Architecture du système

Nous proposons un système conjoint intégrant la formation de voies et SA-ASR pour la transcription de réunions multicanales à microphones distants. Comme illustré dans la Figure 8.7, le signal multicanal est d'abord traité par une formation de voies pour générer un signal monocanal rehaussé. Ce signal alimente ensuite SA-ASR pour obtenir les résultats de reconnaissance vocale et de locuteur. Nous comparons la performance de trois méthodes de formation de voies pour le fine-tuning de SA-ASR: *delay-and-sum* (DAS) fixe, *minimum variance distortionless response* (MVDR) hybride ou FaSNet neuronale. De plus, nous rétro-propageons le coût d'apprentissage de SA-ASR à FaSNet, afin d'ajuster la formation de voies en fonction des objectifs d'apprentissage de SA-ASR.

Formation de voies DAS

La formation de voies DAS (Johnson and Dudgeon, 1993) est une formation de voies fixe, qui dépend uniquement des délais entre les signaux des microphones et un microphone de référence. Elle a été utilisée comme prétraitement dans de nombreuses études d'ASR pour un seul locuteur, par exemple par Xiao et al. (2016). Elle consiste à calculer les délais

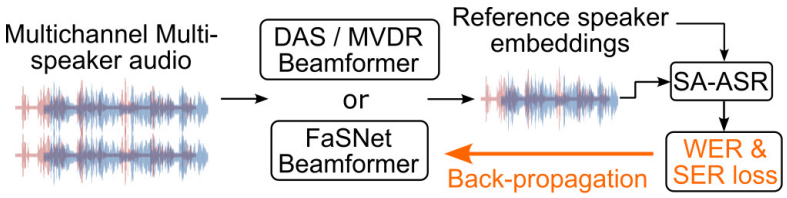


Figure 8.7: Système conjoint proposé de formation de voies et SA-ASR.

en utilisant un estimateur de différence de TDOA tel que GCC-PHAT (Knapp and Carter, 1976), à ajuster la phase des signaux des microphones en conséquence dans le domaine de la STFT, puis à les additionner.

Formation de voies MVDR

La formation de voies MVDR hybride (Lu et al., 2022; Kim et al., 2023) combine un réseau de neurones avec le filtrage MVDR classique. Le réseau de neurones est entraîné à estimer un masque temps-fréquence afin de rehausser les signaux souhaités et de supprimer les interférences. Ces masques sont ensuite utilisés pour calculer les filtres MVDR, qui minimisent la puissance de sortie tout en préservant les signaux cibles.

Formation de voies FaSNet-TAC

FaSNet (Luo et al., 2019) vise à estimer directement les filtres de formation de voies dans le domaine temporel. Il adopte un design en deux étapes : la première étape estime les filtres pour un microphone de référence, et la seconde étape estime les filtres pour les microphones restants en se basant sur les caractéristiques croisées entre la sortie pré-séparée et chaque microphone. Le paradigme de conception TAC (Luo et al., 2020a) aborde la permutation des canaux et est capable de gérer différents nombres de microphones.

Déréverbération avec WPE

La déréverberation multicanale par l'erreur de prédiction pondérée (WPE) (Nakatani et al., 2010) réduit la réverbération en modélisant et en soustrayant les composants réverbérants tardifs du signal audio observé. Cette technique optimise les coefficients de prédiction et les poids d'erreur pour améliorer la clarté et l'intelligibilité de la parole dans des environnements réverbérants.

SA-ASR

Un système SA-ASR de bout-en-bout basé sur Transformer (Kanda et al., 2021b) est utilisé pour l'ASR et la diarisation.

8.5.2 Protocole expérimental

Jeu de données

Nous appliquons une méthode d'alignement similaire à celle décrite dans la Section 8.3.1 pour pré-entraîner les formations de voies MVDR et FaSNet. La différence est que nous additionnons les extraits de microphones-casques alignés correspondants pour obtenir les mélanges rehaussés de référence. Pour obtenir de bonnes performances sur AMI réel, nous avons pré-entraîné le modèle SA-ASR sur le jeu de données LibriSpeech train-960, en utilisant la méthode décrite dans la Section 8.4.2 pour générer des données avec 2, 4 et 8 microphones contenant de 1 à 3 locuteurs. Le modèle SA-ASR est ensuite fine-tuné et évalué sur des données réelles d'AMI. Nous utilisons la méthode de segmentation de la Section 8.3.2 pour diviser les données en segments de 5 s.

Modèles et apprentissage

Nous utilisons l'implémentation de DAS provenant de l'outil SpeechBrain (Ravanelli et al., 2021). Pour le modèle MVDR, nous utilisons l'implémentation disponible dans TorchAudio (Yang et al., 2022; Lu et al., 2022). De plus, nous testons la performance obtenue avec WPE, implémentée par Drude et al. (2018). Dans les systèmes SA-ASR et MC-SA-ASR, l'encodeur ASR, le décodeur ASR et le décodeur de locuteur ont respectivement 12, 6 et 2 couches. Les filtres MVDR et FaSNet sont pré-entraînées sur AMI mélangé pendant 80 époques en utilisant l'optimiseur Adam avec un taux d'apprentissage de 10^{-3}. La configuration d'apprentissage pour SA-ASR et MC-SA-ASR sur AMI réel est identique à celle décrite dans la Section 8.3.2.

Métriques

Nous évaluons la performance de la formation de voies à l'aide du SI-SDR et de son amélioration (SI-SDRi) en dB sur le jeu de test AMI mélangé. La performance de SA-ASR est quantifiée par le WER et SER au niveau de la phrase (Kanda et al., 2020a) en % sur le jeu de test AMI réel.

8.5.3 Résultats

Fine-tuning de SA-ASR avec DAS par rapport à MVDR et FaSNet gelés

La Table 8.5 présente les résultats des modèles de base (SA-ASR et MC-SA-ASR) ainsi que de la combinaison de SA-ASR avec les trois méthodes de formation de voies, où les paramètres de MVDR et FaSNet sont figés lors du fine-tuning. Nous observons que le fine-tuning du modèle SA-ASR sur des données audio rehaussées par formation de voies

Table 8.5: SI-SDR (%), SI-SDRi (%), WER (%) et SER (%) pour les modèles fine-tunés et testés sur des données non traitées (SA-ASR et MC-SA-ASR) ou des données traitées par formation de voies (DAS-SA-ASR, MVDR-SA-ASR, FaSNet-SA-ASR). Par concision, nous notons SA-ASR, SI-SDR et SI-SDRi respectivement comme SA, SDR et SDRi.

Système	#Prm	#Cn	Test AMI mélangé								Test AMI réel							
			1-loc		2-loc		3-loc		1,2,3,4-loc		1-loc		2-loc		3-loc		1,2,3,4-loc	
			SDR	SDRi	SDR	SDRi	SDR	SDRi	SDR	SDRi	WER	SER	WER	SER	WER	SER	WER	SER
SA	69M	1	5,41	0	5,75	0	5,79	0	5,66	0	26,76	11,92	40,23	32,66	52,31	45,11	44,54	34,73
MC-SA	59M	2	5,41	0	5,75	0	5,79	0	5,66	0	26,41	11,73	40,79	32,64	52,59	43,82	44,99	34,43
+WPE (test)		2	5,72	0,31	5,93	0,18	5,92	0,13	5,87	0,21	26,43	12,13	40,80	32,54	52,32	44,14	44,72	34,65
DAS-SA	69M	2	5,62	0,21	5,42	-0,33	5,23	-0,56	5,39	-0,27	25,59	12,82	40,36	33,87	52,04	45,55	44,03	35,56
		4	5,66	0,25	5,47	-0,28	5,25	-0,54	5,35	-0,31	24,43	12,23	39,51	33,26	50,25	42,98	42,34	34,29
		8	5,66	0,25	5,48	-0,27	5,30	-0,49	5,38	-0,28	23,51	12,13	38,41	33,12	50,43	43,44	41,71	34,40
+WPE		2	6,08	0,67	5,70	-0,05	5,40	-0,39	5,66	0,00	24,65	11,59	38,64	32,14	50,29	43,87	42,39	34,05
		4	6,33	0,92	5,82	0,07	5,40	-0,39	5,67	0,01	23,50	11,26	37,22	32,00	49,34	42,34	**40,96**	**33,33**
		8	6,18	0,77	5,54	-0,21	5,04	-0,75	5,35	-0,31	23,49	11,43	37,89	32,67	50,12	43,59	41,37	**33,84**
MVDR-SA	74M	2	7,40	1,98	7,42	1,67	7,46	1,80	7,44	1,78	26,54	12,94	41,07	34,47	52,81	45,18	44,39	35,92
		4	7,94	2,52	7,98	2,23	7,99	2,19	7,99	2,33	27,15	12,63	41,35	34,92	52,72	44,94	44,76	35,78
		8	8,14	2,72	8,10	2,34	8,10	2,30	8,11	2,44	27,31	12,42	41,27	34,52	52,63	45,07	44,23	35,21
+WPE		2	7,75	2,34	7,48	1,73	7,48	1,69	7,55	1,89	26,40	12,78	41,09	33,66	52,00	44,89	44,29	35,54
		4	8,25	2,83	8,01	2,26	7,97	2,18	8,05	2,39	26,70	13,04	41,28	34,57	52,34	44,19	44,35	35,54
		8	8,40	2,99	8,09	2,34	8,03	2,24	8,14	2,48	26,35	12,93	41,03	33,72	52,12	44,26	44,12	35,37
FaSNet-SA	72M	2	10,21	4,79	9,85	4,09	9,56	3,76	**9,76**	**4,10**	26,86	11,33	40,91	35,67	52,57	47,12	44,57	36,24
		4	10,22	4,80	9,89	4,13	9,63	3,83	**9,82**	**4,15**	26,82	12,23	40,29	34,87	51,52	45,73	43,85	35,90
		8	10,41	4,99	10,01	4,25	9,72	3,92	**9,96**	**4,29**	26,53	10,73	39,93	34,78	51,70	45,28	44,11	35,51
+WPE		2	7,35	1,93	7,18	1,42	7,04	1,24	7,16	1,50	26,75	12,40	41,07	34,75	52,38	46,16	44,48	36,01
		4	6,48	1,07	6,26	0,51	6,13	0,34	6,24	0,58	25,86	12,63	39,27	33,15	51,78	44,16	43,29	34,97
		8	5,88	0,47	5,88	0,47	5,66	-0,13	5,85	0,19	26,16	10,78	39,35	34,89	51,22	45,25	43,39	35,41

améliore la performance d'ASR. En particulier, avec 8 canaux, l'utilisation de DAS entraîne une réduction relative de 6% du WER sans WPE (41,71%) et de 8% avec WPE (40,96%) par rapport à SA-ASR. Il est aussi intéressant de noter que, malgré la performance supérieure de FaSNet en termes de SI-SDRi pour le débruitage et la déréverbération, le modèle SA-ASR entraîné sur la parole traitée par FaSNet est moins performant que celui sur la parole traitée par DAS. Ceci est dû au fait que, bien que FaSNet réduise efficacement le bruit, il élimine également une partie du signal de parole. Au contraire, la formation de voies DAS peut préserver une portion significative de tous les signaux de parole tout en fournissant une certaine réduction du bruit, ce qui se traduit par de meilleurs résultats en reconnaissance de la parole et du locuteur.

Optimisation conjointe de FaSNet et SA-ASR

Étant donné que le fine-tuning de SA-ASR avec MVDR et FaSNet gelés n'améliore

pas significativement la performance de SA-ASR, nous avons effectué une optimisation conjointe de SA-ASR et de FaSNet. Les résultats de la Table 8.6 montrent qu'en l'absence de WPE, l'optimisation conjointe de FaSNet et de SA-ASR (40,52%) réduit le WER de 9% relatifs par rapport à FaSNet gelé (44,57%) et à SA-ASR (44,54%). Nous observons également une réduction de 7% du SER (33,68%) par rapport à FaSNet gelé (36,24%). Cependant, FaSNet fine-tuné présente un SI-SDRi plus bas que celui du pré-entraînement. Cela indique que l'apprentissage conjoint optimise FaSNet pour l'ASR plutôt que pour la réduction maximale du bruit et de la réverbération, au prix d'une plus grande distorsion de la parole. De plus, bien que le nombre d'époques d'apprentissage de FaSNet ait un impact significatif sur le SI-SDRi, il n'affecte pas de manière significative le résultat de l'optimisation conjointe.

Table 8.6: SI-SDR(%), SI-SDRi (%), WER (%) et SER (%) pour FaSNet à 2 canaux et SA-ASR appris conjointement.

Usage de WPE	**Test AMI mélangé**					**Test AMI réel**							
	Pré-entraîné			**Fine-tuné**		**1-loc**		**2-loc**		**3-loc**		**1,2,3,4-loc**	
	# Epo	**SDR**	**SDRi**	**SDR**	**SDRi**	**WER**	**SER**	**WER**	**SER**	**WER**	**SER**	**WER**	**SER**
Non	0	5,66	0	-16,21	-21,87	25,31	11,37	38,04	32,28	48,63	43,07	41,71	**33,68**
	5	9,27	3,61	5,13	-0,53	24,91	13,06	36,84	33,54	47,49	45,00	**40,60**	34,87
	50	9,69	4,02	7,05	1,39	24,54	13,51	36,93	33,36	47,71	43,87	**40,52**	34,43
Fine-tuning	0	5,87	0,21	-14,37	-20,03	25,51	11,75	38,58	32,03	50,23	42,53	43,41	**33,68**
	5	7,63	1,97	4,37	-1,29	25,04	12,59	37,46	31,16	48,05	41,59	41,00	**32,84**
	50	7,29	1,63	5,82	0,16	24,75	12,59	37,87	30,89	48,01	42,23	**40,92**	**33,14**
Test	50	7,29	1,63	7,29	1,63	25,11	11,90	38,03	31,73	48,61	42,52	41,46	33,33

8.5.4 Conclusion

Cette section a exploré l'intégration de la formation de voies avec SA-ASR pour la reconnaissance conjointe de la parole et du locuteur dans le contexte de réunions à distance. Nous avons évalué l'impact du fine-tuning de SA-ASR sur les sorties des formation de voies DAS, MVDR ou FaSNet, ainsi que le fine-tuning conjoint de SA-ASR avec FaSNet. Nous avons comparé ces approches avec la fusion de canaux basée sur MFCCA. Les expériences révèlent qu'au contraire des résultats publiés précédemment sur des données simulées, MFCCA est inefficace sur AMI réel. Cela souligne l'importance d'évaluer systématiquement SA-ASR sur des données de réunions réelles. L'utilisation de la formation de voies DAS et l'optimisation conjointe de SA-ASR avec FaSNet ont conduit à des réductions relatives du WER de 8% et 9%, respectivement. L'emploi de WPE dans le système DAS-SA-ASR peut entraîner une réduction relative de 3% à la fois pour le WER et le SER.

8.6 Pipeline de transcription de réunions en conditions réelles et son optimisation

Les sections précédentes ont introduit différentes architectures pour traiter des corpus de réunions synthétiques ou réelles déjà segmentées. Dans cette section, nous proposons un pipeline VAD-SD-SA-ASR pour la transcription de réunions réelles. Nous proposons d'affiner le modèle de SA-ASR sur les segments de sortie de la VAD au lieu de segments de taille fixe ou issus de la vérité terrain afin de mieux s'adapter aux conditions de test. Nous discutons également des stratégies pour améliorer l'identification des locuteurs dans SA-ASR en exploitant divers attributs liés à la VAD et la diarisation (SD), tels que le nombre et la longueur des segments utilisés pour obtenir les plongements des locuteurs (plongements de référence).

8.6.1 Méthodes proposées

Architecture du système

Notre système se compose des modules VAD, SD et SA-ASR. La Figure 8.8 illustre le pipeline global, qui commence par la VAD pour diviser l'ensemble de la réunion en segments de parole et de non-parole. Le système VAD prend en entrée le signal monocanal obtenu par la formation de voies DAS appliquée au signal à 8 canaux d'AMI. Nous utilisons les résultats de VAD pour effectuer la SD sur les segments de parole. Ensuite, sur la base des résultats de la SD, nous utilisons les segments de parole non-superposées de chaque locuteur pour calculer un plongement moyen pour ce locuteur. Enfin, les segments obtenus par la VAD et les plongements de locuteurs obtenus grâce à la SD servent d'entrée à SA-ASR, nous permettant de récupérer le contenu de la parole de tous les différents locuteurs dans un ordre premier entré, premier sorti. Le système de SA-ASR est monocanal et fonctionne sur le premier canal du signal d'AMI.

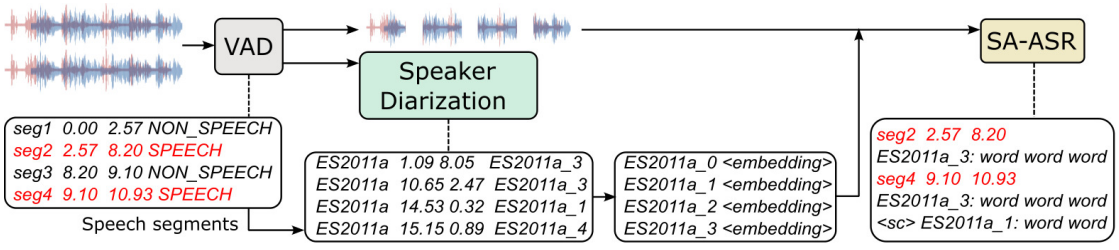


Figure 8.8: Pipeline proposé : VAD-SD-SA-ASR.

Pour la VAD, nous avons choisi d'utiliser l'architecture CRDNN (*Convolutional, Re-*

current, and Dense Neural Network) Sainath et al. (2015b); Xiang et al. (2021) qui combine des couches convolutives pour les variations de fréquence, des couches récurrentes pour la modélisation temporelle, et des couches denses pour la cartographie des caractéristiques, ce qui a réussi dans diverses tâches liées à la parole. Pour la SD et la SA-ASR, nous utilisons les modèles ECAPA-TDNN et SA-ASR employés dans la Section 8.4.

Préparation des données réelles

Dans les applications réelles, les positions de silence et les frontières de mots de la vérité terrain ne sont pas disponibles lors des tests. Il est donc nécessaire d'utiliser un système de VAD pour la segmentation. Afin d'adapter le modèle à la longueur des segments de VAD pendant les tests, nous utilisons également des segments de VAD pendant la phase d'apprentissage. Le processus d'obtention des segments de parole, des locuteurs et des étiquettes de texte est le même pour les données d'entraînement, de développement et de test. Comme le montre la Figure 8.9, toutes les données de réunion sont segmentées en segments de parole de taille variable à l'aide de VAD. Nous fusionnons les segments de parole adjacents séparés par un silence plus court qu'un seuil de durée donné, ce qui donne une longueur moyenne de segment dépendant du seuil spécifié. En utilisant les fichiers d'annotation d'AMI contenant des informations sur les segments de locuteurs et les frontières de mots, nous attribuons des séquences de texte et de locuteurs à chaque segment dans un ordre premier entré, premier sorti.

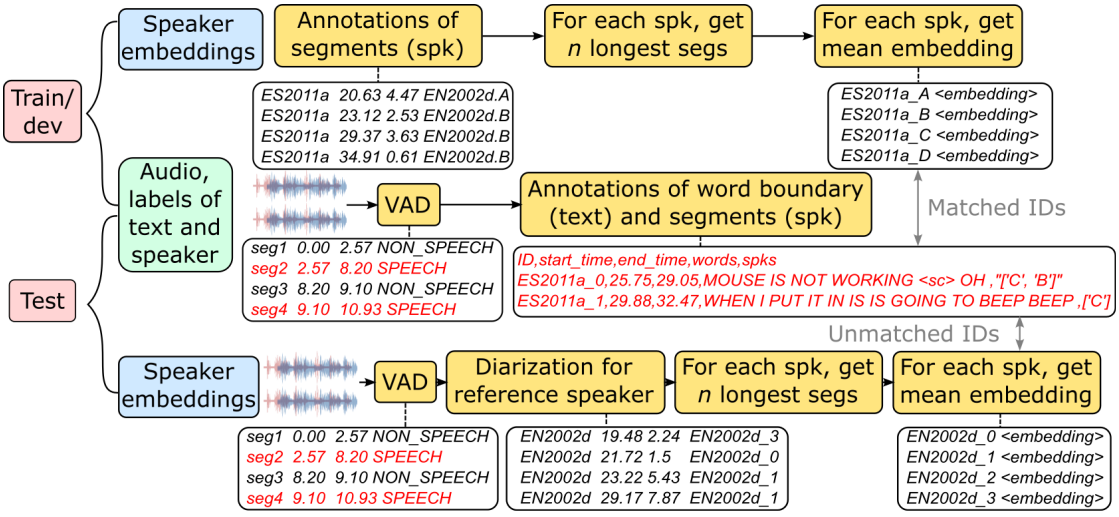


Figure 8.9: Préparation du corpus AMI pour les ensembles de données d'entraînement, de développement et de test.

Pour les jeux d'apprentissage et de développement, nous extrayons tous les segments

de parole pour chaque locuteur dans chaque réunion en nous basant sur les segments de locuteurs annotés. Nous sélectionnons les n segments les plus longs pour chaque locuteur. Nous utilisons un modèle de plongement de locuteur pré-entraîné pour encoder tous les n segments et calculons le plongement moyen comme référence pour le locuteur associé. Le processus pour l'ensemble de test est différent car, dans les scénarios de test réels, le nombre de locuteurs n'est pas connu à l'avance. Pour résoudre ce problème, nous appliquons le modèle de SD aux segments de VAD pour estimer le nombre de locuteurs et les segments de parole pour chaque locuteur. Les plongements de locuteurs sont ensuite dérivés de la même manière que pour les ensembles d'apprentissage et de développement.

8.6.2 Protocole expérimental

Jeu de données

Nous avons effectué le fine-tuning et les tests sur le corpus AMI en utilisant trois méthodes de segmentation. La première est la même que celle décrite dans la Section 8.3, et la deuxième est la méthode utilisant l'annotation de vérité terrain proposée par Kanda et al. (2021c) et Yang et al. (2023). La troisième méthode de segmentation utilise la VAD pour extraire les segments. Nous avons fusionné les segments adjacents en fonction des seuils de durée de silence de 0,1, 0,3, 0,5, 0,7 et 0,9 s, ce qui a donné des ensembles de données avec des nombres de segments et des longueurs moyennes de segments différents. Pour pré-entraîner SA-ASR, nous avons utilisé la méthode décrite dans la Section 8.4.2 pour générer des données multi-locuteurs simulées en utilisant le corpus LibriSpeech.

Modèles et apprentissage

Nos expériences ont été mises en œuvre en utilisant l'outil SpeechBrain. Pour la VAD, nous avons utilisé le modèle CRDNN[4] pré-entraîné sur Libriparty (Ravanelli, 2023). Le modèle SD est le même modèle ECAPA-TDNN pré-entraîné dans la Section 8.4. Le module ASR dans SA-ASR a été pré-entraîné sur des données simulées de LibriSpeech multi-locuteurs pendant 360 000 itérations. Les modules ASR et locuteur dans SA-ASR ont ensuite été pré-entraînés pendant 300 000 itérations supplémentaires. Enfin, nous avons fine-tuné le modèle sur AMI réel SDM pendant 20 000 itérations. Pendant les 10 000 premières itérations, nous avons uniquement ajusté le module ASR, tandis que les 15 000 itérations restantes ont été utilisées pour l'ajustement conjoint des modules ASR et locuteur.

[4]Disponible sur https://huggingface.co/speechbrain/vad-crdnn-libriparty

8.6.3 Résultats

Efficacité du fine-tuning sur les segments VAD

La Table 8.7 présente les résultats obtenus sur les segments de VAD avec un seuil de silence de 0,5 s. À titre de comparaison, nous rapportons également la performance de chaque modèle sur le jeu de test correspondant, segmenté de la même manière que le jeu d'apprentissage. Les modèles fine-tunés sur des segments de VAD présentent systématiquement une performance supérieure lorsqu'ils sont testés sur des segments de VAD comparés aux modèles fine-tunés sur des segments de taille fixe ou de vérité terrain. La réduction relative du WER peut atteindre jusqu'à 19% (de 55,91% à 45,09%), et la réduction relative du SER peut être aussi élevée que 28% (de 34,94% à 25,04%). Cependant, cette différence n'est pas aussi marquée lorsque les modèles sont testés sur leurs ensembles de test correspondants. Cela souligne l'importance de l'apprentissage sur des données qui sont étroitement alignées avec celle de test. Pour les applications en conditions réelles reposant sur des segments de VAD, il est conseillé de procéder à un fine-tuning sur les segments de VAD, même lorsque les annotations des segments de locuteur sont disponibles dans l'ensemble d'entraînement.

Table 8.7: WER et SER (%) sur le jeu de test AMI en utilisant différentes méthodes de segmentation pour le fine-tuning et le test de SA-ASR. Les plongements de locuteur sont calculés en moyennant les plongements obtenus à partir des 3 segments annotés les plus longs. Le nombre de segments et la durée moyenne des segments VAD 1,2,3,4-locuteurs sont respectivement 3 586 et 6,96 s. La colonne Dur indique la durée moyenne en secondes.

Méthode de segmentation	Données d'entraînement			Test (matchés)				Test (seg de VAD avec 0,5 s silence)							
				1,2,3,4-loc				1-loc		2-loc		3-loc		1-4-loc	
	Seuil	#Seg	Dur	#Seg	Dur	WER	SER	WER	SER	WER	SER	WER	SER	WER	SER
Fixée	-	90 426	6,59 s	10 234	6,63 s	44,54	28,54	49,41	28,92	50,91	29,78	57,48	36,29	55,91	34,94
Oracle	-	21 917	10,19 s	2 504	8,67 s	**43,42**	**21,81**	44,47	23,08	50,74	28,21	56,77	31,47	54,00	31,02
VAD	0,1 s	93 150	1,59 s	10 605	1,59 s	44,62	29,03	**30,53**	17,12	45,67	29,15	53,04	38,07	47,02	30,19
	0,3 s	57 589	3,40 s	6 095	3,71 s	44,50	28,85	31,95	16,17	43,54	28,99	52,38	35,69	46,33	28,99
	0,5 s	35 091	6,18 s	3 586	6,96 s	45,09	27,46	31,25	15,53	42,05	23,21	50,57	30,54	**45,09**	27,46
	0,7 s	22 027	10,33 s	2 265	10,24 s	45,80	24,82	32,13	**13,51**	**41,74**	**22,17**	**49,82**	**27,00**	46,13	27,02
	0,9 s	14 460	15,65 s	1 519	13,15 s	47,08	24,95	33,05	**13,35**	42,52	**22,31**	51,39	**27,30**	46,27	**25,04**

Amélioration de l'assignation des locuteurs par une meilleure création de plongements de locuteur

Nous avons examiné plus en détail comment différentes configurations, y compris la présence de segments chevauchés, le nombre de segments candidats et la longueur des segments candidats, influencent les plongements de locuteurs et le SER résultant. La Table 8.8

illustre l'influence de ces attributs en utilisant les segments de VAD avec un seuil de silence de 0,5 s comme entrées de la SD. L'utilisation de tous les segments candidats au lieu des 2 ou 10 plus longs conduit à une meilleure représentation des locuteurs, indépendamment de la longueur des segments, comme le montre une réduction du SER allant jusqu'à 15% relatifs (de 15,02% à 12,83%). De plus, des segments candidats plus courts montrent une plus grande sensibilité au nombre de segments. Cela peut être attribué au fait qu'un plus grand nombre de segments permet une représentation plus robuste et plus complète de chaque locuteur. La présence de segments de la parole superposée parmi les segments candidats ne semble pas avoir un impact significatif.

Table 8.8: SER (%) sur le jeu de test AMI avec un seuil de silence de VAD de 0,5 s en faisant varier le nombre et la longueur des segments candidats utilisés pour extraire les plongements des locuteurs. Le WER reste inchangé et équivalent à celui indiqué dans la Table 8.7.

Segments candidats			SER		
Taux superposition	**# Seg**	**Dur**	**1-loc**	**2-loc**	**3-loc**
non	2	2–5 s	15,02	23,72	29,75
non	10	2–5 s	14,44	21,92	27,82
non	tous	2–5 s	**12,83**	**20,93**	27,59
oui	tous	2–5 s	**12,80**	**20,81**	**27,28**
non	2	5–10 s	13,09	21,29	28,00
non	10	5–10 s	14,03	21,69	27,76
non	tous	5–10 s	13,88	21,82	27,97
oui	tous	5–10 s	13,93	21,72	27,92
non	2	6–50 s	13,04	21,42	27,53
non	10	6–50 s	13,33	21,08	**27,13**
non	tous	6–50 s	**12,97**	**20,76**	**27,43**
oui	tous	6–50 s	13,06	**20,81**	**27,41**

8.6.4 Conclusion

Dans cette section, nous nous sommes concentrés sur l'amélioration de l'identification des locuteurs dans le système SA-ASR pour les réunions en conditions réelles. Nous avons proposé un pipeline de VAD-SD-SA-ASR et de fine-tuning sur des segments de VAD plutôt que sur des segments de taille fixe. Nous avons montré que cela peut conduire à une réduction relative du SER allant jusqu'à 28%. Nous avons ensuite exploré des stratégies pour extraire les plongements de locuteurs en variant les configurations de VAD et de SD. Nos résultats révèlent que l'extraction des plongements à partir des sorties de SD plutôt que des segments de locuteurs annotés réduit le SER de jusqu'à 16% en relatif. De plus, bien que l'extraction des plongements de locuteurs à partir de segments courts ou longs puisse

donner un SER similaire, les segments plus longs permettent une meilleure représentation des locuteurs et une meilleure similarité entre les locuteurs à travers les segments, et sont moins sensibles au nombre de segments moyennés.

8.7 Perspectives

Intégration de la séparation de la parole et de MC-SA-ASR

Si l'on compare les résultats expérimentaux dans la Section 8.3 et dans la Section 8.4 pour le jeu de données de test à 2 locuteurs et 2 canaux sur le corpus AMI en utilisant des paramètres de fine-tuning comparables, la performance du modèle ASR avec séparation (41,67%) n'est pas meilleure que celle du modèle MC-SA-ASR (39,54%). Cependant, la méthode utilisant la séparation présente une meilleure performance en identification du locuteur, avec un SER de 12,67% comparé à 24,83%. Cela démontre que, d'une part, les erreurs causées par le modèle de séparation peuvent se propager à l'étape ASR et dégrader sa performance. Ainsi, l'utilisation d'une méthode SA-ASR de bout-en-bout peut offrir de meilleures performances pour la reconnaissance de la parole chevauchée. D'autre part, prédire l'identité du locuteur en utilisant les segments de parole séparés donne de meilleurs résultats que l'utilisation du décoder de locuteur, qui fournit une prédiction au niveau des tokens en utilisant l'architecture Transformer avec un nombre plus élevé de paramètres. Cela nous conduit à considérer comment intégrer ces deux méthodes pour améliorer la performance de reconnaissance.

Amélioration de la prédiction du token de changement de locuteur dans SA-ASR

Dans la Section 8.3, sur le jeu de données de test AMI, la précision du comptage des locuteurs pour 1, 2, 3 et 4 locuteurs est respectivement de 92,40%, 77,05%, 50,74% et 20,04%. Cette précision se dégrad rapidement à mesure que le nombre de locuteurs augmente, ce qui indique que, bien que la MC-SA-ASR soit plus performante que la SC-SA-ASR et la MC-ASR, elle présente encore des limitations dans la détermination précise du nombre de locuteurs dans des scénarios impliquant trois participants ou plus. Plusieurs perspectives sont possibles concernant cette proposition. Par exemple, l'assignation de l'étiquette de locuteur à un segment peut être basée sur les tokens de changement de locuteur dans la sortie de l'ASR. Nous pouvons considérer conjointement la sortie du bloc locuteur, ou bien de la SD, pour assister à la décision de changement de locuteur.

Architecture avancée de bout-en-bout avec une intégration plus poussée des informations d'ASR et de locuteur

Dans la Section 8.6, nous avons constaté que l'utilisation de meilleures représentations de locuteur n'améliore pas le WER. De plus, ne pas utiliser les informations de locuteur pour le décodage ASR ne conduit qu'à une augmentation relative du WER de 3%. Cela suggère que, dans l'architecture actuelle, le décodeur ASR est déjà optimisé pour utiliser efficacement les informations de locuteur pour le décodage du texte en fonction des probabilités de locuteur testées. La prochaine étape serait de concevoir un mécanisme de rétroaction efficace permettant au décodeur de locuteur d'assister le décodeur d'ASR. Cette perspective met en évidence la nécessité d'explorer de nouvelles stratégies pour intégrer plus efficacement les informations de locuteur afin d'améliorer la performance du système d'ASR.

Adaptation à d'autres langues

Dans cette thèse, le système de SA-ASR est appris sur des données en anglais. L'adaptation de ce système à une autre langue avec peu ou pas de données de réunions réelles transcrites serait particulièrement utile car toutes les langues ne disposent pas de grandes quantités de données. L'apprentissage par transfert pour l'adaptation d'un modèle d'une langue à une autre et l'apprentissage de modèles multilingues ont été largement étudiés pour l'ASR. Les recherches sur l'apprentissage par transfert pour les systèmes d'ASR multi-locuteur restent cependant limitées. Il s'agit donc d'une direction prometteuse pour des études futures, qui pourrait conduire à des améliorations significatives de l'applicabilité et la performance de ces systèmes dans différents contextes linguistiques.

Bibliography

Affes, S. and Grenier, Y. (1997). A signal subspace tracking algorithm for microphone array processing of speech. *IEEE Transactions on Speech and Audio Processing*, 5(5):425–437.

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

Araki, S., Fujimoto, M., Ishizuka, K., Sawada, H., and Makino, S. (2008). A DOA based speaker diarization system for real meetings. In *Joint Workshop on Hands-Free Speech Communication and Microphone Arrays*, pages 29–32.

Arora, A., Raj, D., Subramanian, A. S., Li, K., Ben-Yair, B., Maciejewski, M., Żelasko, P., Garcia, P., Watanabe, S., and Khudanpur, S. (2020). The JHU multi-microphone multi-speaker ASR system for the CHiME-6 challenge. In *6th International Workshop on Speech Processing in Everyday Environments (CHiME)*, pages 48–54.

Arthur, D. and Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *18th Annual ACM-SIAM Symposium on Discrete Algorithms*, volume 7, pages 1027–1035.

Asif, M., Vinodbhai, M. T., Mishra, S., Gupta, A., and Tiwary, U. S. (2022). Emotion recognition in VAD space during emotional events using CNN-GRU hybrid model on EEG signals. In *International Conference on Intelligent Human Computer Interaction*, pages 75–84.

Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., and Bengio, Y. (2016). End-to-end attention-based large vocabulary speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4945–4949.

Barker, J., Marxer, R., Vincent, E., and Watanabe, S. (2017). The third 'CHiME' speech separation and recognition challenge: Analysis and outcomes. *Computer Speech & Language*, 46:605–626.

Barker, J., Vincent, E., Ma, N., Christensen, H., and Green, P. (2013). The PASCAL CHiME speech separation and recognition challenge. *Computer Speech & Language*, 27(3):621–633.

Barker, J., Watanabe, S., Vincent, E., and Trmal, J. (2018). The fifth 'CHiME' speech separation and recognition challenge: Dataset, task and baselines. In *Interspeech*, pages 1561–1565.

Berouti, M., Schwartz, R., and Makhoul, J. (1979). Enhancement of speech corrupted by acoustic noise. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages 208–211.

Boeddeker, C., Heitkaemper, J., Schmalenstroeer, J., Drude, L., Heymann, J., and Haeb-Umbach, R. (2018). Front-end processing for the CHiME-5 dinner party scenario. In *5th International Workshop on Speech Processing in Everyday Environments (CHiME)*, volume 35–40.

Boeddeker, C., Subramanian, A. S., Wichern, G., Haeb-Umbach, R., and Le Roux, J. (2024). TS-SEP: Joint diarization and separation conditioned on estimated speaker embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:1185–1197.

Boll, S. (1979). Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(2):113–120.

Bonastre, J.-F., Delacourt, P., Fredouille, C., Merlin, T., and Wellekens, C. (2000). A speaker tracking system based on speaker turn detection for NIST evaluation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (ICASSP)*, volume 2, pages 1177–1180.

Bourlard, H. A. and Morgan, N. (1994). *Connectionist Speech Recognition: A Hybrid Approach*. Springer.

Bredin, H. and Laurent, A. (2021). End-to-end speaker segmentation for overlap-aware resegmentation. In *Interspeech*, pages 3111–3115.

Bredin, H., Yin, R., Coria, J. M., Gelly, G., Korshunov, P., Lavechin, M., Fustes, D., Titeux, H., Bouaziz, W., and Gill, M.-P. (2020). Pyannote. audio: neural building blocks for speaker diarization. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7124–7128.

Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kraaij, W., and Kronenthal, M. (2005). The AMI meeting corpus: A pre-announcement. In *International Workshop on Machine Learning for Multimodal Interaction*, pages 28–39.

Chan, W., Jaitly, N., Le, Q., and Vinyals, O. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4960–4964.

Chang, F.-J., Radfar, M., Mouchtaris, A., King, B., and Kunzmann, S. (2021). End-to-end multi-channel Transformer for speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5888.

Chang, X., Zhang, W., Qian, Y., Le Roux, J., and Watanabe, S. (2019). Mimo-speech: End-to-end multi-channel multi-speaker speech recognition. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 237–244.

Chang, X., Zhang, W., Qian, Y., Le Roux, J., and Watanabe, S. (2020). End-to-end multi-speaker speech recognition with Transformer. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6134–6138.

Chen, H., Zhang, P., and Yan, Y. (2019). Multi-talker MVDR beamforming based on extended complex Gaussian mixture model. *arXiv preprint arXiv:1910.07753*.

Chen, S. and Gopalakrishnan, P. (1998). Speaker, environment and channel change detection and clustering via the bayesian information criterion. In *DARPA Broadcast News Transcription and Understanding Workshop*, volume 8, pages 127–132.

Chen, S., Wu, Y., Chen, Z., Wu, J., Li, J., Yoshioka, T., Wang, C., Liu, S., and Zhou, M. (2021). Continuous speech separation with Conformer. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5749–5753.

Chen, Z., Luo, Y., and Mesgarani, N. (2017). Deep attractor network for single-microphone speaker separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 246–250.

Chen, Z., Yoshioka, T., Lu, L., Zhou, T., Meng, Z., Luo, Y., Wu, J., Xiao, X., and Li, J. (2020). Continuous speech separation: Dataset and analysis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7284–7288.

Chorowski, J., Bahdanau, D., Cho, K., and Bengio, Y. (2014). End-to-end continuous speech recognition using attention-based recurrent NN: First results. In *NIPS Workshop on Deep Learning*.

Chung, J. S., Nagrani, A., Coto, E., Xie, W., McLaren, M., Reynolds, D. A., and Zisserman, A. (2019). VoxSRC 2019: The first VoxCeleb speaker recognition challenge. *arXiv preprint arXiv:1912.02522*.

Cord-Landwehr, T., Boeddeker, C., Zorilă, C., Doddipatla, R., and Haeb-Umbach, R. (2023). Frame-wise and overlap-robust speaker embeddings for meeting diarization. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Cornell, S., Jung, J.-w., Watanabe, S., and Squartini, S. (2024). One model to rule them all? towards end-to-end joint speaker diarization and speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11856–11860.

Cornell, S., Wiesner, M., Watanabe, S., Raj, D., Chang, X., Garcia, P., Masuyama, Y., Wang, Z.-Q., Squartini, S., and Khudanpur, S. (2023). The CHiME-7 DASR challenge: Distant meeting transcription with multiple devices in diverse scenarios. *arXiv preprint arXiv:2306.13734*.

Cui, C., Sheikh, I., Sadeghi, M., and Vincent, E. (2023). End-to-end multichannel speaker-attributed ASR: Speaker guided decoder and input feature analysis. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8.

Cui, C., Sheikh, I. A., Sadeghi, M., and Vincent, E. (2024). Improving speaker assignment in speaker-attributed ASR for real meeting applications. In *The Speaker and Language Recognition Workshop (Odyssey)*, pages 99–106.

Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., and Ouellet, P. (2010). Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798.

Deng, L. and Li, X. (2013). Machine learning paradigms for speech recognition: An overview. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(5):1060–1089.

Desplanques, B., Thienpondt, J., and Demuynck, K. (2020). ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification. In *Interspeech*, pages 3830–3834.

Diaz-Guerra, D., Miguel, A., and Beltran, J. R. (2021). gpuRIR: A Python library for room impulse response simulation with GPU acceleration. *Multimedia Tools and Applications*, 80:5653–5671.

Doclo, S., Moonen, M., Van den Bogaert, T., and Wouters, J. (2009). Reduced-bandwidth and distributed MWF-based noise reduction algorithms for binaural hearing aids. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(1):38–51.

Dong, L., Xu, S., and Xu, B. (2018). Speech-Transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5888.

Dowerah, S., Serizel, R., Jouvet, D., Mohammadamini, M., and Matrouf, D. (2022). How to leverage DNN-based speech enhancement for multi-channel speaker verification? In *International Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI)*, page 168.

Drude, L., Hasenklever, D., and Haeb-Umbach, R. (2019). Unsupervised training of a deep clustering model for multichannel blind source separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 695–699.

Drude, L., Heymann, J., Boeddeker, C., and Haeb-Umbach, R. (2018). NARA-WPE: A python package for weighted prediction error dereverberation in Numpy and Tensorflow for online and offline processing. In *Speech Communication; 13th ITG-Symposium*, pages 1–5.

Duong, L. T., Nguyen, P. T., Di Sipio, C., and Di Ruscio, D. (2020). Automated fruit recognition using EfficientNet and MixNet. *Computers and Electronics in Agriculture*, 171:105326.

Fujita, Y., Kanda, N., Horiguchi, S., Nagamatsu, K., and Watanabe, S. (2019a). End-to-end neural speaker diarization with permutation-free objectives. In *Interspeech*, pages 4300–4304.

Fujita, Y., Kanda, N., Horiguchi, S., Xue, Y., Nagamatsu, K., and Watanabe, S. (2019b). End-to-end neural speaker diarization with self-attention. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 296–303.

Fujita, Y., Watanabe, S., Horiguchi, S., Xue, Y., Shi, J., and Nagamatsu, K. (2020). Neural speaker diarization with speaker-wise chain rule. *arXiv preprint arXiv:2006.01796*.

Gangadharaiah, R., Narayanaswamy, B., and Balakrishnan, N. (2004). A novel method for two-speaker segmentation. In *Interspeech*, pages 2337–2340.

Gannot, S., Burshtein, D., and Weinstein, E. (2001). Signal enhancement using beamforming and nonstationarity with applications to speech. *IEEE Transactions on Signal Processing*, 49(8):1614–1626.

Gao, Z., Zhang, S., McLoughlin, I., and Yan, Z. (2022). Paraformer: Fast and accurate parallel Transformer for non-autoregressive end-to-end speech recognition. In *Interspeech*, pages 2063–2067.

Gelly, G. and Gauvain, J.-L. (2017). Optimization of RNN-based speech activity detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(3):646–656.

Gish, H., Siu, M.-H., and Rohlicek, J. R. (1991). Segregation of speakers for speech recognition and speaker identification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 91, pages 873–876.

Graves, A. (2012). Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *23rd International Conference on Machine Learning*, pages 369–376.

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

Gu, R., Wu, J., Zhang, S.-X., Chen, L., Xu, Y., Yu, M., Su, D., Zou, Y., and Yu, D. (2019). End-to-end multi-channel speech separation. *arXiv preprint arXiv:1905.06286*.

Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., and Wu, Y. (2020). Conformer: Convolution-augmented Transformer for speech recognition. In *Interspeech*, pages 5036–5040.

Guo, P., Chang, X., Watanabe, S., and Xie, L. (2021). Multi-speaker ASR combining non-autoregressive Conformer CTC and conditional speaker chain. In *Interspeech*, pages 1401–1405.

Haeb-Umbach, R. (2023). Keynote talk at IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)): Multi-Talker Meeting Transcription. https://rc.signalprocessingsociety.org/workshops/asru/spsasru23vid0007.

Hampshire, J. and Waibel, A. (1989). A novel objective function for improved phoneme recognition using time delay neural networks. In *International Joint Conference on Neural Networks*, pages 235–241.

Han, E., Lee, C., and Stolcke, A. (2021). BW-EDA-EEND: Streaming end-to-end neural speaker diarization for a variable number of speakers. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7193–7197.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., and Coates, A. (2014). Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Härkönen, M., Broughton, S. J., and Samarakoon, L. (2024). EEND-M2F: Masked-attention mask Transformers for speaker diarization. *arXiv preprint arXiv:2401.12600*.

Harper, M. (2015). The automatic speech recogition in reverberant environments (ASpIRE) challenge. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 547–554.

Hayashi, T., Watanabe, S., Zhang, Y., Toda, T., Hori, T., Astudillo, R., and Takeda, K. (2018). Back-translation-style data augmentation for end-to-end ASR. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 426–433.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

He, M., Raj, D., Huang, Z., Du, J., Chen, Z., and Watanabe, S. (2021). Target-speaker voice activity detection with improved i-vector estimation for unknown number of speaker. In *Interspeech*, pages 3555–3559.

Hershey, J. R., Chen, Z., Le Roux, J., and Watanabe, S. (2016). Deep clustering: Discriminative embeddings for segmentation and separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 31–35.

Hori, T., Watanabe, S., Zhang, Y., and Chan, W. (2017). Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM. In *Interspeech*, pages 949–953.

Horiguchi, S., Fujita, Y., Watanabe, S., Xue, Y., and Nagamatsu, K. (2020). End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors. In *Interspeech*, pages 269–273.

Huang, Z., Watanabe, S., Fujita, Y., García, P., Shao, Y., Povey, D., and Khudanpur, S. (2020). Speaker diarization with region proposal network. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6514–6518.

Hummersone, C., Stokes, T., and Brookes, T. (2014). On the ideal ratio mask as the goal of computational auditory scene analysis. *Blind Source Separation: Advances in Theory, Algorithms and Applications*, pages 349–368.

Inaguma, H., Cho, J., Baskar, M. K., Kawahara, T., and Watanabe, S. (2019). Transfer learning of language-independent end-to-end ASR with language model fusion. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6096–6100.

Isik, Y., Le Roux, J., Chen, Z., Watanabe, S., and Hershey, J. R. (2016). Single-channel multi-speaker separation using deep clustering. In *Interspeech*, pages 545–549.

Ito, N., Araki, S., and Nakatani, T. (2016). Complex angular central Gaussian mixture model for directional statistics in mask-based microphone array signal processing. In *24th European Signal Processing Conference (EUSIPCO)*, pages 1153–1157.

Janin, A., Baron, D., Edwards, J., Ellis, D., Gelbart, D., Morgan, N., Peskin, B., Pfau, T., Shriberg, E., and Stolcke, A. (2003). The ICSI meeting corpus. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 364–367.

Jiang, D., He, Z., Lin, Y., Chen, Y., and Xu, L. (2021). An improved unsupervised single-channel speech separation algorithm for processing speech sensor signals. *Wireless Communications and Mobile Computing*, 2021(1):6655125.

Johnson, D. H. and Dudgeon, D. E. (1993). *Array Signal Processing: Concepts and Techniques*. Prentice Hall.

Kalinli, O., Seltzer, M. L., Droppo, J., and Acero, A. (2010). Noise adaptive training for robust automatic speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(8):1889–1901.

Kamo, N., Tawara, N., Matsuura, K., Ashihara, T., Moriya, T., Ogawa, A., Sato, H., Ochiai, T., Ando, A., and Ikeshita, R. (2023). NTT multi-speaker ASR system for the DASR task of CHiME-7 challenge. In *7th International Workshop on Speech Processing in Everyday Environments (CHiME)*, pages 45–50.

Kanda, N., Boeddeker, C., Heitkaemper, J., Fujita, Y., Horiguchi, S., Nagamatsu, K., and Haeb-Umbach, R. (2019a). Guided source separation meets a strong ASR backend: Hitachi/Paderborn university joint investigation for dinner party ASR. In *Interspeech*, pages 1248–1252.

Kanda, N., Chang, X., Gaur, Y., Wang, X., Meng, Z., Chen, Z., and Yoshioka, T. (2021a). Investigation of end-to-end speaker-attributed ASR for continuous multi-talker recordings. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 809–816.

Kanda, N., Gaur, Y., Wang, X., Meng, Z., Chen, Z., Zhou, T., and Yoshioka, T. (2020a). Joint speaker counting, speech recognition, and speaker identification for overlapped speech of any number of speakers. In *Interspeech*, pages 36–40.

Kanda, N., Gaur, Y., Wang, X., Meng, Z., and Yoshioka, T. (2020b). Serialized output training for end-to-end overlapped speech recognition. In *Interspeech*, pages 2797–2801.

Kanda, N., Horiguchi, S., Fujita, Y., Xue, Y., Nagamatsu, K., and Watanabe, S. (2019b). Simultaneous speech recognition and speaker diarization for monaural dialogue recordings with target-speaker acoustic models. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 31–38.

Kanda, N., Ikeshita, R., Horiguchi, S., Fujita, Y., Nagamatsu, K., Wang, X., Manohar, V., Soplin, N. E. Y., Maciejewski, M., and Chen, S.-J. (2018). The Hitachi/JHU CHiME-5 system: Advances in speech recognition for everyday home environments using multiple microphone arrays. In *5th International Workshop on Speech Processing in Everyday Environments (CHiME)*, pages 6–10.

Kanda, N., Wu, J., Wang, X., Chen, Z., Li, J., and Yoshioka, T. (2023). Vararray meets T-SOT: Advancing the state of the art of streaming distant conversational speech recog-

nition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Kanda, N., Wu, J., Wu, Y., Xiao, X., Meng, Z., Wang, X., Gaur, Y., Chen, Z., Li, J., and Yoshioka, T. (2022a). Streaming speaker-attributed ASR with token-level speaker embeddings. In *Interspeech*, pages 521–525.

Kanda, N., Xiao, X., Gaur, Y., Wang, X., Meng, Z., Chen, Z., and Yoshioka, T. (2022b). Transcribe-to-diarize: Neural speaker diarization for unlimited number of speakers using end-to-end speaker-attributed ASR. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8082–8086.

Kanda, N., Ye, G., Gaur, Y., Wang, X., Meng, Z., Chen, Z., and Yoshioka, T. (2021b). End-to-end speaker-attributed ASR with Transformer. In *Interspeech*, pages 4413–4417.

Kanda, N., Ye, G., Wu, Y., Gaur, Y., Wang, X., Meng, Z., Chen, Z., and Yoshioka, T. (2021c). Large-scale pre-training of end-to-end multi-talker ASR for meeting transcription with single distant microphone. In *Interspeech*, pages 3430–3434.

Karita, S., Chen, N., Hayashi, T., Hori, T., Inaguma, H., Jiang, Z., Someki, M., Soplin, N. E. Y., Yamamoto, R., and Wang, X. (2019). A comparative study on Transformer vs RNN in speech applications. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 449–456.

Kemp, T., Schmidt, M., Westphal, M., and Waibel, A. (2000). Strategies for automatic segmentation of audio data. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 3, pages 1423–1426.

Kim, J., Kim, J., Lee, S., Park, J., and Hahn, M. (2016). Vowel based voice activity detection with LSTM recurrent neural network. In *8th International Conference on Signal Processing Systems*, pages 134–137.

Kim, M., Cheong, S., and Shin, J. W. (2023). DNN-based parameter estimation for MVDR beamforming and post-filtering. In *Interspeech*, pages 3879–3883.

Kinoshita, K., Delcroix, M., Araki, S., and Nakatani, T. (2020). Tackling real noisy reverberant meetings with all-neural source separation, counting, and diarization system. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 381–385.

Knapp, C. and Carter, G. (1976). The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(4):320–327.

Ko, T., Peddinti, V., Povey, D., Seltzer, M. L., and Khudanpur, S. (2017). A study on data augmentation of reverberant speech for robust speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5220–5224.

Kolbæk, M., Yu, D., Tan, Z.-H., and Jensen, J. (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.

Kudo, T. and Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97.

Kunze, J., Kirsch, L., Kurenkov, I., Krug, A., Johannsmeier, J., and Stober, S. (2017). Transfer learning for speech recognition on a budget. In Blunsom, P., Bordes, A., Cho, K., Cohen, S., Dyer, C., Grefenstette, E., Hermann, K. M., Rimell, L., Weston, J., and Yih, S., editors, *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 168–177, Vancouver, Canada. Association for Computational Linguistics.

Lang, K. J., Waibel, A. H., and Hinton, G. E. (1990). A time-delay neural network architecture for isolated word recognition. *Neural Networks*, 3(1):23–43.

Le Roux, J., Wisdom, S., Erdogan, H., and Hershey, J. R. (2019). SDR–half-baked or well done? In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 626–630.

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

Lee, H.-S., Peng, Y.-H., Huang, P.-T., Tseng, Y.-C., Wu, C.-H., Tsao, Y., and Wang, H.-M. (2020). The Academia Sinica systems of speech recognition and speaker diarization for the CHiME-6 Challenge. In *6th International Workshop on Speech Processing in Everyday Environments (CHiME)*, pages 29–32.

Li, B., Gulati, A., Yu, J., Sainath, T. N., Chiu, C.-C., Narayanan, A., Chang, S.-Y., Pang, R., He, Y., and Qin, J. (2021a). A better and faster end-to-end model for streaming ASR. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5634–5638.

Li, C., Shi, J., Zhang, W., Subramanian, A. S., Chang, X., Kamo, N., Hira, M., Hayashi, T., Boeddeker, C., and Chen, Z. (2021b). ESPnet-SE: End-to-end speech enhancement and separation toolkit designed for ASR integration. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 785–792.

Li, J., Zhao, R., Hu, H., and Gong, Y. (2019). Improving RNN transducer modeling for end-to-end speech recognition. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 114–121.

Li, Y., Yu, F., Liang, Y., Guo, P., Shi, M., Du, Z., Zhang, S., and Xie, L. (2023). Sa-Paraformer: Non-autoregressive end-to-end speaker-attributed ASR. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–7.

Lin, J., Cai, X., Dinkel, H., Chen, J., Yan, Z., Wang, Y., Zhang, J., Wu, Z., Wang, Y., and Meng, H. (2023a). Av-Sepformer: Cross-attention sepformer for audio-visual target speaker extraction. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Lin, J., Un, K.-F., Yu, W.-H., Martins, R. P., and Mak, P.-I. (2023b). A 47-nW voice activity detector (VAD) featuring a short-time CNN feature extractor and an RNN-Based classifier with a non-volatile CAP-ROM. *IEEE Journal of Solid-State Circuits*, 58:3020–3029.

Liu, K., Du, Z., Wan, X., and Zhou, H. (2023). X-Sepformer: End-to-end speaker extraction network with explicit optimization on speaker confusion. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Liu, Y. C., Han, E., Lee, C., and Stolcke, A. (2021). End-to-end neural diarization: From Transformer to Conformer. In *Interspeech*, pages 3081–3085.

Love, B. C. (2002). Comparing supervised and unsupervised category learning. *Psychonomic Bulletin & Review*, 9(4):829–835.

Lu, L., Kanda, N., Li, J., and Gong, Y. (2021). Streaming multi-talker speech recognition with joint speaker identification. In *Interspeech*, pages 1782–1782.

Lu, Y.-J., Cornell, S., Chang, X., Zhang, W., Li, C., Ni, Z., Wang, Z.-Q., and Watanabe, S. (2022). Towards low-distortion multi-channel speech enhancement: The ESPNet-SE submission to the L3DAS22 challenge. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 9201–9205.

Luo, J., Wang, J., Cheng, N., Xiao, E., Zhang, X., and Xiao, J. (2022). Tiny-Sepformer: A tiny time-domain Transformer network for speech separation. In *Interspeech*, pages 5313–5317.

Luo, Y., Chen, Z., Mesgarani, N., and Yoshioka, T. (2020a). End-to-end microphone permutation and number invariant multi-channel speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6394–6398.

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

Luo, Y., Han, C., Mesgarani, N., Ceolini, E., and Liu, S.-C. (2019). FaSNet: Low-latency adaptive beamforming for multi-microphone audio processing. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 260–267.

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

Maas, A. L., Le, Q. V., O'Neil, T. M., Vinyals, O., Nguyen, P., and Ng, A. Y. (2012). Recurrent neural networks for noise reduction in robust ASR. In *Interspeech*, pages 22–25.

Maiti, S., Ueda, Y., Watanabe, S., Zhang, C., Yu, M., Zhang, S.-X., and Xu, Y. (2023). EEND-SS: Joint end-to-end neural speaker diarization and speech separation for flexible number of speakers. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 480–487.

Manohar, V., Chen, S.-J., Wang, Z., Fujita, Y., Watanabe, S., and Khudanpur, S. (2019). Acoustic modeling for overlapping speech recognition: JHU CHiME-5 challenge system. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6665–6669.

Mary, N. J. M. S., Umesh, S., and Katta, S. V. (2021). S-vectors and TESA: Speaker embeddings and a speaker authenticator based on Transformer encoder. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:404–413.

Medennikov, I., Korenevsky, M., Prisyach, T., Khokhlov, Y., Korenevskaya, M., Sorokin, I., Timofeeva, T., Mitrofanov, A., Andrusenko, A., Podluzhny, I., Laptev, A., and Romanenko, A. (2020). Target-speaker voice activity detection: A novel approach for multi-speaker diarization in a dinner party scenario. In *Interspeech*, pages 274–278.

Menne, T., Schlüter, R., and Ney, H. (2019). Investigation into joint optimization of single channel speech enhancement and acoustic modeling for robust ASR. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6660–6664.

Mun, S. H., Han, M. H., Moon, C., and Kim, N. S. (2023). EEND-DEMUX: End-to-end neural speaker diarization via demultiplexed speaker embeddings. *arXiv preprint arXiv:2312.06065*.

Nagrani, A., Son Chung, J., Huh, J., Brown, A., Coto, E., Xie, W., McLaren, M., Reynolds, D. A., and Zisserman, A. (2020). VoxSRC 2020: The second VoxCeleb speaker recognition challenge. *arXiv preprint arXiv:2012.06867*.

Nakatani, T., Yoshioka, T., Kinoshita, K., Miyoshi, M., and Juang, B.-H. (2010). Speech dereverberation based on variance-normalized delayed linear prediction. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(7):1717–1731.

Ning, H., Liu, M., Tang, H., and Huang, T. S. (2006). A spectral clustering approach to speaker diarization. In *Interspeech*, pages 2178–2181.

NIST (2024). SCTK. https://github.com/usnistgov/SCTK.git.

Nugraha, A. A., Liutkus, A., and Vincent, E. (2016). Multichannel audio source separation with deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(9):1652–1664.

Ochiai, T., Watanabe, S., Hori, T., Hershey, J. R., and Xiao, X. (2017). Unified architecture for multichannel end-to-end speech recognition with neural beamforming. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1274–1288.

O'Malley, T., Narayanan, A., Wang, Q., Park, A., Walker, J., and Howard, N. (2021). A Conformer-based ASR frontend for joint acoustic echo cancellation, speech enhancement and speech separation. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 304–311.

Ozerov, A. and Févotte, C. (2009). Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):550–563.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). Librispeech: An ASR corpus based on public domain audio books. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

Pariente, M., Cornell, S., Cosentino, J., Sivasankaran, S., Tzinis, E., Heitkaemper, J., Olvera, M., Stöter, F.-R., Hu, M., Martín-Doñas, J. M., Ditter, D., Frank, A., Deleforge, A., and Vincent, E. (2020). Asteroid: the PyTorch-based audio source separation toolkit for researchers. In *Interspeech*, pages 2637–2641.

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. (2019a). SpecAugment: A simple data augmentation method for automatic speech recognition. In *Interspeech*, page 2613.

Park, T. J., Han, K. J., Kumar, M., and Narayanan, S. (2019b). Auto-tuning spectral clustering for speaker diarization using normalized maximum eigengap. *IEEE Signal Processing Letters*, 27:381–385.

Park, T. J., Kanda, N., Dimitriadis, D., Han, K. J., Watanabe, S., and Narayanan, S. (2022). A review of speaker diarization: Recent advances with deep learning. *Computer Speech & Language*, 72:101317.

Peddinti, V., Chen, G., Manohar, V., Ko, T., Povey, D., and Khudanpur, S. (2015). JHU aspire system: Robust LVCSR with TDNNs, ivector adaptation and RNN-LMs. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 539–546.

Povey, D., Cheng, G., Wang, Y., Li, K., Xu, H., Yarmohammadi, M., and Khudanpur, S. (2018). Semi-orthogonal low-rank matrix factorization for deep neural networks. In *Interspeech*, pages 3743–3747.

Pujol, P., Pol, S., Nadeu, C., Hagen, A., and Bourlard, H. (2004). Comparison and combination of features in a hybrid HMM/MLP and a HMM/GMM speech recognition system. *IEEE Transactions on Speech and Audio processing*, 13(1):14–22.

Quan, C. and Li, X. (2022). Multichannel speech separation with narrow-band Conformer. In *Interspeech*, pages 5378–5382.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. (2023). Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518.

Raj, D., Denisov, P., Chen, Z., Erdogan, H., Huang, Z., He, M., Watanabe, S., Du, J., Yoshioka, T., and Luo, Y. (2021). Integration of speech separation, diarization, and recognition for multi-speaker meetings: System description, comparison, and analysis. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 897–904.

Ravanelli, M. (2023). Libriparty. https://github.com/speechbrain/speechbrain/tree/develop/recipes/LibriParty/generate_dataset.

Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., Subakan, C., Dawalatabad, N., Heba, A., Zhong, J., Chou, J.-C., Yeh, S.-L., Fu, S.-W., Liao, C.-F., Rastorgueva, E., Grondin, F., Aris, W., Na, H., Gao, Y., Mori, R. D., and Bengio, Y. (2021). SpeechBrain: A general-purpose speech toolkit. *arXiv preprint arXiv:2106.04624*.

Saijo, K. and Scheibler, R. (2022). Spatial loss for unsupervised multi-channel source separation. In *Interspeech*, pages 241–245.

Sainath, T. N., Kingsbury, B., Saon, G., Soltau, H., Mohamed, A.-r., Dahl, G., and Ramabhadran, B. (2015a). Deep convolutional neural networks for large-scale speech tasks. *Neural Networks*, 64:39–48.

Sainath, T. N., Vinyals, O., Senior, A., and Sak, H. (2015b). Convolutional, long short-term memory, fully connected deep neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4580–4584.

Sang, M., Zhao, Y., Liu, G., Hansen, J. H., and Wu, J. (2023). Improving Transformer-based networks with locality for automatic speaker verification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Saon, G. and Chien, J.-T. (2012). Large-vocabulary continuous speech recognition systems: A look at some recent advances. *IEEE Signal Processing Magazine*, 29(6):18–33.

Scheibler, R., Zhang, W., Chang, X., Watanabe, S., and Qian, Y. (2023). End-to-end multi-speaker ASR with independent vector analysis. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 496–501.

Seki, H., Hori, T., Watanabe, S., Le Roux, J., and Hershey, J. R. (2018). A purely end-to-end system for multi-speaker speech recognition. In *56th Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 2620–2630.

Sell, G., Snyder, D., McCree, A., Garcia-Romero, D., Villalba, J., Maciejewski, M., Manohar, V., Dehak, N., Povey, D., and Watanabe, S. (2018). Diarization is hard: Some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge. In *Interspeech*, pages 2808–2812.

Shao, Y., Zhang, S.-X., and Yu, D. (2022). Multi-channel multi-speaker ASR using 3D spatial feature. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6067–6071.

Shi, J., Chang, X., Watanabe, S., and Xu, B. (2022). Train from scratch: Single-stage joint training of speech separation and recognition. *Computer Speech & Language*, 76:101387.

Shi, M., Du, Z., Chen, Q., Yu, F., Li, Y., Zhang, S., Zhang, J., and Dai, L.-R. (2023a). CASA-ASR: Context-aware speaker-attributed ASR. In *Interspeech*, pages 411–415.

Shi, M., Zhang, J., Du, Z., Yu, F., Chen, Q., Zhang, S., and Dai, L.-R. (2023b). A comparative study on multichannel speaker-attributed automatic speech recognition in multi-party meetings. In *APSIPA Annual Summit and Conference*, pages 1943–1948.

Siegler, M. A., Jain, U., Raj, B., and Stern, R. M. (1997). Automatic segmentation, classification and clustering of broadcast news audio. In *DARPA Speech Recognition Workshop*, pages 97–99.

Sivaraman, A., Wisdom, S., Erdogan, H., and Hershey, J. R. (2022). Adapting speech separation to real-world meetings using mixture invariant training. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 686–690.

Sklyar, I., Piunova, A., and Liu, Y. (2021). Streaming multi-speaker ASR with RNN-T. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6903–6907.

Snyder, D., Chen, G., and Povey, D. (2015). MUSAN: A music, speech, and noise corpus. *arXiv preprint arXiv:1510.08484*.

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., and Khudanpur, S. (2018). X-vectors: Robust DNN embeddings for speaker recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333.

Srinivasu, P. N., SivaSai, J. G., Ijaz, M. F., Bhoi, A. K., Kim, W., and Kang, J. J. (2021). Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM. *Sensors*, 21(8):2852.

Subakan, C., Ravanelli, M., Cornell, S., Bronzi, M., and Zhong, J. (2021). Attention is all you need in speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 21–25.

Taherian, H. and Wang, D. (2021). Time-domain loss modulation based on overlap ratio for monaural conversational speaker separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5744–5748.

Takahashi, N., Parthasaarathy, S., Goswami, N., and Mitsufuji, Y. (2019). Recursive speech separation for unknown number of speakers. In *Interspeech*, pages 1348–1352.

Thomas, S., Ganapathy, S., Saon, G., and Soltau, H. (2014). Analyzing convolutional neural networks for speech activity detection in mismatched acoustic conditions. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2519–2523.

Trentin, E. and Gori, M. (2001). A survey of hybrid ANN/HMM models for automatic speech recognition. *Neurocomputing*, 37(1-4):91–126.

Tritschler, A. and Gopinath, R. A. (1999). Improved speaker segmentation and segments clustering using the Bayesian information criterion. In *6th European Conference on Speech Communication and Technology*, pages 679–682.

Variani, E., Lei, X., McDermott, E., Moreno, I. L., and Gonzalez-Dominguez, J. (2014). Deep neural networks for small footprint text-dependent speaker verification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4052–4056.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Vincent, E., Araki, S., Theis, F., Nolte, G., Bofill, P., Sawada, H., Ozerov, A., Gowreesunker, V., Lutter, D., and Duong, N. Q. (2012). The signal separation evaluation campaign (2007–2010): Achievements and remaining challenges. *Signal Processing*, 92(8):1928–1936.

Vincent, E., Barker, J., Watanabe, S., Le Roux, J., Nesta, F., and Matassoni, M. (2013). The second 'CHiME' speech separation and recognition challenge: Datasets, tasks and baselines. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 126–130.

Vincent, E., Gribonval, R., and Févotte, C. (2006). Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1462–1469.

Vincent, E., Virtanen, T., and Gannot, S. (2018). *Audio Source Separation and Speech Enhancement*. John Wiley & Sons.

Vincent, E., Watanabe, S., Barker, J., and Marxer, R. (2016). The 4th CHiME speech separation and recognition challenge. https://www.chimechallenge.org/challenges/chime4/.

von Neumann, T., Boeddeker, C., Cord-Landwehr, T., Delcroix, M., and Haeb-Umbach, R. (2023). Meeting recognition with continuous speech separation and transcription-supported diarization. *arXiv preprint arXiv:2309.16482*.

von Neumann, T., Boeddeker, C., Drude, L., Kinoshita, K., Delcroix, M., Nakatani, T., and Haeb-Umbach, R. (2020a). Multi-talker ASR for an unknown number of sources: Joint training of source counting, separation and asr. In *Interspeech*, pages 3097–3101.

Von Neumann, T., Kinoshita, K., Delcroix, M., Araki, S., Nakatani, T., and Haeb-Umbach, R. (2019). All-neural online source separation, counting, and diarization for meeting analysis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 91–95.

von Neumann, T., Kinoshita, K., Drude, L., Boeddeker, C., Delcroix, M., Nakatani, T., and Haeb-Umbach, R. (2020b). End-to-end training of time domain audio separation and recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7004–7008.

Waibel, A. (1989). Modular construction of time-delay neural networks for speech recognition. *Neural Computation*, 1(1):39–46.

Waibel, A., Sawai, H., and Shikano, K. (1989). Modularity and scaling in large phonemic neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(12):1888–1898.

Wang, D. (2005). On ideal binary mask as the computational goal of auditory scene analysis. *Speech Separation by Humans and Machines*, pages 181–197.

Wang, D. and Chen, J. (2018). Supervised speech separation based on deep learning: An overview. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(10):1702–1726.

Wang, D., Wang, X., and Lv, S. (2019). An overview of end-to-end automatic speech recognition. *Symmetry*, 11(8):1018.

Wang, Q., Downey, C., Wan, L., Mansfield, P. A., and Moreno, I. L. (2018). Speaker diarization with LSTM. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5239–5243.

Wang, R., He, M., Du, J., Zhou, H., Niu, S., Chen, H., Yue, Y., Yang, G., Wu, S., and Sun, L. (2023a). The USTC-NERCSLIP systems for the CHiME-7 DASR challenge. In *7th International Workshop on Speech Processing in Everyday Environments (CHiME)*.

Wang, S., Kong, X., Peng, X., Movassagh, H., Prakash, V., and Lu, Y. (2023b). Dasformer: Deep alternating spectrogram Transformer for multi/single-channel speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Wang, Z., Zhou, Y., Gan, L., Chen, R., Tang, X., and Liu, H. (2022). DE-DPCTnet: Deep encoder dual-path convolutional Transformer network for multi-channel speech separation. In *IEEE Workshop on Signal Processing Systems (SiPS)*, pages 1–5.

Wang, Z.-Q., Cornell, S., Choi, S., Lee, Y., Kim, B.-Y., and Watanabe, S. (2023c). TF-GridNet: Integrating full-and sub-band modeling for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:3221–3236.

Wang, Z.-Q., Cornell, S., Choi, S., Lee, Y., Kim, B.-Y., and Watanabe, S. (2023d). TF-GridNet: Making time-frequency domain models great again for monaural speaker separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Wang, Z.-Q., Erdogan, H., Wisdom, S., Wilson, K., Raj, D., Watanabe, S., Chen, Z., and Hershey, J. R. (2021a). Sequential multi-frame neural beamforming for speech separation and enhancement. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 905–911.

Wang, Z.-Q., Wang, P., and Wang, D. (2021b). Multi-microphone complex spectral mapping for utterance-wise and continuous speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:2001–2014.

Watanabe, S., Mandel, M., Barker, J., Vincent, E., Arora, A., Chang, X., Khudanpur, S., Manohar, V., Povey, D., and Raj, D. (2020). CHiME-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings. In *6th International Workshop on Speech Processing in Everyday Environments (CHiME)*, pages 1–7.

Wisdom, S., Tzinis, E., Erdogan, H., Weiss, R., Wilson, K., and Hershey, J. (2020). Unsupervised sound separation using mixture invariant training. *Advances in Neural Information Processing Systems*, 33:3846–3857.

Wu, J., Chen, Z., Chen, S., Wu, Y., Yoshioka, T., Kanda, N., Liu, S., and Li, J. (2021). Investigation of practical aspects of single channel speech separation for ASR. In *Interspeech*, pages 3066–3070.

Xiang, Y., Tang, T., Su, T., Brach, C., Liu, L., Mao, S. S., and Geimer, M. (2021). Fast CRDNN: Towards on site training of mobile construction machines. *IEEE Access*, 9:124253–124267.

Xiao, X., Kanda, N., Chen, Z., Zhou, T., Yoshioka, T., Chen, S., Zhao, Y., Liu, G., Wu, Y., and Wu, J. (2021). Microsoft speaker diarization system for the VoxCeleb speaker recognition challenge 2020. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5824–5828.

Xiao, X., Xu, C., Zhang, Z., Zhao, S., Sun, S., Watanabe, S., Wang, L., Xie, L., and Jones, D. L. (2016). A study of learning based beamforming methods for speech recognition. In *International Workshop on Speech Processing in Everyday Environments (CHiME)*, pages 26–31.

Yang, J.-Y. and Chang, J.-H. (2020). Virtual acoustic channel expansion based on neural networks for weighted prediction error-based speech dereverberation. In *Interspeech*, pages 3930–3934.

Yang, M., Kanda, N., Wang, X., Wu, J., Sivasankaran, S., Chen, Z., Li, J., and Yoshioka, T. (2023). Simulating realistic speech overlaps improves multi-talker ASR. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Yang, X., Tan, B., Ding, J., Zhang, J., and Gong, J. (2010). Comparative study on voice activity detection algorithm. In *International Conference on Electrical and Control Engineering*, pages 599–602.

Yang, Y.-Y., Hira, M., Ni, Z., Astafurov, A., Chen, C., Puhrsch, C., Pollack, D., Genzel, D., Greenberg, D., Yang, E. Z., et al. (2022). Torchaudio: Building blocks for audio and speech processing. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6982–6986.

Ye, L., Lu, H., Cheng, G., Chen, Y., Shang, Z., and Li, X. (2023). The IACAS-Thinkit system for CHiME-7 challenge. In *7th International Workshop on Speech Processing in Everyday Environments (CHiME)*, pages 23–26.

Yoshioka, T., Abramovski, I., Aksoylar, C., Chen, Z., David, M., Dimitriadis, D., Gong, Y., Gurvich, I., Huang, X., and Huang, Y. (2019). Advances in online audio-visual meeting transcription. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 276–283.

Yoshioka, T., Erdogan, H., Chen, Z., and Alleva, F. (2018a). Multi-microphone neural speech separation for far-field multi-talker speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5739–5743.

Yoshioka, T., Erdogan, H., Chen, Z., Xiao, X., and Alleva, F. (2018b). Recognizing overlapped speech in meetings: A multichannel separation approach using neural networks. In *Interspeech*, pages 3038–3042.

Yoshioka, T. and Nakatani, T. (2012). Generalization of multi-channel linear prediction methods for blind MIMO impulse response shortening. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(10):2707–2720.

Yoshioka, T., Nakatani, T., Miyoshi, M., and Okuno, H. G. (2010). Blind separation and dereverberation of speech mixtures by joint optimization. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(1):69–84.

Yoshioka, T., Wang, X., Wang, D., Tang, M., Zhu, Z., Chen, Z., and Kanda, N. (2022). VarArray: Array-geometry-agnostic continuous speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6027–6031.

Yu, D., Chang, X., and Qian, Y. (2017a). Recognizing multi-talker speech with permutation invariant training. In *Interspeech*, pages 2456–2460.

Yu, D., Kolbæk, M., Tan, Z.-H., and Jensen, J. (2017b). Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 241–245.

Yu, F., Du, Z., Zhang, S., Lin, Y., and Xie, L. (2022a). A comparative study on speaker-attributed automatic speech recognition in multi-party meetings. In *Interspeech*, pages 560–564.

Yu, F., Zhang, S., Fu, Y., Xie, L., Zheng, S., Du, Z., Huang, W., Guo, P., Yan, Z., and Ma, B. (2022b). M2MeT: The ICASSP 2022 multi-channel multi-party meeting transcription challenge. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6167–6171.

Yu, F., Zhang, S., Guo, P., Liang, Y., Du, Z., Lin, Y., and Xie, L. (2023). MFCCA: Multi-frame cross-channel attention for multi-speaker ASR in multi-party meeting scenario. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 144–151.

Zeghidour, N. and Grangier, D. (2021). Wavesplit: End-to-end speech separation by speaker clustering. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:2840–2849.

Zeyer, A., Bahar, P., Irie, K., Schlüter, R., and Ney, H. (2019). A comparison of Transformer and LSTM encoder decoder models for ASR. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 8–15.

Zhang, W., Boeddeker, C., Watanabe, S., Nakatani, T., Delcroix, M., Kinoshita, K., Ochiai, T., Kamo, N., Haeb-Umbach, R., and Qian, Y. (2021). End-to-end dereverberation, beamforming, and speech recognition with improved numerical stability and advanced frontend. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6898–6902.

Zhao, T., Zhao, Y., Wang, S., and Han, M. (2021). Unet++-based multi-channel speech dereverberation and distant speech recognition. In *12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 1–5.

Zheng, S., Huang, W., Wang, X., Suo, H., Feng, J., and Yan, Z. (2021). A real-time speaker diarization system based on spatial spectrum. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7208–7212.

Zhou, T., Zhao, Y., Li, J., Gong, Y., and Wu, J. (2019). CNN with phonetic attention for text-independent speaker verification. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 718–725.