Localization guided speech separation

Sunit Sivasankaran

UNIVERSITÉ DE LORRAINE

Thèse de Doctorat

Sunit SIVASANKARAN

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

École doctorale : IAEM

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

Soutenue le 4 Septembre 2020
Thèse N°:

Séparation de la parole guidée par la localisation

JURY

Rapporteur :	**Nobutaka ONO**, Professeur, Tokyo Metropolitan University, Japon
Rapporteur :	**Sylvain MARCHAND**, Professeur, Université de La Rochelle, France
Examinateur :	**François PORTET**, Maître de Conférences, Université Grenoble-Alpes, France
Examinatrice :	**Marie-Odile BERGER**, Directrice de Recherche, Inria Nancy - Grand Est, France
Directeur de thèse :	**Emmanuel VINCENT**, Directeur de Recherche, Inria Nancy - Grand Est, France
Co-directeur de thèse :	**Dominique FOHR**, Chargé de Recherche, CNRS, France

Résumé

Les assistants vocaux font partie de notre vie quotidienne. Leurs performances sont mises à l'épreuve en présence de distorsions du signal, telles que le bruit, la réverbération et les locuteurs simultanés. Cette thèse aborde le problème de l'extraction du signal d'intérêt dans de telles conditions acoustiques difficiles en localisant d'abord le locuteur cible puis en utilisant la position spatiale pour extraire le signal de parole correspondant.
Dans un premier temps, nous considérons la situation courante où le locuteur cible prononce un mot ou une phrase connue, comme le mot de réveil d'un système de commande vocale mains-libres. Nous proposons une méthode afin d'exploiter cette information textuelle pour améliorer la localisation du locuteur en présence de locuteurs simultanés. La solution proposée utilise un système de reconnaissance vocale pour aligner le mot de réveil au signal vocal corrompu. Un spectre de référence représentant les phones alignés est utilisé pour calculer un identifiant qui est ensuite utilisé par un réseau de neurones profond pour localiser le locuteur cible. Les résultats sur des données simulées montrent que la méthode proposée réduit le taux d'erreur de localisation par rapport à la méthode classique GCC-PHAT. Des améliorations similaires sont constatées sur des données réelles.
Étant donnée la position spatiale estimée du locuteur cible, la séparation de la parole est effectuée en trois étapes. Dans la première étape, une simple formation de voie *delay-and-sum* (DS) est utilisée pour rehausser le signal provenant de cette direction, qui est utilisé dans la deuxième étape par un réseau de neurones pour estimer un masque temps-fréquence. Ce masque est utilisé pour calculer les statistiques du second ordre et pour effectuer une formation de voie adaptative dans la troisième étape. Un ensemble de données réverbéré, bruité avec plusieurs canaux et plusieurs locuteurs — inspiré du célèbre corpus WSJ0-2mix — a été généré et la performance de la méthode proposée a été étudiée en terme du taux d'erreur sur les mots (WER). Pour rendre le système plus robuste aux erreurs de localisation, une approche par déflation guidée par la localisation (SLOGD) qui estime les sources de manière itérative est proposée. À chaque itération, la position spatiale d'un locuteur est estimée puis utilisée pour estimer un masque correspondant à ce même locuteur. La source estimée est retirée du mélange avant d'estimer la position et le masque de la source suivante. La méthode proposée surpasse Conv-TasNet.
Enfin, le problème d'expliquer la robustesse des réseaux de neurones utilisés pour calculer les masques temps-fréquence à des conditions de bruit différentes. Nous utilisons la méthode dite SHAP pour quantifier la contribution de chaque point temps-fréquence du signal d'entrée au masque temps-fréquence estimé. Nous définissons une métrique qui résume les valeurs SHAP et montrons qu'elle est corrélée au WER obtenu sur la parole séparée. À notre connaissance, il s'agit de la première étude sur l'explicabilité des réseaux de neurones dans le contexte de la séparation de la parole.

Abstract

Voice based personal assistants are part of our daily lives. Their performance suffers in the presence of signal distortions, such as noise, reverberation, and competing speakers. This thesis addresses the problem of extracting the signal of interest in such challenging conditions by first localizing the target speaker and using the location to extract the target speech.

In a first stage, a common situation is considered when the target speaker utters a known word or sentence such as the wake-up word of a distant-microphone voice command system. A method that exploits this text information in order to improve the speaker localization performance in the presence of competing speakers is proposed. The proposed solution uses a speech recognition system to align the wake-up word to the corrupted speech signal. A model spectrum representing the aligned phones is used to compute an identifier which is then used by a deep neural network to localize the target speaker. Results on simulated data show that the proposed method reduces the localization error rate compared to the classical GCC-PHAT method. Similar improvements are observed on real data.

Given the estimated location of the target speaker, speech separation is performed in three stages. In the first stage, a simple delay-and-sum (DS) beamformer is used to enhance the signal impinging from that location which is then used in the second stage to estimate a time-frequency mask corresponding to the localized speaker using a neural network. This mask is used to compute the second-order statistics and to derive an adaptive beamformer in the third stage. A multichannel, multispeaker, reverberated, noisy dataset — inspired from the famous WSJ0-2mix dataset — was generated and the performance of the proposed pipeline was investigated in terms of the word error rate (WER). To make the system robust to localization errors, a Speaker LOcalization Guided Deflation (SLOGD) based approach which estimates the sources iteratively is proposed. At each iteration the location of one speaker is estimated and used to estimate a mask corresponding to that speaker. The estimated source is removed from the mixture before estimating the location and mask of the next source. The proposed method is shown to outperform Conv-TasNet.

Finally, we consider the problem of explaining the robustness of neural networks used to compute time-frequency masks to mismatched noise conditions. We employ the so-called SHAP method to quantify the contribution of every time-frequency bin in the input signal to the estimated time-frequency mask. We define a metric that summarizes the SHAP values and show that it correlates with the WER achieved on separated speech. To the best of our knowledge, this is the first known study on neural network explainability in the context of speech separation.

Acknowledgements

I would like to thank all the people who contributed in their own particular way to the achievement of this doctoral thesis. First and foremost, I thank my doctoral supervisors Emmanuel Vincent and Dominique Fohr. I greatly appreciate the freedom they gave me to do my research, and the encouragement and support that they always offered me. Their insights, advice and discussion have all been important for the materialisation of this thesis. I hope to emulate their integrity, work ethic and passion in my own research career. I want to thank my thesis committee members, Nobutaka Ono and Sylvain Marchand, for investing their time and providing valuable feedback. It was really great to have them in my thesis committee. Special thanks to Marie-Odile Berger and François Portet for agreeing to be part of my jury.

It was a great experience to be part of the Multispeech group at LORIA-Inria. Being part of this lab for close to 5 years, I have seen it grow leaps and bounds. I would like to thank all the members of this group - including but not limited to - Denis Jouvet, Antoine Deleforge, Romain Serizel, Irina Illina, Théo Biasutto-Lervat, Ashwin D'sa, Tulika Bose, Raphaël Duroselle, Michel Olvera, Mathieu Hu, Md Sahidullah, Elodie Gauthier, Mathieu Fontaine, Jen-Yu Liu - for the welcoming and fruitful environment they created. Special thanks to my officemates — the C145 gang — Manuel Pariente, Nicolas Furnon, Nicolas Turpault, and Ken Déguernel. Thanks for all the discussions and creating a lively environment which kept me going even when the chips were down. I couldn't have asked for a better set of people to share office space with. My heartfelt thanks to Imran Sheikh, Aditya Arie Nugraha, Aghilas Sini, Dung Tran and Guillaume Carajabal for all those white board discussions. I learnt a lot from them.
Thanks to Hélène Cavallini, Sabrina Ferry-Tritz and Souad Boutaguermouchet for their administrative assistance and making research the only thing I needed to spend my mental energy on, during my stay at Nancy.

Coming to Nancy 5 years ago, with no knowledge of the French language and little exposure to the culture, I found myself in a sea of unfamiliarity. Krishnan, Aghilas Sini, Ameur Douib, Imran Sheikh, C145 gang and Antoine Luitkus - thank you very much for all your support. I couldn't have had an enjoyable stay at Nancy if not for all your help.

I would also like to thank my teachers from Indian Institute of Technology-Madras and Rashtreeya Vidyalaya College of Engineering, Bangalore. These are the places where I picked up initial interests in my field and I am grateful for all their support.

My family played an important role and supported me all through out. Special thanks to my parents Thevarupakil Sivasankaran and Leelavathi Melepat Sivasankaran for their sacrifices and for believing in me. I couldn't have done this without them. Finally, I thank my wife Anupama Chingacham for being part of this journey. Her unconditional love and support made things lot easier.

This thesis was funded by the French national research agency, in the framework of the project VOCADOM "Robust voice command adapted to the user and to the context of assisted living" (ANR-16-CE33-0006). Experiments presented in this thesis were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr). High performance computing resources were partially provided by the EXPLOR centre hosted by the University de Lorraine. Part of the research was conducted at the 2019 Frederick Jelinek memorial summer workshop on speech and language technologies, hosted at L'École de Technologie Supérieure (Montreal, Canada) and sponsored by Johns Hopkins University with unrestricted gifts from Amazon, Facebook, Google, and Microsoft.

Contents

List of figures

List of tables

List of acronyms

ASR automatic speech recognition
ATF acoustic transfer function
BAN blind analytic normalization
BCE binary cross entropy
Bi-LSTM bidirectional long short term memory
CGMM complex Gaussian mixture model
CNN convolutional neural network
CRNN convolutional and recurrent neural network
CSIPD cosine-sine interchannel phase difference
DNN deep neural network
DOA direction-of-arrival
DRR direct-to-reverberant ratio
DS delay-and-sum
ELR early-to-late ratio
EM expectation-maximization
FIR finite impulse response
GAN generative adversarial network
GCC-PHAT generalized cross-correlation with phase transform
GEV generalized eigenvalue
GEVD generalized eigenvalue decomposition
GMM Gaussian mixture model
HMM hidden Markov model
IBM ideal binary mask
ICA independent component analysis
ILD interchannel level difference
IRM ideal ratio mask
ITD interchannel time difference
LRP layerwise relevance propagation
LSTM long short-term memory
MVDR minimum variance distortionless response
MWF multichannel Wiener filter
MFCC Mel frequency cepstral coefficient
MSE mean squared error
NMF nonnegative matrix factorization
PIT permutation invariant training
ReLU rectified linear unit
RIR room impulse response

RTF relative transfer function
RNN recurrent neural network
SAD speaker activity detection
SDW-MWF speech distortion weighted multichannel Wiener filter
SIR signal-to-interference ratio
SNR signal-to-noise ratio
SRP steered response power
STFT short-time Fourier transform
TDOA time difference of arrival
WER word error rate
WFST weighted finite state transducer
WPE weighted prediction error
WSJ Wall Street Journal

1 Introduction

1.1 Motivation

Humans are social beings. We interact with each other using a set of verbal and visual cues. Over generations, we have developed sophisticated sounds which when strung together have a meaning. We use these sounds to communicate with each other and pass on knowledge across generations. These sounds, which are referred to as speech, have been central to our evolution and are an essential part of our daily life.
With the advent of industrialization, we have developed machines which have now become ubiquitous. It is only natural to want to interact with these machines using speech. The quest for such interactions has led to what is popularly known as voice assistants and, with the improvements in their quality and robustness, these assistants have made inroads in our daily life. Typical examples of such commercially available assistants are Google Assistant and Amazon Alexa. They are found across many devices such as mobile phones, smart speakers, refrigerators, televisions, remote controls and robots, enabling easier interaction. The demand for such voice-enabled devices has grown exponentially over the years and its numbers are expected to triple from existing 2.5 billion to 8 billion devices by the end of 2023 (Perez, 2019).
The speech signal captured by these devices gets distorted due to reverberation, interfering speech and noise (Loizou, 2013; Virtanen et al., 2012; Wölfel and McDonough, 2009). These distortions reduce the intelligibility of speech and impact the performance of voice assistants.

1.2 Adversaries of speech

Adversaries of speech are ubiquitous and comes in many forms. In a typical home environment as shown in Fig. 1.1, the sounds emitted by television and radio sets, the sound resulting from cooking activities in the kitchen, the sound of water flowing through a tap and the sink, and communication between other members of the family can be considered as unwanted signals for the voice assistant. The device captures all these signals and their reflections on surfaces along with the target signal[1]. A brief overview of these distortions is provided below.

[1] We refer to the signal of interest as the target signal/speech in the rest of the thesis.

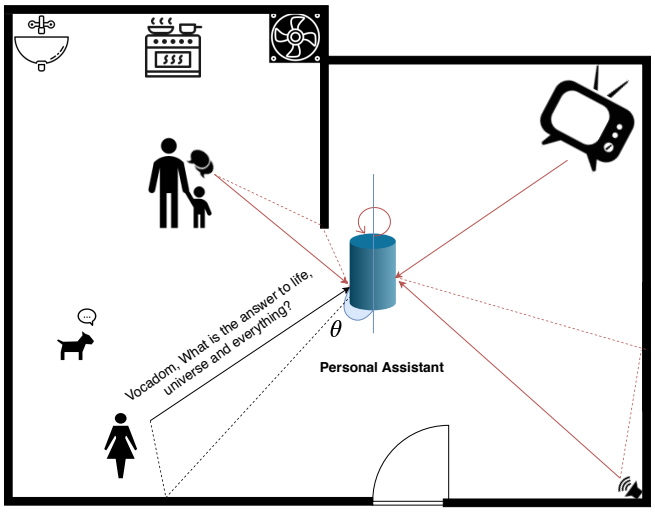


Figure 1.1: A typical home environment. A small subset of noises in the environment are marked in red and echoes in dotted lines.

1.2.1 Reverberation

Speech is a wave and propagates by oscillating the particles in a medium such as air. These waves get reflected, refracted and absorbed by surfaces such as walls, mirrors and tables. The direct and (attenuated) reflected waves impinge on a microphone at different time intervals. This phenomenon can be approximated using a finite impulse response (FIR) filter containing a very large number of taps (in the order of thousands), which is referred to as the room impulse response (RIR). The RIR depends on the room characteristics and the position of the speaker and the microphone. Reverberation was shown to have a larger impact on speech intelligibility than other distortions for people with cochlear implants (Hazrati and Loizou, 2012).
In this context, a room is characterized by its reverberation time (T_{60}) which is defined as the time taken for the reverberant tail to decay by 60 decibels (dB). T_{60} is a function of the room dimension and the sound absorption properties of the materials inside the room. It can be approximated using Eyring's Formula (Eyring, 1930) as

$$T_{60} = \frac{0.163V}{-S \log\left(1 - \frac{\sum_i S_i \alpha_i}{S}\right)}. \tag{1.1}$$

Here, V is the volume of the room, S_i and α_i are the area and the absorption coefficient of the i-th surface and $S = \sum_i S_i$. This expression can further be approximated for small absorption coefficients as

$$T_{60} = \frac{0.163V}{\sum_i S_i \alpha_i}. \tag{1.2}$$

A longer T_{60} has a detrimental effect on the ASR performance (Yoshioka et al., 2012).
A typical RIR consists of three parts as shown in Fig. 1.2:

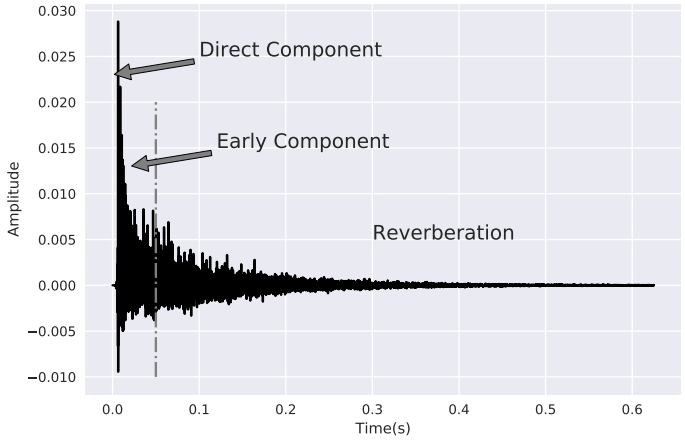


Figure 1.2: Simulated room impulse response of a "shoe-box" room with a dimension of $[7.2 \times 6.8 \times 3]$ m and $T_{60} = 0.67$ s. The source is placed at a distance of 2.25 m from the microphone. The simulation is done using the image-source method (Allen and Berkley, 1979), using RIR-generator (Habets, 2018).

1. The direct component, that is the direct path between the source and the microphone.
2. The early component, which includes the reflections corresponding to the first 50 ms after the arrival of the direct component. These are also referred to as early echoes in the literature (Vincent et al., 2018).
3. Reverberation, that is composed of the late reflections and decays exponentially in time.

The direct-to-reverberant ratio (DRR), which is defined as the ratio of the energy of the direct component to the reverberated component, quantifies the amount of reverberation present in a recorded speech signal. The early-to-late ratio (ELR), is defined similarly as the ratio of the energy of the sum of early and direct components to the reverberated component. Signals with lower DRR and ELR have low speech intelligibility and are a concern for voice assistants (Wölfel and McDonough, 2009). The process of dereverberating a signal (Habets and Naylor, 2018) involves the removal of reverberation while retaining only the direct path and the early component. Indeed, there is evidence that early echoes aid speech recognition performance (Brutti and Matassoni, 2016).

1.2.2 Interfering speech

Interfering speech is a highly non-stationary source of speech distortion. The problem of separating target and interfering speech in a noisy mixture is often referred to as the cocktail party problem (McDermott, 2009; Bregman, 1994). The signal-to-interference ratio (SIR), defined as the ratio of the target speech energy to the interfering speech en-

ergy, quantifies the influence of interfering speech in the mixture. Even though humans have innate abilities to direct attention to a target speaker while suppressing interfering speech, speech intelligibility is known to degrade in scenarios with low SIR values (Bronkhorst, 2000).
Separating speech means figuring out which bits of speech belong to which speaker. The human auditory system uses multiple clues such as the intensity, pitch and onset and offset times to cluster these bits together. This is hard for machines and complexity increases multi-fold in the presence of reverberation and other noises.

1.2.3 Noise

Noises can broadly be classified as stationary and non-stationary depending on the change in their signal statistics over time. The mean and variance of the sound emitted from a kitchen ventilator or a car engine do not change drastically over time and are therefore termed as stationary. On the other hand these statistics change over time for signals such as music which are therefore referred to as non-stationary. It is harder to remove non-stationary noises compared to stationary noises. Another categorization of noise is based on its spatial characteristics. Noises which have the same energy in all incoming directions (at the point of capture by the microphone) are referred to as diffuse noise whereas noises having higher energy in one particular direction are referred to as localized or directional noise. The signal-to-noise ratio (SNR), that is the ratio between the energy of speech and noise in the mixture, influences the intelligibility of the target speech. To make any sense of the target speech, the device will have to first remove or reduce the interfering noise, thereby increasing the SNR value.
Another type of distortion which is of interest is the acoustic echo (Hänsler and Schmidt, 2005). This term refers to signals for which a reference can be obtained. For example, voice assistants are often equipped with loudspeakers in order to "speak" back to the user. The delayed version of the signal generated by the assistant, and its reverberation are also captured by the assistant, resulting in an acoustic echo. The assistant has access to the reference signal, which can be used to minimize the impact of acoustic echo.

1.3 Hands-free voice assistants

Voice assistants can be broadly classified into two categories based on the impact of noise, reverberation and interfering speech on the system performance, namely:

1. *Handheld assistants*, which expect users to be in close proximity to the device. In such situations, even though reverberation, noise and interfering speech may exist, these distortions have much lower energy than the direct component of the target speech signal. This results in better quality of the captured signal and by extension, a richer user experience. In such scenarios, speech recognition performance has been reported to be on par with human performance (Xiong et al., 2017). This is typically the case of voice-based applications in handheld devices such as mobile phones.

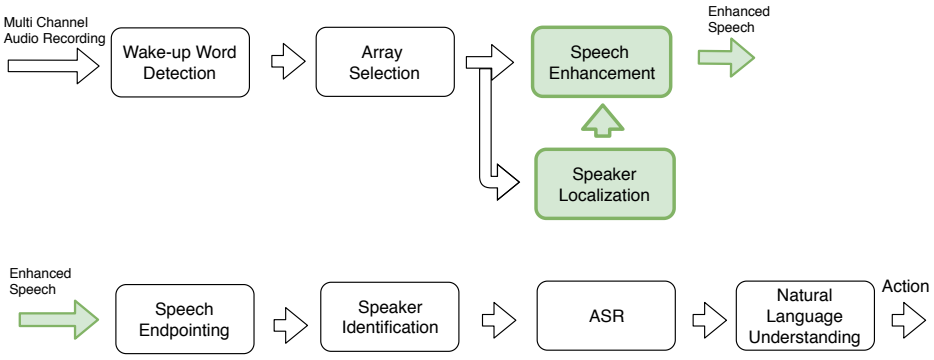


Figure 1.3: Personal assistant pipeline considered as part of the ANR Vocadom project.

2. *Hands-free assistants*, where the user is at a significant physical distance from the device, often in the order of $0.5 - 5$ m. In such scenarios, reverberation, noise and possibly interfering speech have higher energy compared to the direct component thereby reducing the quality of speech captured by the device (Wölfel and McDonough, 2009). This is typically the case in smart speakers such as Google Home and Amazon Alexa.

In the rest of the thesis, we focus on the latter category of voice assistants. Let us consider, for example, the situation when a person asks a hands-free voice assistant: "Vocadom, what is the answer to life, universe and everything?". Before it can come with the answer of 42 (Adams, 1980), the device must execute a series of steps.

The hands-free voice assistant pipeline proposed as part of the ANR Vocadom project (Vacher et al., 2018)[2] is shown in Fig. 1.3. The project envisions placing a 4-microphone array in each room of a home. The processing modules include:

1. *Wake-up word detector:* The first module facing the user is a wake-up word detector. This module is always active and triggers when a particular wake-up word, the word "Vocadom" in this example, is uttered by the speaker.
2. *Array selection:* After the wake-up word has been detected, the array selection module decides which microphone array is most suited to process the captured signal.
3. *Speech localization:* Once the array has been selected, its 4 microphones are used to find the position of the speaker in the room.
4. *Speech enhancement:* The localization information is used to enhance speech coming from that speaker.
5. *Speech endpointing:* The enhanced signal is used to decide the endpoint, i.e., the time instant when the user has completed his interaction with the device.
6. *Speaker identification:* The captured signal after endpointing is used to estimate the identity of the speaker (Hansen and Hasan, 2015; Greenberg et al., 2020) which is

[2]This thesis work is carried out under the Vocadom project supported by the French National Research Agency (ANR) under the contract ANR-16-CE33-0006. The project was designed for homes which requires assisted living. Details available at http://vocadom.imag.fr/.

used to improve the ASR performance. The identity can also be used to determine whether the speaker is allowed to interact with the device.

7. *Automatic speech recognition (ASR):* The enhanced speech along with the speaker identity is used to convert the spoken message to text (Li et al., 2015).
8. *Natural language understanding (NLU):* An NLU module (Tur, 2011) is used to understand the transcribed text and act upon it.
9. *Text-to-speech (TTS):* The answer is communicated back to the user using speech synthesized from text (Taylor, 2009).

1.4 Tools used to address the problem

The focus of this thesis is on the speech separation and speaker localization modules as highlighted in green in Fig. 1.3. For both of these problems we propose new solutions at the interface of deep learning and multichannel signal processing, a.k.a. beamforming.

Deep learning: Deep neural networks (DNN) (Deng and Yu, 2014; Goodfellow et al., 2016) are machine learning algorithms which have achieved state-of-the art performance in many machine learning tasks such as ASR (Watanabe et al., 2017), object recognition and machine translation (Koehn, 2020). Though neural networks have existed for a long time (Rosenblatt, 1958), they have recently come to the fore owing to the extensive availability of data in certain domains and advances in computing technologies.
DNNs consist of a set of linear transforms (parameterized by a set of weights) and non-linear functions connected in a particular order. The order of the connections determines the network architecture. They are often stacked in layers, with the objective that each layer learns a higher-level representation based on the lower-level representation learned by the preceding layer (Goodfellow et al., 2016). Multilayer networks provide better representations at the cost of an increase in the number of model parameters.
Convolutional neural networks (CNN) and recurrent neural networks (RNN) are some of the popular choices of neural network architectures (LeCun et al., 2015). A CNN involves a set of filters trained to extract patterns from the data. Each layer contains multiple such filters. Eeach convolutional layer involves the convolution — or correlation to be precise — of the input signal or the output of the previous layer with each of the filters, called a feature map. RNNs are designed for time-series data which are common in domains such as speech and finance. The general idea is to make the network outputs depend on both the current data and data from the previous time instants. Long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and gated recurrent unit (GRU) (Cho et al., 2014b) are popular choices of units to design a recurrent layer.
DNNs are most often trained in a supervised fashion, where the model is given input data along with the expected outputs. The model then learns to estimate the desired output given the input features. DNNs can also be trained in an unsupervised fashion, where the model is given only the input data and learns to unearth the hidden structure in that data. In either of these training strategies, an error function is constructed and the weights are updated using optimization algorithms such as stochastic gradient descent

(SGD) and Adam (Kingma and Ba, 2015). In this thesis, all our networks are trained in a supervised fashion.
The focus of this thesis is not to propose new DNN architectures, but rather to leverage standard architectures in a new way to address signal processing problems.

Beamforming: Beamformers (or spatial filters) (Van Trees, 2002) are linear filters which process multichannel speech input captured from multiple microphones and produce an enhanced output signal. They can exploit both the spectral and spatial characteristics of the speech and noise signals. Beamformers can be broadly classified as data-independent (or fixed) beamformers and data-dependent (or adaptive) beamformers. Fixed beamformers such as the delay-and-sum beamformer (Anguera et al., 2007) are time-invariant and depend only on the spatial position of the source of interest. In contrast, adaptive beamformers such as the multichannel Wiener filter (Souden et al., 2013) are computed using the statistics of the input signal and they may vary over time. Adaptive beamformers generally outperform fixed beamformers at the cost of substantially higher computational complexity (Gannot et al., 2017). Some of these beamformers are discussed in detail in Section 2.3.3.

1.5 Objectives and contributions

Multichannel speech enhancement methods for mixtures containing a single speaker and noise with low reverberation have already reached very good performance as shown in multiple works such as in CHiME-3 evaluation challenge (Barker et al., 2017) and by Xiong et al. (2017). Therefore, the focus of this thesis is on multichannel speech enhancement for hands-free voice assistants. Specifically we are interested in recovering the target speech in the presence of an interfering speaker in a reverberant, noisy environment. Since the interfering speaker can also be considered as a source, this problem is generally referred to as speech separation in the community and can be considered as a special case of audio source separation.
As shown by Xiao et al. (2016) multichannel speech enhancement methods based on end-to-end deep learning tend to perform poorly compared to signal processing based enhancement methods when facing such difficult mixtures. Our goal is to demonstrate the usefulness of localization information for multichannel speech separation in the presence of noise and reverberation. Parallel work done by Chen et al. (2018a) further validates the importance of location information for speech separation.

1.5.1 Contributions of the thesis

The contributions of the thesis are as follows.

1. It is well known that the performance of speaker localization algorithms deteriorates in the presence of noise and reverberation (Evers et al., 2020). In this thesis, the usefulness of the text uttered by the speaker to improve the robustness of speaker localization is investigated. This is a new task, which has not been reported in

the literature before. A method is proposed to use the short-duration wake-up word uttered by the target speaker to improve the localization performance in adverse conditions involving competing speakers, noise and reverberation. The proposed solution uses a speech recognition system to obtain phonetic alignments using the wake-up word and the corrupted speech signal. A "summary spectrum" representing the aligned phones is used to compute an identifier which is then used by a DNN to localize the speaker. We record and conduct experiments on both real and simulated data containing two competing speakers, in order to evaluate the performance of the proposed algorithm. We find that the textual information helps to reduce localization errors.

2. Given the speaker location, we separate the corresponding speech from the mixture in three stages. In the first stage, a simple delay-and-sum (DS) beamformer is used to enhance the target signal. In the second stage, a time-frequency mask corresponding to the localized speaker is obtained using a neural network with features extracted from the DS beamformed signal. This mask is used to derive an adaptive beamformer in the third stage. A thorough study of the impact of localization errors on speech separation is conducted. Experiments are conducted on a reverberated, noisy multichannel version of the well-studied WSJ0-2mix dataset using the word error rate (WER) as the metric. We show that the usage of true localization information drastically improves the WER and localization errors impact the source separation performance. To account for localization errors, a deflation-based strategy is proposed wherein the sources are estimated iteratively. At each iteration, the location of one speaker is estimated which is then used to estimate a mask corresponding to that speaker. The estimated source is removed from the mixture before estimating the location and mask of the next source. The proposed deflation-based approach is observed to reduce the impact of localization errors.
3. All DNNs in this thesis are trained in a supervised fashion using simulated data — an artificial mix of clean speech and noise. We wish to explain the experimental observation that speech enhancement models trained using synthetically generated noise can provide as good a performance as those trained with real noise while evaluating speech recognition performance on real data. To do so, we employ the SHAP method (Lundberg and Lee, 2017) to assign an importance to each dimension of the input feature vector using game theoretical approaches, where each input dimension is considered as a player in a game and the attributions as the distribution of the payout obtained from the game. In the context of speech enhancement, the dimension of the input feature refers to each time-frequency bin of the noisy speech spectrogram — the input to the DNN — and the payouts are the mask predictions.

   Attributing an importance to every time-frequency bin of the input signal gives us a way to visualize which aspects of input signal the network relies on to estimate the mask. Based on the argument that a network which generalizes well to unseen noise conditions should estimate mask using the time-frequency bins associated with speech and not those corresponding to the noise, we propose an objective

metric to evaluate the feature importance and use it to explain the effectiveness of the synthetically generated noise. To the best of our knowledge, this is the first study on neural network explainability in the context of speech enhancement.

Parts of the thesis have been published in the following articles:

1. **Sunit Sivasankaran**, Emmanuel Vincent, Dominique Fohr. Analyzing the impact of speaker localization errors on speech separation for automatic speech recognition. In *28th European Signal Processing Conference*, Jan 2021, Amsterdam, The Netherlands. (Accepted)
2. **Sunit Sivasankaran**, Emmanuel Vincent, Dominique Fohr. SLOGD: speaker location guided deflation approach to speech separation. In *45th IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 2020, Barcelona, Spain.
3. **Sunit Sivasankaran**, Emmanuel Vincent, Dominique Fohr. Keyword-based speaker localization: Localizing a target speaker in a multi-speaker environment. In *Interspeech*, Sep 2018, Hyderabad, India.

1.5.2 Outside the scope of the thesis

Though the proposed mechanisms can theoretically handle any number of interfering sources, all evaluations are conducted using a single interfering speech source in the presence of reverberation and noise. No acoustic echo is considered. Throughout the thesis, it is assumed that the speakers are in the far-field, i.e., the distance between the speakers and the center of the microphone array is significantly greater than the distance between the microphones.

1.6 Organization of the thesis

The rest of the thesis is organized as follows.

Chapter 2 sets up the problem and introduces the notations used in this thesis. An overview of the state-of-the-art for speaker localization, multichannel speech enhancement, and neural network explainability is provided. We also discuss the corresponding metrics used to evaluate the performance.

Chapter 3 deals with the problem of speaker localization in situations when the target speaker utters a known word or sentence such as the wake-up word of a distant-microphone voice command system.

Chapter 4 deals with separating speech from a mixture containing two speakers and noise using speaker location information.

Chapter 5 deals with interpretability of speech enhancement models.

Chapter 6 summarizes the thesis and provides future research directions.

2 State of the art

In this chapter the state of the art in different domains of the thesis is introduced. The problem setup is presented in Section 2.1. Section 2.2 describes different techniques used to localize a speaker. Section 2.3 describes different techniques used for speech enhancement and separation. Section 2.4 presents the state of the art methods used to analyze a DNN model.

2.1 General concepts

Mixture, sources, and spatial images: In the most general setting, the multichannel signal $\mathbf{x}(t) = [x_1(t), \ldots, x_I(t)]^T$ captured by an array of I microphones can be expressed as

$$\mathbf{x}(t) = \sum_{j=1}^{J} \mathbf{c}_j(t) \tag{2.1}$$

where J is the number of active sound sources and $\mathbf{c}_j(t) = [c_{1j}(t), \ldots, c_{Ij}(t)]^T$ is the spatial image of source j, i.e., the signal emitted by the source and captured at the microphones. The microphone index and the time index are denoted by i and t, respectively. This general formulation is valid for both point sources as well as diffuse noise. For point sources such as human speakers, the spatial image can be expressed as a linear convolution of the time-invariant[1] room impulse response (RIR) $\mathbf{a}_j(t) = [a_{1j}(t), \ldots, a_{Ij}(t)]^T$ and a single-channel source signal $s_j(t)$ as

$$\mathbf{c}_j(t) = \mathbf{a}_j \star s_j(t) \tag{2.2}$$

$$= \sum_{\tau=0}^{\infty} \mathbf{a}_j(\tau)\, s_j(t-\tau). \tag{2.3}$$

Time-frequency representation: Discrete time domain signals can be transformed to the frequency domain using the discrete Fourier transform (DFT), computed using the fast Fourier transform (FFT) algorithm (Oppenheim and Schafer, 2009)[2]. For non-stationary signals such as speech, the Fourier representation does not capture the temporal variations of the signal. In order to do so, signals are transformed into time-frequency representations. One such transform is the short-time Fourier transform (STFT).
The STFT of a signal x is obtained by first breaking it down to smaller segments called frames, using a window function $w(t)$. The DFTs of the windowed signals are stacked

[1] We assume that RIRs are time-invariant throughout the thesis.
[2] Assuming that the number of frequency bins is a power of 2.

together sequentially, forming an $F \times N$ complex-valued matrix with entries

$$x(n,f) = \sum_{t=0}^{T-1} x(n, t+nH) w(t) e^{-2j\pi tf/F}, \quad f \in \{0,\dots,F-1\} \tag{2.4}$$

where F is the number of frequency bins, T is the frame length or the length of the window function, $n \in \{0,\dots,N-1\}$ is the time frame index, H is the hop size and N is the total number of time frames. Throughout the thesis we use $F = T$.
The convolution in Eq. (2.3) can be rewritten in the STFT domain as

$$\mathbf{c}_j(n,f) = \sum_{f'=0}^{F-1} \sum_{n'=0}^{\infty} \mathbf{a}_j(n',f',f)\, s_j(n-n',f'). \tag{2.5}$$

Assuming that the window function reduces the influence of neighboring frequency bins, i.e., $\mathbf{a}_j(n',f,f') \approx \mathbf{0}$ for all $f' \neq f$, Eq. (2.5) can be approximated as

$$\mathbf{c}_j(n,f) \approx \sum_{n'=0}^{\infty} \mathbf{a}_j(n',f)\, s_j(n-n',f). \tag{2.6}$$

Another useful approximation is the narrowband approximation, which assumes that the length of the RIR is much shorter than the STFT window. In such cases the RIR will not "leak" across frames which gives us

$$\mathbf{c}_j(n,f) = \mathbf{a}_j(f)\, s_j(n,f). \tag{2.7}$$

Assuming that there are J' speech sources and that the remaining $J - J'$ sources are noise, Eq. (2.1) can then be written in time-frequency domain as

$$\mathbf{x}(n,f) = \mathbf{A}(f)\, \mathbf{s}(n,f) + \mathbf{u}(n,f) \tag{2.8}$$

where $\mathbf{A}(f) = [\mathbf{a}_1(f),\dots,\mathbf{a}_{J'}(f)]$ is the mixing matrix, $\mathbf{u}(n,f) = \sum_{j=J'+1}^{J} \mathbf{c}_j(n,f)$ are the noise sources and $\mathbf{s}(n,f) = [s_1(n,f),\dots,s_{J'}(n,f)]^T$.

Relative transfer functions and steering vectors: Each element of the mixing vector $\mathbf{a}_j(f)$ is the Fourier transform of the corresponding RIR and is referred to as an acoustic transfer function (ATF). The ratio between the ATFs of a microphone pair, say microphone $i \neq 1$ and microphone 1, is called a relative transfer function (RTF):

$$\tilde{a}_{i\,1\,j}(f) = \frac{1}{a_{1\,j}(f)}\, a_{i\,j}(f) \tag{2.9}$$

For notation simplicity and since the reference microphone is always the first microphone in this thesis, we set

$$\tilde{a}_{i\,j}(f) = \tilde{a}_{i\,1\,j}(f). \tag{2.10}$$

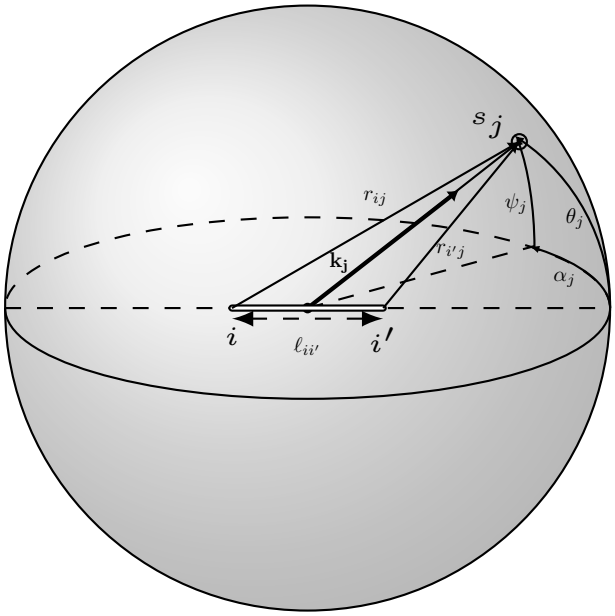


Figure 2.1: Position of the j-th source with respect to the microphone pair (i, i'). The polar coordinates of the source are denoted as (α_j, ψ_j, r_j): the azimuth, the elevation and the radial distance respectively. The direction of arrival is denoted as θ_j.

The RTF encodes the interchannel level difference (ILD) and the interchannel phase difference (IPD) between the two microphones, which are useful cues for source localization. In the absence of reverberation and early echoes, the ATF represents the direct component of a RIR, which is referred to as the steering vector $\mathbf{d}_j$:

$$\mathbf{d}_j(f) = \begin{bmatrix} \frac{1}{\sqrt{4\pi} r_{1j}} e^{-2j\pi r_{1\,j}\,\nu_{\mathrm{f}}\,/c} \\ \vdots \\ \frac{1}{\sqrt{4\pi} r_{Ij}} e^{-2j\pi r_{I\,j}\,\nu_{\mathrm{f}}\,/c} \end{bmatrix}. \tag{2.11}$$

where $r_{i\,j}$ is the distance between the j-th source and the i-th microphone as shown in Fig. 2.1, ν_{f} is the frequency in Hz corresponding to the f-th bin and c is the velocity of sound. In the far field, the difference between attenuation factors is negligible, giving us

$$\mathbf{d}_j(f) = \begin{bmatrix} e^{-2j\pi r_{1\,j}\,\nu_{\mathrm{f}}\,/c} \\ \vdots \\ e^{-2j\pi r_{I\,j}\,\nu_{\mathrm{f}}\,/c} \end{bmatrix}. \tag{2.12}$$

Similarly to the RTF, we can define a relative steering vector with respect to the first microphone as

$$\tilde{\mathbf{d}}_j(f) = \frac{1}{d_{1\,j}(f)}\,\mathbf{d}_j(f) \tag{2.13}$$

which can be expressed as

$$\tilde{\mathbf{d}}_j(f) = \begin{bmatrix} 1 \\ e^{-2j\pi\,\Delta_{2\,j}\,\nu_{\mathrm{f}}} \\ \vdots \\ e^{-2j\pi\,\Delta_{I\,j}\,\nu_{\mathrm{f}}} \end{bmatrix} \tag{2.14}$$

where $\Delta_{i\,j} = \frac{r_{i\,j} - r_{1\,j}}{c}$ is referred to as the time difference of arrival (TDOA).
In the far field, the TDOA can be defined using only the direction of arrival (DOA) θ_j of the source with respect to the microphone axis as

$$\Delta_{i\,j} = \frac{\ell_{i\,1}\cos\theta_j}{c} \tag{2.15}$$

where ℓ_{i1} is the distance between the i-th and the first microphone.
The focus of this thesis is to estimate θ_j and use it to estimate the spatial images of all speech sources ($\widehat{\mathbf{c}}_j$) from the mixture signal $\mathbf{x}$, containing multiple speakers and ambient noise.

2.2 Speaker localization

The spatial location of a speaker can be represented using its polar coordinates (α_j, ψ_j, r_j): the azimuth, elevation and radial distance of the source from the center of the microphone array, respectively, as shown in Fig. 2.1. Depending on the application, different aspects of the position may be relevant. To separate two speakers with different DOAs for example, it is often enough to estimate their DOAs. But if both speakers have the same DOA (but different radial distance), acoustic spotforming (Taseska and Habets, 2016) has to be employed which is not the focus of this thesis.
Speaker localization generally relies on the interchannel time difference (ITD) and interchannel level difference (ILD) clues encoded in the magnitude and phase spectra of the captured multichannel signals. It can also be done using single-channel signals (Deleforge et al., 2019b; El Badawy et al., 2017; Parhizkar et al., 2014), but it is then sensitive to the exact acoustic conditions and spectral characteristics of the sound source and the models do not generalize to different rooms. We therefore use multichannel signals for localization.
Sections 2.2.1 and 2.2.2 provide an overview of the signal processing based methods and learning-based methods used for speaker localization.

2.2.1 Signal processing based localization

Signal processing based source localization techniques can broadly be classified into angular spectrum based, clustering-based or subspace-based techniques.

Angular spectrum based methods compute the empirical correlation $\rho_{i\,i'}$ of the input signals captured at a pair of microphones as

$$\rho_{i\,i'}(n,k) = \sum_{f=0}^{F/2} \mathrm{Re}\{w(n,f)\, x_i(n,f)\, x_{i'}^*(n,f) e^{j2\pi kf/F}\} \tag{2.16}$$

where w is a frequency weighting function, k is the TDOA in samples, * denotes the conjugate of a complex number, and Re(.) denotes the real part of a complex number. Different weighting functions such as the smoothed coherence factor (SCOT) (Carter et al., 1973) and Roth correlation (Roth, 1971) have been proposed over the years but the popular phase transform (PHAT) weighting function

$$w(n,f) = \frac{1}{|\,x_i(n,f)||\,x_{i'}(n,f)|} \tag{2.17}$$

has stood the test of time. The resulting generalized cross-correlation with phase transform (GCC-PHAT) (Knapp and Carter, 1976) method computes the weighted cross-correlation for every frame as

$$\rho_{i\,i'}(n,k) = \sum_{f=0}^{F/2} \mathrm{Re}\left\{ \frac{x_i(n,f)\, x_{i'}^*(n,f)}{|\,x_i(n,f)||\,x_{i'}(n,f)|} e^{j2\pi kf/F} \right\} \tag{2.18}$$

$$= \sum_{f=0}^{F/2} \mathrm{Re}\{e^{j\phi_{i\,i'}(n,f)} e^{j2\pi kf/F}\} \tag{2.19}$$

where $\phi_{i\,i'}(n,f) = \angle\, x_i(n,f) - \angle\, x_{i'}(n,f)$ is the IPD, with $\angle(.)$ denoting the phase of a complex number.
Given a discrete set of TDOAs $k_{\min}, \ldots, k_{\max}$, the vector of GCC-PHAT values $\boldsymbol{\rho}_{i\,i'}(n) = [\rho_{i\,i'}(n,k_{\min}), \ldots, \rho_{i\,i'}(n,k_{\max})]^T$ can be written as:

$$\boldsymbol{\rho}_{i\,i'}(n) = \boldsymbol{\Xi}\, \mathbf{CSIPD}(n) \tag{2.20}$$

where $\boldsymbol{\Xi}$ is the sinusoidal subspace matrix

$$\boldsymbol{\Xi} = \begin{bmatrix} 1 & 0 & \cdots & \cos(2\pi k_{\min}\frac{f}{F}) & -\sin(2\pi k_{\min}\frac{f}{F}) & \cdots & \cos(\pi k_{\min}) & -\sin(\pi k_{\min}) \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\ 1 & 0 & \cdots & \cos(2\pi k_{\max}\frac{f}{F}) & -\sin(2\pi k_{\max}\frac{f}{F}) & \cdots & \cos(\pi k_{\max}) & -\sin(\pi k_{\max}) \end{bmatrix} \tag{2.21}$$

and $\mathbf{CSIPD}(n)$ is the vector of cosine-sine interchannel phase difference (CSIPD) features

$$\mathbf{CSIPD}(n) = \left[\cos\phi_{i\,i'}(n,0), \sin\phi_{i\,i'}(n,0), \ldots, \cos\phi_{i\,i'}\left(n,\frac{F}{2}\right), \sin\phi_{i\,i'}\left(n,\frac{F}{2}\right)\right]^T. \tag{2.22}$$

In essence, $\boldsymbol{\rho}_{i\,i'}(n)$ is obtained by linear transformation of the CSIPD feature vector. The TDOA for frame n can then be obtained as

$$\mathrm{TDOA}(n) = \arg\max_k \rho_{i\,i'}(n,k). \tag{2.23}$$

When the source is not moving, the robustness of the TDOA estimates can be improved by averaging the estimates $\mathrm{TDOA}(n)$ over time or by pooling the objectives $\boldsymbol{\rho}_{i\,i'}(n)$ over time before picking the maximum.
A related approach is to compute the energy of the signal arriving from every direction by simple delay-and-sum beamforming as

$$\widehat{s}(n,f,\theta) = \mathbf{d}^H(\theta,f)\,\mathbf{x}(n,f) \tag{2.24}$$

where $\mathbf{d}$ is the steering vector defined in Eq. (2.14) steered towards the direction θ. The resulting steered response power (SRP)

$$\mathrm{SRP}(n,\theta) = \sum_f |\widehat{s}(n,f,\theta)|^2 \tag{2.25}$$

can be used to obtain the position of the source as

$$\widehat{\theta}(n) = \operatorname*{argmax}_\theta \mathrm{SRP}(n,\theta). \tag{2.26}$$

Similarly to Eq. (2.17), the SRP can also be weighted using PHAT to obtain the popular SRP-PHAT objective (Dibiase, 2000). It should be noted that SRP-PHAT is a generalization of GCC-PHAT for $I \geq 2$ microphones. For $I = 2$, SRP-PHAT and GCC-PHAT are equivalent.

Subspace-based approaches are an alternative class of algorithms for source localization. A popular subspace-based approach is the multiple signal classification (MUSIC) (Schmidt, 1986) algorithm, which works by first computing the so-called MUSIC spectrum $\mathcal{P}(\theta)$ as

$$\mathcal{P}(\theta) = \frac{1}{\mathbf{d}(\theta,f)\,\mathbb{Q}_{\mathbf{u}}\,\mathbb{Q}_{\mathbf{u}}^H\,\mathbf{d}^H(\theta,f)} \tag{2.27}$$

where $\mathbb{Q}_{\mathbf{u}}$ is the matrix of the eigenvectors of the signal covariance matrix corresponding to the $J - J'$ lowest eigenvalues. The speaker directions are obtained from the peaks of the spectrum.
More recent subspace-based algorithms (Malioutov et al., 2005; Asaei et al., 2016; Gretsistas and Plumbley, 2010; Yin and Chen, 2011) define an over-complete dictionary $\mathbf{D}(f)$ containing the steering vectors corresponding to all possible spatial positions and compute a spatially sparse representation of $\mathbf{x}$ by

$$\min_{\mathbf{s}} \sum_{n,f} \|\,\mathbf{x}(n,f) - \mathbf{D}(f)\,\mathbf{s}(n,f)\|_2^2 + \lambda\Omega(\mathbf{s}) \tag{2.28}$$

where $\mathbf{s}(n, f)$ is the vector of source signals associated with these positions, $\|.\|_p$ is the l^p norm of a vector, λ is the regularization parameter, and $\Omega(\mathbf{s})$ is a structured sparsity constraint which ensures that the source signals are nonzero for a few positions only. Alternative subspace-based methods for localization are reported by Pavlidi et al. (2013) and Griffin et al. (2012). They pick up neighboring sets of time-frequency bins called zones and estimate the DOA in each zone assuming the dominance of a single source in the zone. The histogram of DOAs contains peaks corresponding to the DOAs of the sources. The DOAs of the sources are estimated one at a time using a dictionary-based approach. Each estimated source is removed from the histogram before estimating the DOA of the next source.

Clustering-based approaches for source localization work by iteratively estimating time-frequency masks representing the contribution of each source to every time-frequency bin and using these masks to reestimate the TDOAs. The masks can either be binary masks (Sawada et al., 2007) where each time-frequency bin is associated to the closest source or soft masks based on some probabilistic model. Araki et al. (2009) discretize the TDOA space into K bins and model the phase difference in each time-frequency bin using a Gaussian mixture model (GMM) as

$$p(\phi_{i\,i'}(n,f);\boldsymbol{\mu},\boldsymbol{\sigma}) = \sum_{k=1}^{K}\sum_{m=-M_f}^{M_f} \frac{\alpha_k}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{(\phi_{i\,i'}(n,f)+2\pi m-2\pi\mu_k f/F)^2}{2\sigma_k^2}\right) \tag{2.29}$$

where m is an integer to handle spatial aliasing, M_f is a function of the frequency and the distance between the microphones ℓ, and the k-th TDOA bin is modeled by a single Gaussian with mean μ_k, variance σ_k and weight α_k. A Dirichlet distribution (Bishop, 2006) is used as a sparsity-promoting prior over α_k to prevent each source from being modeled by multiple Gaussians.

Shortcomings of signal processing based methods: GCC-PHAT based algorithms have proven to be robust over the years and have shown to work well even in challenging acoustic conditions (Evers et al., 2020; Deleforge et al., 2019a). Figure 2.2 shows the GCC-PHAT angular spectrum as a function of the TDOA k for various signals. The peak(s) in each curve point(s) to the TDOAs of the source(s). The blue curves are for a single speaker at a distance of 0.6 m from the center of the microphone pair while the red curves are for a speaker positioned at a distance of 1.1 m. The distance between the two microphones is 0.226 m. The dotted lines represent the GCC-PHAT spectra computed for the direct components of the spatial images (involving direct path signals only) whereas the solid lines represent the GCC-PHAT spectra computed for the full spatial images (also involving early echoes and reverberation). A single dominant peak can be seen when only the direct path is present. In the presence of reverberation, the peaks are distorted, thereby introducing uncertainty in the TDOA estimation. The green curve shows the angular spectrum when both speakers are simultaneously speaking in the presence of noise. Though clear peaks can be seen at positions corresponding to both

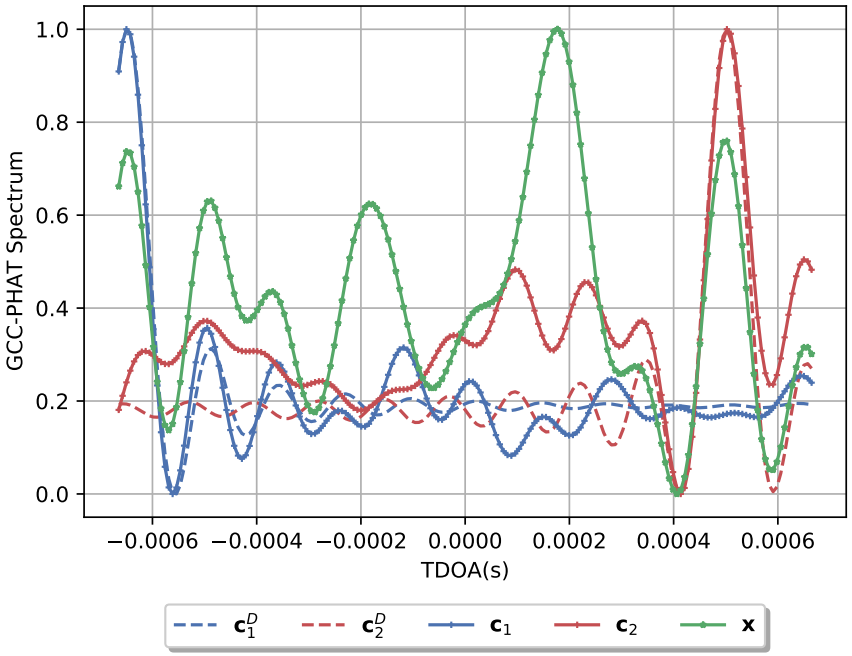


Figure 2.2: GCC-PHAT angular spectra computed for the direct components of the spatial images of two speakers $\mathbf{c}_1^D$ and $\mathbf{c}_2^D$, the full spatial images $\mathbf{c}_1$ and $\mathbf{c}_2$, and the mixture $\mathbf{x} = \mathbf{c}_1 + \mathbf{c}_2 + \mathbf{u}$. The speakers are in a room with $T_{60} = 0.47$ s and dimension of $[3.7, 3.67, 2.5]$ m. The duration of all signals is 5.16 s.

speakers, the highest peak in the curve points to neither of the speaker TDOAs. This shows the difficulty in localizing the speakers in the presence of noise, reverberation and multiple speakers. Other localization algorithms face similar difficulties in the presence of noise and reverberation. Due to these limitations, learning-based methods are often used for source localization instead.

2.2.2 Learning-based localization

Learning-based methods map high-dimensional input features extracted from $\mathbf{x}$ to a low-dimensional manifold parameterized by the DOA.

Pre-DNN approaches: Nishino and Takeda (2008) train a GMM for every source direction and estimate the source location from the posterior probabilities computed during evaluation. Mouba and Marchand (2006) and May et al. (2011) use GMMs to learn the interdependency of ITDs and ILDs, while Kayser and Anemüller (2014) use discriminative support vector machines with GCC patterns. A manifold learning approach for source localization based on the concept of diffusion maps (Talmon et al., 2012) is inves-

tigated by Laufer et al. (2016), Talmon et al. (2011), and Laufer et al. (2013). Similar ideas are used by Deleforge et al. (2014, 2015) to study the low-dimensional manifold of the high-dimensional input feature space. It was shown that the low-dimensional manifold is a smooth 2-dimensional manifold, parameterized by the DOA. Another interesting approach proposed by Marchand and Vialard (2009) is to convert the ILD or IPD cues of binaural signals to an image. The Hough transform can then be used to find a line in the image whose slope gives the DOA of the source.

DNN-based localization has become the most popular approach in the last few years. It relies on a DNN to learn a mapping between the features extracted from the input signal $\mathbf{x}$ and the space of physical locations which can either be discrete or continuous. In the discrete case, a finite set of locations is assumed and every location is assigned a class number $p \in \{1, \ldots, \mathrm{P}\}$. The DNN is trained as a classifier and learns to estimate the posterior probability of every class p given the input signal $\mathbf{x}$. In the continuous case the DNN is trained to directly predict the spatial coordinates of the speaker. DNN-based methods can also be used to estimate the high-dimensional RTF (Wang et al., 2018a; Laufer et al., 2016), which can then be used to infer the speaker location.
DNNs are known for their generalization capabilities and can be trained on a large number of different room configurations. This is typically obtained by simulating RIRs and convolving them with undistorted acoustic sources such as speech recorded in anechoic conditions. Simulated data has the added advantage of creating a balanced dataset involving a uniform distribution of speaker locations which is useful for learning-based methods. It must be ensured that the source is active at all time instants which is sometimes hard in the case of human speech which involves silence intervals between utterances. To overcome this problem, Chakrabarty and Habets (2017a,b) used white noise as the acoustic source and the trained network surprisingly generalized for speech localization.
Different features have been used for source localization. Chakrabarty and Habets (2017b) used the concatenation of the phases of all channels, called a phasemap, along with a convolutional neural network (CNN) whereas Vesperini et al. (2016) and Xiao et al. (2015) used features extracted from GCC-PHAT patterns and Salvati et al. (2018) used the SRP spectrum as input. Tashev et al. (2017) uses both the phase differences and the magnitude spectra across multiple channels as inputs, while Takeda and Komatani (2016) use the dominant eigenvector of the covariance matrix $\mathbf{R}_{\mathbf{x}}$ at each frequency which encodes the IPD and ILD. Adavanne et al. (2018) concatenate the magnitude and phase of the STFT of the input signal and provide them as inputs to the network. Localization was also done using the raw audio waveform by Vecchiotti et al. (2019), with the objective that the network learns to extract relevant features for localization.

Localizing multiple speakers using DNNs: For localizing multiple speakers, DNNs are usually trained as classifiers (Chakrabarty and Habets, 2017b; Perotin et al., 2019b; Takeda and Komatani, 2016; Ma et al., 2015; Adavanne et al., 2018). One approach is to create new output dimensions for each speaker (Takeda and Komatani, 2016). Cross-

entropy can then be used to train the network since each source location is independent of the others. The so-called block-wise consistent labeling was used to ensure label consistency across each output dimensions. An alternative approach is to use a sigmoid nonlinearity for each class and compute the presence or absence of a source in each direction by training the network using binary cross-entropy (BCE) as the cost function (Perotin et al., 2019b; Chakrabarty and Habets, 2017b). The classes corresponding to the J' highest values indicate the speaker locations. A network trained to localize a single speaker can also be used to localize two speakers albeit not as robustly as a network trained to localize two speakers (Perotin, 2019). All the above works assume the knowledge of the number of active speakers, J'.

Training DNN-based localization models: Multilayer perceptrons (MLPs) (Xiao et al., 2015; Takeda and Komatani, 2016; Ma et al., 2015), convolutional neural networks (CNNs) (Vecchiotti et al., 2019; Chakrabarty and Habets, 2017a,b; Nguyen et al., 2018), and combinations of CNNs and recurrent neural networks (CRNNs) (Adavanne et al., 2018; He et al., 2018b; Perotin et al., 2019a) are the preferred choices of neural network architectures for localization. For a localization network trained as a classifier, cross-entropy (Chakrabarty and Habets, 2017a), mean squared error (MSE) with soft Gibbs target (He et al., 2018a) and Gibbs-weighted loss are used as cost functions whereas for regression MSE (Vesperini et al., 2016), angular loss and regression with Cartesian targets (Adavanne et al., 2019) are used. A comparison of regression and classification for DNN-based localization with different cost functions was conducted by Perotin et al. (2019a). It was shown that a DNN trained as a classifier outperforms its regression counterpart in the presence of noise.

2.2.3 Speaker localization performance metrics

In this thesis, it is assumed that the number of speech sources is $J' = 2$. In our localization experiments (Chapter 3), there is one target speaker and one interfering speaker. Three different metrics are used to evaluate the performance of the localization system, namely:

1. The gross error rate (GER) measures the percentage of estimated DOAs whose difference with the true target DOA is above an error threshold θ_{Th}:

$$f_{\mathrm{GER}}(j,k) = \begin{cases} 1 & \text{if } |\widehat{\theta}_j(k) - \theta_j(k)| > \theta_{\mathrm{Th}} \\ 0 & \text{otherwise} \end{cases} \tag{2.30}$$

$$\mathrm{GER}(j) = \frac{1}{\mathrm{N_{ET}}} \sum_{k=1}^{\mathrm{N_{ET}}} f_{\mathrm{GER}(j,k)} \tag{2.31}$$

where $\theta_j(k)$ and $\widehat{\theta}_j(k)$ are the real and estimated DOAs of the target speaker as shown in Fig. 2.3 for the k-th test sample, and $\mathrm{N_{ET}}$ is the number of test samples.

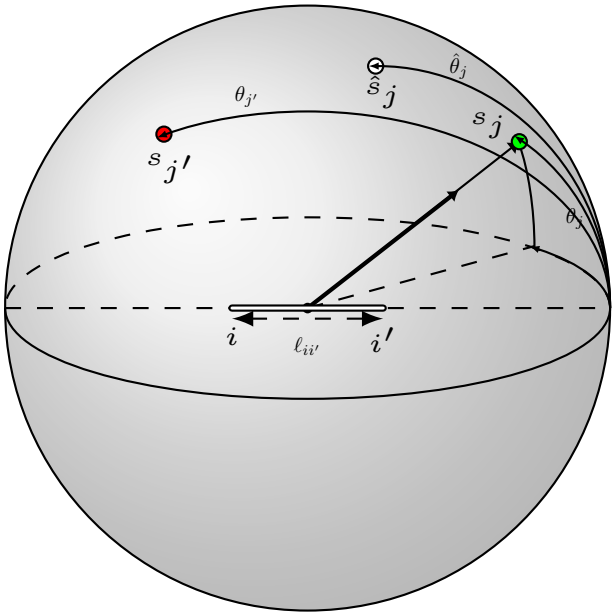


Figure 2.3: True (θ_j) and estimated ($\widehat{\theta}_j$) DOAs of the target speaker along with the true DOA of the interfering speaker ($\theta_{j'}$).

2. The interference closeness rate (ICR) measures the percentage of estimated DOAs which are close to the interference DOA, thereby misidentifying the speaker as interference j' instead of target j. It is computed as

$$f_{\mathrm{ICR}}(k,j,j') = \begin{cases} 1 & \text{if } |\,\widehat{\theta}_j(k) - \theta_{j'}(k)| < \theta_{\mathrm{Th}} \\ 0 & \text{otherwise} \end{cases} \tag{2.32}$$

$$\mathrm{ICR}(j,j') = \frac{1}{\mathrm{N_{ET}}} \sum_{k=1}^{\mathrm{N_{ET}}} f_{\mathrm{ICR}(k,j,j')} \tag{2.33}$$

3. The mean absolute error (MAE) is the average of the absolute error in degrees made by the system while localizing the target:

$$\mathrm{MAE}(j) = \frac{1}{\mathrm{N_{ET}}} \sum_{k=1}^{\mathrm{N_{ET}}} |\,\widehat{\theta}_j(k) - \theta_j(k)| \tag{2.34}$$

The goal is to achieve the lowest possible value for all three metrics.

2.3 Speech separation

The problem of speech separation has been widely studied in the literature. A large number of methods rely on the concept of time-frequency mask, which is introduced in

Section 2.3.1. Methods for single-channel separation are presented in Section 2.3.2. We then introduce the general principles of multichannel filtering in Section 2.3.3 and deep learning based methods for multichannel separation in Section 2.3.4.

2.3.1 Time-frequency masks

Speech signals are naturally sparse in the time-frequency domain, i.e., they occupy only a small proportion of time-frequency bins. In the presence of multiple speech signals, the probability that the signals occupy the same time-frequency bins is low.
A two-dimensional binary-valued mask representing source dominance per time-frequency bin is referred to as an ideal binary mask (IBM) and defined as

$$\mathcal{M}_j^{\text{IBM}}(n,f) = \begin{cases} 1 & \text{if } |\, c_j(n,f)| > |\, c_{j'}(n,f)| \quad \forall j' \in \{1,\dots,J\}, \quad j \neq j' \\ 0 & \text{otherwise} \end{cases} \tag{2.35}$$

where $|.|$ represents the magnitude spectrum of the signal.
A softer version of the IBM is the ideal ratio mask (IRM) defined as

$$\mathcal{M}_j^{\text{IRM}}(n,f) = \frac{|\, c_j(n,f)|^p}{\sum_{j'=0}^{J} |\, c_{j'}(n,f)|^p}. \tag{2.36}$$

The value of p is often fixed to $p = 1$ in the literature. With $p = 2$, the IRM becomes the well known single-channel Wiener filter.
Another interesting mask is the phase-sensitive mask (Erdogan et al., 2015) which takes into account the phase of the source. It is defined as

$$\mathcal{M}_j^{\text{PS}}(n,f) = \frac{|\, c_j(n,f)|}{|\, x(n,f)|} \cos(\angle s_j(n,f) - \angle x(n,f)). \tag{2.37}$$

A related concept is the speech presence probability (SPP) (Martin and Cohen, 2018). If $\mathcal{H}_1$ is the hypothesis that the source of interest is present and $\mathcal{H}_0$ denotes the hypothesis that this source is absent, the posterior SPP can be defined as

$$p(\mathcal{H}_1(n,f)|\, x(n,f)) = \frac{p(x(n,f)|\mathcal{H}_1(n,f))p(\mathcal{H}_1(n,f))}{p(x(n,f)|\mathcal{H}_1(n,f))p(\mathcal{H}_1(n,f)) + p(x(n,f)|\mathcal{H}_0(n,f))p(\mathcal{H}_0(n,f))} \tag{2.38}$$

$$= \frac{\varrho(n,f)}{1+\varrho(n,f)} \tag{2.39}$$

where ϱ is the generalized likelihood ratio defined by

$$\varrho(n,f) = \frac{p(x(n,f)|\mathcal{H}_1(n,f))p(\mathcal{H}_1(n,f))}{p(x(n,f)|\mathcal{H}_0(n,f))p(\mathcal{H}_0(n,f))}. \tag{2.40}$$

In practice, the true masks $\mathcal{M}_j$ are unknown at test time, and some estimated masks $\widehat{\mathcal{M}}_j$ must be computed from $\mathbf{x}$ instead. In the single-channel case, the source signals can be directly derived by

$$\widehat{c}_j(n,f) = \widehat{\mathcal{M}}_j(n,f)\, x(n,f). \tag{2.41}$$

The quality of the estimated signal can further be improved using the Griffin-Lim algorithm (Griffin and Lim, 1984). In the multichannel case, better separation quality can be achieved by multichannel filtering (see Section 2.3.3 below).

2.3.2 Single-channel speech separation

Nonnegative matrix factorization (NMF) (Lee and Seung, 1999) is the most popular method for single-channel separation prior to the emergence of deep learning. The magnitude spectrogram $|\mathbf{X}|$ of the speech mixture in the time-frequency domain, an $F \times N$ matrix with nonnegative entries $|x(n,f)|$, is approximated as the product of two nonnegative matrices as

$$|\mathbf{X}| \approx \mathbf{G}\mathbf{H} \tag{2.42}$$

where $\mathbf{G} = [\mathbf{g}_1, \ldots, \mathbf{g}_K] \in \mathbb{R}_+^{F \times K}$ is a set of basis spectra representing the spectral structure and $\mathbf{H} = [\mathbf{h}_1^T, \ldots, \mathbf{h}_K^T]^T \in \mathbb{R}_+^{K \times N}$ are the corresponding time-varying activations. The magnitude spectrogram $|\widehat{\mathbf{C}}_j|$ of each source with entries $|\widehat{\mathbf{c}}_j(n,f)|$ can be estimated as

$$|\widehat{\mathbf{C}}_j| = \sum_{k \in \mathcal{K}_j} \mathbf{g}_k \mathbf{h}_k \tag{2.43}$$

where $\mathcal{K}_j$ are the component indexes associated with the j-th source. The IRM for each source can then be estimated as in Eq. (2.35).
The basis spectra can either be learned in an unsupervised fashion (Smaragdis and Brown, 2003), i.e., directly obtained from the mixture x or in a supervised fashion (Smaragdis, 2007), i.e., precomputed using a set of training samples. They can also be obtained in a semi-supervised fashion where some of the basis spectra are learned from training data and others are obtained from the mixture (Mysore and Smaragdis, 2011). NMF-based models are linear models and they fall short of accurately capturing the temporal variation complexities of speech. They were shown to be outperformed by DNNs by Le Roux et al. (2015) and Nugraha et al. (2016) among others.

DNN-based methods in the time-frequency domain Multiple DNN-based methods have been proposed for speech separation using STFT-based features as input. Deep clustering (Hershey et al., 2016; Isik et al., 2016) trains a two-layer bidirectional long short-term memory (Bi-LSTM) network to output an embedding for each time-frequency bin with the objective of forming clusters in the embedding space. The clustering objective is enforced by training the network via the following loss function:

$$\mathcal{L}_{\text{DC}} = \sum_{n,f} \| \mathcal{E}(x(n,f))\, \mathcal{E}(x(n,f))^T - \mathbb{I}^{\mathbf{IBM}}(n,f)\, \mathbb{I}^{\mathbf{IBM}}(n,f)^T \|_F^2 \tag{2.44}$$

where $\mathcal{E}(x)$ is the DNN embedding of the mixture x, $\mathbb{I}^{\mathbf{IBM}}$ is the one-hot representation of the ground truth IBM and $\|.\|_F$ is the Frobenius norm. At test time, clusters in embedding space are formed using the k-means algorithm. Each cluster is assigned to a source and an IBM for that source is constructed using the corresponding time-frequency

bins. The number of clusters, which corresponds to the number of sources, is assumed to be known. The use of deep clustering as a preprocessing step was shown to improve the ASR performance for mixtures of two speakers (Menne et al., 2019b). A related idea is the deep attractor network (Chen et al., 2017) which creates so-called attractor points in the projected space. The attractor points pull together the time-frequency bins of the corresponding sources, thereby creating clusters that can be used to extract IBMs for the sources.

Permutation invariant training (PIT) (Yu et al., 2017; Kolbæk et al., 2017) was proposed as an alternative approach where a CNN or an MLP with J' output heads is trained to estimate the IRMs of all sources simultaneously. During training, each output head $\widehat{\mathcal{M}}_j$ is matched with the source j' with the lowest magnitude spectrum approximation (MSA) loss

$$\mathcal{L}(\widehat{\mathcal{M}}_j, j') = \sum_{n,f} |\, \widehat{\mathcal{M}}_j(n,f)\, x(n,f) - c_{j'}(n,f)|^2. \tag{2.45}$$

Losses are summed across all the output heads to compute the gradients for training the network as

$$\mathcal{L}_{\text{PIT}} = \min_{\mathbb{P}} \sum_{j,j'} \mathcal{L}(\widehat{\mathcal{M}}_j, \mathbb{P}(j')) \tag{2.46}$$

where $\mathbb{P}(.)$ is a permutation of the source indices $\{1, \ldots, J\}$. The deep clustering loss and the MSA loss can be combined to obtain better speech separation performance (Wang et al., 2018c).

Another interesting approach is to iteratively estimate the IRM for each source (Kinoshita et al., 2018) using a deflation-based strategy. A summary of the method is shown in Algorithm 1. Given the input mixture x and a remainder mask initialized to 1 for all (n, f), a DNN is used to estimate a mask $\widehat{\mathcal{M}}$. The source corresponding to the estimated mask is chosen based on the one having the least loss value computed using Eq. (2.45), similar to the PIT loss. The remainder mask is updated by the removing the estimated mask as in Eq. (2.48).

All these methods operate using the magnitude spectrum of the signal as input to the DNN while neglecting the phase information of the source and using the phase information of the mixture to reconstruct it. Some recent works try to explicitly reconstruct the source phase information (Le Roux et al., 2019a; Wang et al., 2018e, 2019).

DNN-based methods from the raw waveform: A recent trend in the speech separation community is to train networks using the raw waveform as input instead of features extracted from it (Luo and Mesgarani, 2019; Shi et al., 2019; Zhang et al., 2020; Pariente et al., 2020). The advantage of such systems is that the phase reconstruction of the source is not a bottleneck anymore. The general idea is to create trainable encoder, separator and decoder blocks connected in sequence. The separator estimates a mask which is multiplied by the encoder output before feeding it to the decoder as shown in Fig. 2.4. The encoder and decoder can be seen as counterparts to the STFT and the inverse STFT in traditional approaches.

Algorithm 1: Iterative estimation of source IRMs using DNN (Kinoshita et al., 2018).

Input: Input data x, Remainder mask $\mathcal{M}_R^{\text{IRM}}(n,f) = 1 \quad \forall (n,f)$, DNN
Output: $\mathcal{M}_j^{\text{IRM}} \quad \forall j \in \{1,\ldots,J'\}$
for $j \in \{1,\ldots,J'\}$ **do**

1. Estimate a mask $\widehat{\mathcal{M}}$ with DNN using x and remainder mask $\mathcal{M}_R^{\text{IRM}}$
2. Find the source index which has the least MSE with the estimated mask:

$$\widehat{j} = \underset{j \in \{1,\ldots J'\}}{\operatorname{argmin}} \parallel \widehat{\mathcal{M}}\, x - c_j \parallel_F^2 \tag{2.47}$$

3. Remove the source from the remainder mask as

$$\mathcal{M}_R^{\text{IRM}} = \max(\mathcal{M}_R^{\text{IRM}} - \widehat{\mathcal{M}}, 0) \tag{2.48}$$

4. $\mathcal{M}_{\widehat{j}}^{\text{IRM}} \leftarrow \widehat{\mathcal{M}}$
5. Remove $\widehat{j}$ from the source index set.

end

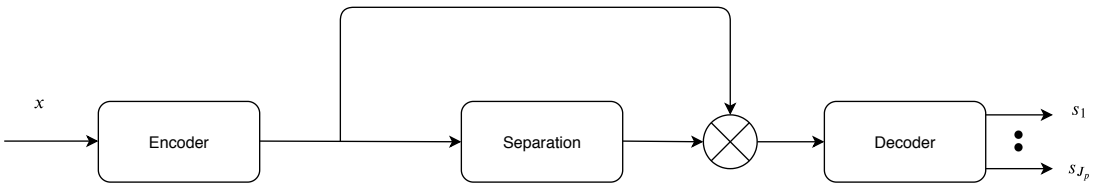


Figure 2.4: End-to-end speech separation framework.

Different choices of architecture were used for each of these components. In Conv-TasNet (Luo and Mesgarani, 2019), a 1-dimensional CNN was used as the encoder and the decoder. A series of CNNs with skip connections was used as the separator. Zhang et al. (2020) used a 1-dimensional CNN with parametric ReLU as the encoder, a series of gated dilated temporal CNNs as the separator and an MLP as the decoder. In order to model both short- and long-term structure, Luo et al. (2020) proposed to replace CNNs with dual-path recurrent neural networks (DPRNN), which represent information both within a frame and across frames. The cost function to train waveform-based methods is either the source-to-distortion ration (SDR) (Vincent et al., 2006) defined as

$$\text{SDR} = 10 \log_{10} \frac{\parallel c_j \parallel^2}{\parallel \widehat{c}_j - c_j \parallel^2} \tag{2.49}$$

or the scale invariant SDR (SI-SDR) (Le Roux et al., 2019b) defined as

$$\text{SI-SDR} = 10\log_{10}\frac{\left\|\frac{<\widehat{c}_j, c_j>}{\|c_j\|^2}\, c_j\right\|^2}{\left\|\widehat{c}_j - \frac{<\widehat{c}_j, c_j>}{\|c_j\|^2}\, c_j\right\|^2} \tag{2.50}$$

where $<.,.>$ denotes the inner product of two signals.
Large gains have been reported for speech separation using waveform-based approaches, albeit in unrealistic acoustic conditions containing no noise and reverberation.

Datasets used for evaluating single-channel speech separation: Single-channel speech separation experiments are typically conducted using the WSJ0-2mix and/or WSJ0-3mix (Isik et al., 2016) datasets which contain mixtures simulated using the Wall Street Journal (WSJ) dataset. A noisy version of the dataset was recently proposed by Wichern et al. (2019). Menne et al. (2019b) proposed to add naturalness to the WSJ02mix utterances by including copious amount of silence in between words. None of these datasets contains reverberation which is critical to evaluate performance in far-field situations.

2.3.3 Multichannel speech separation

Multichannel speech enhancement and speech separation rely on beamforming. A beamformer is a linear filter used to recover signals from microphone arrays (or any array in general (Van Trees, 2002)). In this section, without loss of generality, we assume that we wish to recover the spatial image of the first source $\mathbf{c}_1$ and that all other sources are collectively considered as noise $\mathbf{v}$, i.e.,

$$\mathbf{x} = \mathbf{c}_1 + \mathbf{v} \tag{2.51}$$

$$\mathbf{v} = \mathbf{u} + \sum_{j=2}^{J} \mathbf{c}_j\,. \tag{2.52}$$

A beamformer can be used to estimate the desired source from the mixture as

$$\widehat{c}_1(n,f) = \mathbf{w}^H(n,f)\,\mathbf{x}(n,f) \tag{2.53}$$

where $\mathbf{w}(f) = [\mathrm{w}_1(n,f), \ldots, \mathrm{w}_I(n,f)]^T$ is a possibly time-varying, frequency-dependent, complex-valued vector. Beamformers can also be applied in the time domain. However, since the models considered in this thesis operate in the time-frequency domain, the rest of the discussion regarding beamformers focuses on the time-frequency domain.
Beamformers can broadly be categorized into data-independent and data-dependent beamformers.

Data-independent beamformers do not rely on the signal $\mathbf{x}$ to estimate the beamformer but rather on the position $\mathbf{p}_j$ of the speaker. A classical example of data-

independent beamformer is the delay-and-sum (DS) beamformer defined by

$$\widehat{c}_1(n,f) = \frac{1}{I}\sum_{i=1}^{I} x_i(n,f) e^{j2\pi\,\Delta_{i\,1}\,\nu_{\mathrm{f}}} \tag{2.54}$$

$$= \frac{1}{I}\tilde{\mathbf{d}}^H(f)\,\mathbf{x}(n,f) \tag{2.55}$$

where $\Delta_{i\,1}$ is the TDOA of the target source at the i-th microphone with respect to the first microphone and $\tilde{\mathbf{d}}^H(f)$ is the relative steering vector. The popular Beamformit speech enhancement method (Anguera et al., 2007) relies on a time-varying weighted version of the DS beamformer.

Adaptive beamformers: Adaptive or data-dependent beamformers rely on the mixture $\mathbf{x}$ to estimate $\mathbf{w}$. Assuming that the signal at the output of microphone 1 is of interest, the multichannel Wiener filter (MWF) (Doclo and Moonen, 2002) is obtained using a minimum mean square error criterion (MMSE) as

$$\mathbf{w}_{\mathrm{MWF}}(n,f) = \underset{\mathbf{w}}{\operatorname{argmin}}\,\mathbb{E}[|\mathbf{w}^H(n,f)\,\mathbf{x}(n,f) - s_1(n,f)|^2] \tag{2.56}$$

$$= \underset{\mathbf{w}}{\operatorname{argmin}}\,\underbrace{|1-\mathbf{a}_1^H(f)\,\mathbf{w}(n,f)|^2\sigma_{s_1}^2(n,f)}_{\text{speech distortion}} + \underbrace{\mathbf{w}^H(n,f)\,\mathbf{R}_{\mathbf{v}}(n,f)\,\mathbf{w}(n,f)}_{\text{noise power}} \tag{2.57}$$

$$= \frac{\sigma_{s_1}^2(n,f)\,\mathbf{R}_{\mathbf{v}}^{-1}(n,f)\,\mathbf{a}_1(f)}{1+\sigma_{s_1}^2(n,f)\,\mathbf{a}_1^H(f)\,\mathbf{R}_{\mathbf{v}}^{-1}(n,f)\,\mathbf{a}_1(f)} \tag{2.58}$$

where $\sigma_{s_1}^2(n,f)$ is the variance of the source signal and $\mathbf{R}_{\mathbf{v}}(n,f)$ is the noise covariance. The objective is to reduce the distortion and the power of the noise in the mixture. A tradeoff parameter μ between speech distortion and noise reduction can be introduced to obtain the speech distortion weighted MWF (SDW-MWF) (Doclo et al., 2007) as

$$\mathbf{w}_{\text{SDW-MWF}}(n,f) = \underset{\mathbf{w}}{\operatorname{argmin}}\,|1-\mathbf{a}_1^H(f)\,\mathbf{w}(n,f)|^2\sigma_{s_1}^2(n,f) + \mu\,\mathbf{w}^H(n,f)\,\mathbf{R}_{\mathbf{v}}(n,f)\,\mathbf{w}(n,f) \tag{2.59}$$

$$= \frac{\sigma_{s_1}^2(n,f)\,\mathbf{R}_{\mathbf{v}}^{-1}(n,f)\,\mathbf{a}_1(f)}{\mu+\sigma_{s_1}^2(n,f)\,\mathbf{a}_1^H(f)\,\mathbf{R}_{\mathbf{v}}^{-1}(n,f)\,\mathbf{a}_1(f)}. \tag{2.60}$$

Note that, when $\mu = 1$, $\mathbf{w}_{\text{SDW-MWF}}(n,f)$ is equal to $\mathbf{w}_{\mathrm{MWF}}(n,f)$. Altering the value of μ leads to different types of beamformers. The minimum variance distortionless response (MVDR) beamformer (Gannot et al., 2001) is obtained in the limit when $\mu \to 0$ as

$$\mathbf{w}_{\mathrm{MVDR}}(n,f) = \frac{\mathbf{R}_{\mathbf{v}}^{-1}(n,f)\,\mathbf{a}_1(f)}{\mathbf{a}_1^H(f)\,\mathbf{R}_{\mathbf{v}}^{-1}(n,f)\,\mathbf{a}_1(f)} \tag{2.61}$$

which is the solution to the following optimization problem:

$$\mathbf{w}_{\mathrm{MVDR}}(n,f) = \underset{\mathbf{w}}{\operatorname{argmin}}\,\mathbf{w}^H(n,f)\,\mathbf{R}_{\mathbf{v}}(n,f)\,\mathbf{w}(n,f) \quad \text{s.t.} \quad \mathbf{a}_1^H(f)\,\mathbf{w}(n,f) = 1. \tag{2.62}$$

This beamformer is a special case of the more general linearly constrained minimum variance (LCMV) beamformer, which constrains the response of the beamformer in several directions. A closely related beamformer is the minimum power distortionless response (MPDR) beamformer

$$\mathbf{w}_{\text{MPDR}}(n,f) \leftarrow \underset{\mathbf{w}}{\text{argmin}}\, \mathbf{w}^H(n,f)\,\mathbf{R}_{\mathbf{x}}(n,f)\,\mathbf{w}(n,f) \quad \text{s.t.} \quad \mathbf{a}_1^H(f)\,\mathbf{w}(n,f) = 1 \tag{2.63}$$

$$\mathbf{w}_{\text{MPDR}}(n,f) = \frac{\mathbf{R}_{\mathbf{x}}^{-1}(n,f)\,\mathbf{a}_1(f)}{\mathbf{a}_1^H(f)\,\mathbf{R}_{\mathbf{x}}^{-1}(n,f)\,\mathbf{a}_1(f)} \tag{2.64}$$

where $\mathbf{R}_{\mathbf{x}}(n,f)$ is the covariance of the mixture $\mathbf{x}(n,f)$.
A maximum SNR beamformer (Cox et al., 1987) which maximizes the output SNR can be computed as

$$\mathbf{w}_{\text{SNR}}(n,f) = \underset{\mathbf{w}}{\text{argmax}}\, \frac{|\,\mathbf{a}_1^H(f)\,\mathbf{w}(n,f)|^2}{\mathbf{w}^H(n,f)\,\mathbf{R}_{\mathbf{v}}(n,f)\,\mathbf{w}(n,f)} \tag{2.65}$$

$$= \zeta(f)\,\mathbf{R}_{\mathbf{v}}^{-1}(n,f)\,\mathbf{a}_1(f) \tag{2.66}$$

where $\zeta(f)$ is an arbitrary scaling factor whose value can be set using, e.g., blind analytic normalization (BAN) (Warsitz and Haeb-Umbach, 2007). A closely related beamformer which is popular in the speech enhancement community is the generalized eigenvalue decomposition (GEV) beamformer (Warsitz and Haeb-Umbach, 2007) which is equal to the principal eigenvector of $\mathbf{R}_{\mathbf{v}}^{-1}(n,f)\,\mathbf{R}_{\mathbf{x}}(n,f)$.
A rank-1 constraint can be applied on the source covariance matrix of the classical MWF in Eq. (2.58) to obtain the so-called rank-1 MWF (R1-MWF) (Souden et al., 2010). The R1-MWF has been used for speech enhancement in cochlear devices (Serizel et al., 2014) as well as to obtain state-of-the art performance in multichannel ASR (Wang et al., 2018b).
In order to compute these beamformers, the speech and noise covariance matrices must be estimated from the data. This is typically achieved using the time-frequency mask $\widehat{\mathcal{M}}_1$ estimated for the target source. Time-varying covariance matrices can be estimated recursively as

$$\mathbf{R}_{\mathbf{c}_1}(n,f) = \alpha\,\mathbf{R}_{\mathbf{c}_1}(n-1,f) + (1-\alpha)\,\widehat{\mathcal{M}}_1(n,f)\,\mathbf{x}(n,f)\,\mathbf{x}^H(n,f) \tag{2.67}$$

$$\mathbf{R}_{\mathbf{v}}(n,f) = \alpha\,\mathbf{R}_{\mathbf{v}}(n-1,f) + (1-\alpha)(1-\widehat{\mathcal{M}}_1(n,f))\,\mathbf{x}(n,f)\,\mathbf{x}^H(n,f) \tag{2.68}$$

where α is a forgetting factor. Alternatively, time-invariant covariance matrices can be estimated by pooling across time as

$$\mathbf{R}_{\mathbf{c}_1}(f) = \sum_n \widehat{\mathcal{M}}_1(n,f)\,\mathbf{x}(n,f)\,\mathbf{x}^H(n,f). \tag{2.69}$$

$$\mathbf{R}_{\mathbf{v}}(f) = \sum_n (1-\widehat{\mathcal{M}}_1(n,f))\,\mathbf{x}(n,f)\,\mathbf{x}^H(n,f). \tag{2.70}$$

The terms $\widehat{\mathcal{M}}_1(n,f)$ and $(1-\widehat{\mathcal{M}}_1(n,f))$ in these expressions are sometimes squared, which can be interpreted as computing the empirical covariance matrices of the speech and noise signals $\widehat{\mathcal{M}}_1(n,f)\,\mathbf{x}(n,f)$ and $(1-\widehat{\mathcal{M}}_1(n,f))\,\mathbf{x}(n,f)$ estimated by masking.

Time-invariant beamformers typically result in lesser distortion of the target speech at the cost of lesser noise reduction. For this reason, they are often preferred while enhancing speech for ASR purposes.

Time-frequency mask estimation: Different methods have been proposed to estimate time-frequency masks in a multichannel context prior to the emergence of deep learning. Herbordt and Kellermann (2003) relied on the formulation of the MVDR beamformer as a generalized sidelobe canceler (GSC) (Griffiths and Jim, 1982). The GSC transforms the constrained optimization problem in Eq. (2.64) into an unconstrained formulation. It involves a fixed beamformer outputting a reference speech signal, an orthogonal matrix called a blocking matrix to create a noise reference, and a multichannel adaptive filter to eliminate the noise component from the speech reference. An SPP-based mask can be computed based on the speech reference and the noise reference of the GSC (Herbordt et al., 2003) which can then be used to compute the covariance matrices (Herbordt et al., 2005).
Taseska and Habets (2013) proposed an expectation-maximization (EM) framework, which estimates the posterior SPP $p(\mathcal{H}_1|\,\theta)$ in the E-step and computes the parameters of the DOA model $p(\theta\,|\mathcal{H}_1) = \sum_{j=1}^{J} \mathcal{N}(\theta; \mu_{\theta_j}, \Sigma_{\theta_j})$ in the M-step, where $\mathcal{H}_1$ is the hypothesis that source 1 is present. Taseska and Habets (2017) assume the availability of the approximate knowledge of the speaker DOA and use it to determine the time-frequency bins dominated by the speaker. The obtained DOA-based mask is then used to compute the statistics required to estimate an MVDR beamformer.
Clustering-based approaches can also be used to estimate IBMs or IRMs. Yılmaz and Rickard (2004) cluster time-frequency bins in a joint ITD-ILD space. The ITD information was obtained using the IPD between a pair of microphones. This is reliable only up to a certain cut-off frequency determined by the spacing between the microphone pairs. Spatial aliasing destroys the direct relationship above the cut-off frequency. Unaliased time-frequency bins clustered around a particular IPD belong to a single speaker thereby yielding an IBM for the speaker. The expectation-maximization source separation and localization (MESSL) algorithm was proposed by Mandel and Ellis (2007) to overcome the aliasing problem. MESSL models the likelihood of a time-frequency bin belonging to a particular source and having a particular delay as a GMM. The associated posterior probabilities are collected to create an IRM for the source.

Iterative estimation of spectral and spatial parameters: In the above beamformers, the spectral features of each source are smoothed out by recursive averaging or pooling over time and, in the case of recursive averaging, the estimates of spatial features can also be poor. A Gaussian modeling framework which separates the spatial and spectral parameters and estimates them in an EM framework was proposed by Vincent et al. (2011) and Duong et al. (2010). In this framework, the spatial images of the sources are assumed to be generated as

$$\mathbf{c}_j(n,f) \sim \mathcal{N}_{\mathrm{c}}(\mathbf{c}_j(n,f)|0, \mathbf{\Sigma}_{\mathbf{c}_j}(n,f)) \tag{2.71}$$

where $\mathcal{N}_c$ is a complex-valued multivariate Gaussian distribution. The multichannel covariance matrix $\mathbf{\Sigma}_{\mathbf{c}_j}(n,f)$ is split into two components as

$$\mathbf{\Sigma}_{\mathbf{c}_j}(n,f) = \sigma_j^2(n,f)\,\mathbf{R}_j(f) \tag{2.72}$$

where $\mathbf{R}_j(f)$ is the time-invariant spatial covariance matrix of the source and $\sigma_j^2(n,f)$ its time-varying power spectral density.
Recalling the mixing model

$$\mathbf{x}(n,f) = \sum_{j=1}^{J'} \mathbf{c}_j(n,f) + \mathbf{u}(n,f), \tag{2.73}$$

the log-likelihood of $\mathbf{x}$ is given by

$$\log p(\mathbf{x}\,|\sigma^2,\mathbf{R}) = -\sum_{n,f}\Big[\log\det(\pi\,\mathbf{\Sigma}_{\mathbf{x}}(n,f)) + \mathrm{tr}(\mathbf{\Sigma}_{\mathbf{x}}^{-1}(n,f)\,\mathbf{R}_{\mathbf{x}}(n,f))\Big] \tag{2.74}$$

where det(.) and tr(.) are the determinant and the trace of a matrix,

$$\mathbf{\Sigma}_{\mathbf{x}}(n,f) = \sum_{j=1}^{J'} \mathbf{\Sigma}_{\mathbf{c}_j}(n,f) + \mathbf{\Sigma}_{\mathbf{u}}(n,f) = \sum_{j=1}^{J'} \sigma_j^2(n,f)\,\mathbf{R}_j(f) + \mathbf{\Sigma}_{\mathbf{u}}(n,f) \tag{2.75}$$

is the multichannel covariance of the mixture, and $\mathbf{R}_{\mathbf{x}}(n,f) = \mathbf{x}(n,f)\,\mathbf{x}^H(n,f)$ is its empirical covariance.
Given a model to estimate the spectra of all sources and initial values of $\mathbf{R}_j$, the parameters $\mathbf{R}_j(f)$ and $\sigma_j^2(n,f)$ and the spatial images $\widehat{\mathbf{c}}_j(n,f)$ of all sources can be estimated via the following EM algorithm:

1. Compute the mixture covariance matrix $\mathbf{\Sigma}_{\mathbf{x}}$ using Eq. (2.75)
2. Expectation step: for each source,
   a) Compute the multichannel Wiener filter $\mathbf{W}_{\mathrm{MWF}_j}$ by

$$\mathbf{W}_{\mathrm{MWF}_j}(n,f) = \mathbf{\Sigma}_{\mathbf{x}}^{-1}(n,f)\,\mathbf{\Sigma}_{\mathbf{c}_j}(n,f) \tag{2.76}$$

   b) Estimate the spatial image of source j by

$$\widehat{\mathbf{c}}_j(n,f) = \mathbf{W}_{\mathrm{MWF}_j}^H(n,f)\,\mathbf{x}(n,f) \tag{2.77}$$

   c) Compute $\mathbf{R}_{\mathbf{c}_j}$ as

$$\mathbf{R}_{\mathbf{c}_j}(n,f) = \widehat{\mathbf{c}}_j(n,f)\widehat{\mathbf{c}}_j^H(n,f) + (\mathbf{I} - \mathbf{W}_{\mathrm{MWF}_j}^H(n,f))\,\mathbf{\Sigma}_{\mathbf{c}_j}(n,f) \tag{2.78}$$

   where $\mathbf{I}$ is the identity matrix
3. Maximization step: for each source,
   a) Update $\mathbf{R}_j$ as

$$\mathbf{R}_j(f) = \frac{1}{N}\sum_n \frac{1}{\sigma_j^2(n,f)}\,\mathbf{R}_{\mathbf{c}_j}(n,f) \tag{2.79}$$

b) Update σ_j^2 as

$$\sigma_j^2(n,f) = \frac{1}{I}\,\mathrm{tr}(\mathbf{R}_j^{-1}(f)\,\mathbf{R}_{\mathbf{c}_j}(n,f)) \tag{2.80}$$

4. Refine the estimation of $\sigma_j^2(n,f)$ using a constrained spectral model such as NMF or a DNN (Nugraha, 2017) and recompute $\mathbf{\Sigma}_{\mathbf{c}_j}$ as

$$\mathbf{\Sigma}_{\mathbf{c}_j}(n,f) = \sigma_j^2(n,f)\,\mathbf{R}_j(f) \tag{2.81}$$

This framework typically results in greater interference and noise reduction at the cost of a greater distortion compared to the above beamforming methods, due to the fact that the MWF $\mathbf{W}_{\mathrm{MWF}_j}(n,f)$ can vary widely from one frame to the next.

Applying constraints on the source covariance matrices: Whenever some prior knowledge is available regarding the room dimensions and the speaker distance from the microphone array, a soft constraint can be applied on the covariance matrix based on the theory of statistical room acoustics (Kuttruff, 2016). $\mathbf{R}_j(f)$ can be modeled using the inverse Wishart density (Maiwald and Kraus, 2000), a conjugate prior to the log-likelihood of Eq. (2.74) as (Duong et al., 2013)

$$\mathbf{R}_j(f) \sim \mathcal{IW}(\mathbf{\Psi}_j(f), m) \tag{2.82}$$

where

$$\mathcal{IW}(\mathbf{R}_j(f)|\mathbf{\Psi}_j(f), m) = \frac{|\mathbf{\Psi}_j(f)|^m\,|\mathbf{R}_j(f)|^{-(m+I)} e^{\mathrm{tr}(\mathbf{\Psi}_j(f)\,\mathbf{R}_j^{-1}(f))}}{\pi^{I(I-1)/2}\prod_{i=1}^{I}\Gamma(m-i+1)}. \tag{2.83}$$

$\mathbf{\Psi}_j(f)$ is a positive definite inverse scale matrix and m is called the number of degrees of freedom. Given this distribution, the mean of the spatial covariance matrix $\boldsymbol{\mu}_{\mathbf{R}_j}(f)$ over all possible source and microphone positions can be expressed as (Kuttruff, 2016; Gustafsson et al., 2003)

$$\boldsymbol{\mu}_{\mathbf{R}_j}(f) = \frac{\mathbf{\Psi}_j(f)}{m-I} = \mathbf{d}_j(f)\,\mathbf{d}_j^H(f) + \sigma_{\mathrm{rev}}^2\mathbf{\Omega}(f) \tag{2.84}$$

where σ_{rev}^2 is the power of echoes and reverberation in the room given by

$$\sigma_{\mathrm{rev}}^2 = \frac{4\alpha^2}{S(1-\alpha^2)} \tag{2.85}$$

$$\alpha = e^{-\frac{13.82}{(\frac{1}{L_x}+\frac{1}{L_y}+\frac{1}{L_z})c\,\mathrm{T}_{60}}} \tag{2.86}$$

with S the total surface of the room, α the reflection coefficient of the walls, and L_x, L_y, L_z the room dimensions. The $I \times I$ matrix $\mathbf{\Omega}(f)$ is the covariance matrix of a diffuse sound field whose entries

$$\Omega_{i\,i'}(f) = \frac{\sin(2\pi f\,\ell_{i\,i'}/c)}{2\pi f\,\ell_{i\,i'}/c} \tag{2.87}$$

depend on the distance between microphones $\ell_{i\,i'}$. Based on Eq. (2.84), the inverse scale matrix can be constrained as $\mathbf{\Psi}_j(f) = (m-I)\boldsymbol{\mu}_{\mathbf{R}_j}(f)$ where m is set experimentally. The source spatial covariance matrices can then be estimated in the maximum a posteriori sense using a modified version of Eq. (2.79).

2.3.4 DNN methods for multichannel speech separation

For a single speaker in the time-frequency domain: Following the early methods for mask estimation and beamforming presented above, several deep learning based methods have been proposed for multichannel speech enhancement of a single speaker in noise. One general approach is to obtain an initial clean speech estimate using a first beamformer. The estimated clean speech is used to compute a mask using a DNN which is then used to derive a more accurate beamformer.
For example, Menne et al. (2016) and Du et al. (2016) use a complex Gaussian mixture model (CGMM) (Higuchi et al., 2016) to estimate a first mask. An MVDR beamformer (Menne et al., 2016) or a GEV beamformer (Du et al., 2016) is obtained using this mask which is then used to obtain the first estimate of clean speech. Different DNN architectures such as Bi-LSTM or CNNs are used to obtain a time-frequency mask from that initial estimate. A GEV beamformer is then used to obtain the final estimate of the clean speech from the mixture. A data-independent beamformer based on TDOA estimation is used by Sivasankaran et al. (2015) to obtain the first estimate of clean speech. A Bi-LSTM estimates the mask which is used to derive an MWF. Heymann et al. (2015) ignore the initial estimation step: instead, they compute a mask for every channel using a DNN and average it across all channels. This is eventually used by a GEV beamformer with BAN to enhance speech.
Heymann et al. (2017) extend this approach by connecting an ASR acoustic model (AM) to the output of the GEV beamformer and training the system end-to-end. Improved enhancement results were obtained by incorporating the knowledge of the AM into the system.

For multiple speakers in the time-frequency domain: Fewer deep learning based methods have been proposed for the separation of multiple speakers. These include multichannel extensions of the deep clustering framework. Wang and Wang (2019) used a concatenation of the magnitude spectrum and CSIPD features to train the DNN for clustering. It indirectly uses the location information embedded in CSIPD features for speech separation. This method generalizes well to unseen speakers but clustering faces inherent limitations such as estimating the number of clusters (Higuchi et al., 2017) and choosing an appropriate clustering algorithm to optimally model the embedding space (Kinoshita et al., 2018).
In contrast, Perotin et al. (2018) and Chen et al. (2018a) propose a spatial location based approach, where the DOA of each speaker is assumed to be known and is used to beamform the multichannel signal in that direction. Features extracted from the beamformed signal are fed as inputs to a neural network that estimates a time-frequency mask corresponding to the speaker. The beamformed signal is dominated by the desired speaker which allows the network to estimate the relevant mask. The estimated masks are then used for speech separation using data-dependent beamformers.

From the raw waveform: Waveform-based deep learning methods for enhancement and separation have also been proposed. In one of the earlier works on using the multichannel

raw waveform to learn the front-end for ASR (Hoshen et al., 2015; Sainath et al., 2017), a set of one-dimensional CNNs were used to filter 275 ms of speech with ASR senones (see Section 2.3.5) as output and cross entropy as a cost function. It was shown that the filters were able to create nulls in the direction of the noise. Deepbeam (Qian et al., 2018) is an iterative approach where an estimate of the clean speech obtained from a Wavenet (van den Oord et al., 2016) like DNN is used to estimate a time-domain Wiener filter. A cleaner version of the signal after beamforming is then re-fed to the network to obtain better denoising. FaSNet (Luo et al., 2019) computes the cross-correlation of the channels with respect to a reference channel in the time domain and uses it to compute a first estimate of clean speech using a TasNet (Luo and Mesgarani, 2019) like DNN architecture. Another TasNet, connected in series, is used to compute a beamformer to estimate clean speech.
The above DNN-based methods for speech separation either assume the knowledge of the true position of the speaker, ignore spatial information entirely, or use it in an implicit fashion. By contrast, in this thesis we will estimate the speaker location using a DNN and we will use it explicitly to separate speech.

Datasets used for the evaluation of multichannel speech separation A reverberated version of WSJ0-2mix containing mixtures with 8 microphones without noise was proposed by Wang and Wang (2019). A reverberated dataset based on WSJ, containing larger variations in the mixtures compared to WSJ0-2mix was proposed in the SMS-WSJ dataset (Drude et al., 2019). Random microphone noise was included as part of the mixtures. None of these datasets contains realistic ambient noise which is known to impact the performance of speaker localization and speech separation systems. Simulating such noise is non-trivial and still an open research problem.

2.3.5 Automatic speech recognition

Since our goal is to improve ASR performance, the speech separation algorithms developed in this thesis are evaluated in terms of ASR metrics. For this reason, we also provide an overview of ASR systems below.

Feature extraction: The first component of an ASR system is feature extraction. Popular choices of features include Mel frequency cepstral coefficients (MFCCs) (Davis and Mermelstein, 1980), perceptual linear prediction coefficients (Hermansky, 1990) and power-normalized cepstral coefficients (Kim and Stern, 2016). MFCCs are chosen in the following. They are obtained by computing the short-term magnitude spectrum of the input signal $x_1(t)$ using a window size of 25 ms and frame shift of 10 ms. So-called logmel features are obtained by multiplying the short-term magnitude spectrum by a bank of 40 triangular filters uniformly spaced along the Mel scale (Rabiner and Juang, 1993) and computing the natural logarithm thereafter. MFCCs are obtained as the discrete cosine transform of the logmel features in each frame.

Overall formulation: Denoting by $\mathcal{O} = [\mathbf{o}_1, \ldots, \mathbf{o}_N]$ the MFCCs of the input signal, the problem of ASR can be defined as

$$\widehat{w} = \underset{w \in \mathcal{W}*}{\operatorname{argmax}}\, p(\mathcal{O}\,|w)p(w) \tag{2.88}$$

where $\widehat{w}$ is the estimated word sequence and $\mathcal{W}*$ is the set of all possible word sequences. The number of words in a language is very large and it is hard to build models for each word. Instead, each word w can be broken down into a sequence of subwords such as phonemes or syllables. For example the word "speech" has the following phonetic expansion: S P IY CH.
The number of such phonemes — referred to as monophones in ASR terminology — is much smaller, typically in the range of $[40 - 50]$ and it is therefore much easier to build models for them. To incorporate context information, a sequence of three phones referred to as a triphone or context-dependent phones is typically modeled instead of single phone using a hidden Markov model (HMM). Eq. (2.88) can then be decomposed as

$$\widehat{w} = \underset{w \in \mathcal{W}*}{\operatorname{argmax}} \sum_{q,w'} p(\mathcal{O}, q|w')p(w'|w)p(w) \tag{2.89}$$

$$\approx \underset{w \in \mathcal{W}*}{\operatorname{argmax}} \max_{q,w'} \underbrace{p(\mathcal{O}, \underbrace{q}_{\text{HMM}}\,|w')}_{\text{AM}} \underbrace{p(w'|w)}_{\text{Lexicon}} \underbrace{p(w)}_{\text{LM}} \tag{2.90}$$

where w' is the sequence of context-dependent phones and q the sequence of HMM states. The optimum in Eq. (2.90) is found by searching in a weighted directed graph, implemented using a weighted finite state transducer (WFST) (Mohri et al., 2002). The WFST is composed of HMM triphone models, a lexicon and a language model (LM) as shown in Eq. (2.90). We describe these components briefly below.

Acoustic modeling: Assuming that the number of phones is 40, there are 40^3 possible triphones, which would require us to estimate a large number of HMM parameters. To overcome this issue, a decision tree is employed to identify the HMM states whose parameters can be shared (Young et al., 1994). The tied HMM states are called senones. Each senone is modeled using a GMM and the resulting AM is referred to as a GMM-HMM. In the earlier part of this decade, DNNs were found to provide better modeling capabilities compared to GMMs (Hinton et al., 2012). So-called hybrid DNN models jointly estimate the posterior probabilities of all senones, and are now considered to be the state-of-art for speech recognition.

DNN architectures for acoustic modeling: Initial breakthroughs with hybrid models were obtained using simple MLP architectures (Hinton et al., 2012). DNN architectures have evolved over the years. A CNN based on the network-in-network architecture was proposed by Yoshioka et al. (2015) and was found to provide large ASR improvements in the context of the CHiME-3 challenge. A time-delayed neural network architecture

was proposed by Peddinti et al. (2015) to obtain state-of-art performance in the ASpIRE challenge (Harper, 2015). Other architectures based on residual networks (Resnet) and Bi-LSTMs are popular in the ASR community.

Cost functions to train an acoustic model: Since the AM aims to find the senone class in each time frame, frame-level cross-entropy is a natural choice of cost function. This has the drawback that the AM is trained to classify individual frames without looking at the context, while ASR is inherently a sequence classification problem. Sequence discriminative criteria such as maximum mutual information (MMI) (Bahl et al., 1986), boosted MMI and minimum phone error (MPE) (Povey, 2003) have been shown to outperform DNNs trained using cross-entropy (Veselý et al., 2013). The idea in a sequence discriminative approach such as MMI is to maximize the probability of the ground truth word sequence with respect to the probability of all possible word sequences. This is computationally intractable since the number of possible word sequences grows exponentially with the vocabulary size. To overcome this problem the sum is limited to the decoded speech lattice, thereby reducing the number of possible sequences. A lattice-free variant of sequence discriminative training is proposed by Povey et al. (2016) where phone sequences are used instead of word sequences, thereby reducing the computational complexity substantially. We use lattice-free MMI as the cost function to train AMs in this thesis.

Speaker information can greatly improve ASR performance. A commonly used speaker representation is the i-vector (Dehak et al., 2011). A GMM called the universal background model (UBM) is trained on a corpus of utterances from many speakers. Each Gaussian represents a pseudo-phonetic state, and the mean of that Gaussian represents the average MFCC vector of that phonetic state across all speakers. Given an utterance from a certain speaker, the average MFCC vectors are computed for all phonetic states and they are concatenated into a so-called supervector. This supervector is approximated as the sum of the UBM supervector and the product of a low-rank speaker variability matrix with a low-dimensional speaker vector called i-vector which encodes the specific pronunciation of that speaker. The i-vector is concatenated with the MFCCs as inputs to the AM. We use 100 dimensional i-vectors in this thesis.

Language modeling: The LM estimates the prior probability of a word sequence $w = w_1, \ldots, w_{N'}$ where N' is the number of words. It can be decomposed into products of conditional probabilities as

$$p(w_1, \ldots, w_{N'}) = \prod_{n'=1}^{N'} p(w_{n'} | w_{n'-N'+1}, \ldots, w_{n'-1}) \tag{2.91}$$

For large values of N', Eq. (2.91) becomes computationally intractable. In practice, N' is restricted to a smaller value in the range of $[1-4]$. The LM is then referred to as an

n-gram LM. The conditional probabilities are computed by counting all occurrences of the word sequence as

$$p(w_{n'}|w_{n'-N'+1},\ldots,w_{n'-1}) = \frac{\#(w_{n'-N'+1},\ldots,w_{n'})}{\#(w_{n'-N'+1},\ldots,w_{n'-1})} \tag{2.92}$$

where #() refers to the number of occurrences in the training set.

2.4 Model interpretation

Despite their impressive performance, DNNs are notoriously hard to interpret. Understanding their inner working is important since it enables users to trust their outputs and developers to come up with improved designs. With legal frameworks such as the general data protection regulation (GDPR)[3], consumers also have the right to ask for explanations of decisions made by machine learning models. Progresses along that line are collectively referred to as explainable artificial intelligence (XAI) (Barredo Arrieta et al., 2020; Molnar, 2019).

There are two ways in which DNN outputs can be interpreted:

1. post-hoc interpretability, where the goal is to explain the output of a trained model,
2. intrinsic interpretability, where the model structure lends itself to interpretation due to its simplicity — trees and linear models for example — or its architecture — involving attention mechanisms for example.

In this work we do not modify the DNN architecture and therefore focus on post-hoc interpretability methods.

Different components of the machine learning architecture can be analyzed to explain its outputs. For example, statistics of the input features such as pairwise feature interactions can help to explain why a model gave a certain output. Feature attribution based methods focus on assigning an importance value to each input feature, thereby informing the user as to what aspects of the data was relevant in making a decision. Analyzing model internals such as the weights of a trained model can be used to understand its inherent biases.

Our objective is to understand how speech enhancement models behave in the presence of different noises in the training stage, and how this affects the test performance. Specifically, we are interested in finding the input time-frequency bins which contribute most to the estimation of the output time-frequency mask, which can be seen as computing feature importance for every time-frequency bin. We therefore focus on feature attribution methods in this thesis. Feature attribution methods are either obtained using the gradient of the model outputs with respect to the inputs or by using the product of this gradient with the inputs. These approaches are detailed below.

2.4.1 Feature attribution using gradients

Simonyan et al. (2014) proposed to use the gradient of the model outputs $\mathcal{F}(\mathbf{x})$ with re-

[3] https://gdpr-info.eu/

spect to the inputs $\mathbf{x}$, to obtain a so-called "saliency map". The saliency map highlights the input features whose change in value has maximum impact on the model outputs. Smilkov et al. (2017) proposed a sharper saliency map called Smoothgrad by creating multiple noisy instances of the inputs by adding noise, and averaging the resulting saliency maps.
Deconvolution networks (Zeiler and Fergus, 2014) are similar to the saliency map approach except when backpropagating gradients through a ReLU nonlinearity where they backpropagate the gradient only if it is positive while ignoring the value of the signal coming into the ReLU. In contrast gradients in saliency maps are backpropagated only when the signal coming into ReLU is positive due to the nature of the ReLU nonlinearity. Guided backpropagation (Springenberg et al., 2015) combines both approaches and backpropagates the gradient only when both the signal and the gradient are positive.

2.4.2 Feature attribution using gradients and inputs

Using the inputs along with the gradients is better than using the gradients alone since the magnitude and the sign of the inputs can also be leveraged to obtain useful feature importance values. Common feature attribution methods based on this idea are detailed below.

2.4.2.1 Integrated gradients

The integrated gradients method (Sundararajan et al., 2017) varies the input signal from a reference value to its true value and compute the gradients at all points along the straight line connecting them. The path integral of the obtained gradients is referred to as the integrated gradient. Denoting as $\mathbf{x} = [x_1, \ldots, x_D]$ the input vector, with D the number of features, the integrated gradient for the d-th dimension is computed as

$$\mathrm{IG}(x_d) = (x_d - x_d^{\mathrm{r}}) \times \int_{\alpha=0}^{1} \frac{\partial(\mathcal{F}(\mathbf{x}^{\mathrm{r}} + \alpha(\mathbf{x} - \mathbf{x}^{\mathrm{r}}))}{\partial\, x_d} d\alpha \tag{2.93}$$

where $\mathbf{x}^{\mathrm{r}} = [x_1^{\mathrm{r}}, \ldots, x_D^{\mathrm{r}}]$ is the reference input vector which can be an all-zero vector for example.

2.4.2.2 Layerwise relevance propagation

Layerwise relevance propagation (LRP) (Bach et al., 2015) proposes to explain classification decisions by backpropagating the so-called relevance iteratively through a series of nonlinear layers, from one specific output down to all inputs. The relevance for the last layer of the network is the value for the considered output. The relevance values for the input layer form a heatmap which shows the importance of each input feature.
For a feedforward network, if the activation of a neuron b in a layer l_1 is denoted as ι_b, then the activation $\iota_{b'}$ in the subsequent layer l_2 is defined as

$$\iota_{b'} = f(\iota_b \vartheta_{b\,b'} + \kappa_{b'}) \tag{2.94}$$

where f is a nonlinear activation function such as a sigmoid or a ReLU, $\vartheta_{b\,b'}$ is the weight connecting the two neurons b and b' and κ is the bias.
If $\mathrm{R}_{l_1\,b}$ denotes the relevance for the b-th neuron in layer l_1 and $\mathrm{R}_{l_2\,b'}$ denotes the relevance for the b'-th neuron in layer l_2, the relevance in LRP is computed as

$$\mathrm{R}_{l_1\,b} = \sum_{b'} \left(\frac{\iota_b \vartheta_{b\,b'}}{\sum_b \iota_b \vartheta_{b\,b'}} \right) \mathrm{R}_{l_2\,b'} \,. \tag{2.95}$$

A central concept of LRP is the conservation property:

$$\sum_b \mathrm{R}_{l_1\,b \leftarrow l_2\,b'} = \mathrm{R}_{l_2\,b'} \tag{2.96}$$

and

$$\mathrm{R}_{l_1\,b} = \sum_{b'} \mathrm{R}_{l_1\,b \leftarrow l_2\,b'} \tag{2.97}$$

where $\mathrm{R}_{l_1\,b \leftarrow l_2\,b'}$ denotes the flow of relevance from the respective neurons of layer l_2 to l_1. This implies that the total relevance is fixed across all layers:

$$\sum_b \mathrm{R}_{l_1\,b} = \sum_{b'} \mathrm{R}_{l_2\,b'} = \sum_{b''} \mathrm{R}_{l_3\,b''} = \ldots = \mathcal{F}(\mathbf{x})[k], \tag{2.98}$$

for the considered output k. For the conservation property to hold, the relevance propagation must follow a set of rules. One such rule is the $\alpha\beta$ propagation rule which separates the positive and negative parts of the relevance as

$$\mathrm{R}_{l_1\,b} = \sum_{b'} \left(\alpha \frac{\iota_b \vartheta^+_{b\,b'}}{\sum_b \iota_b \vartheta^+_{b\,b'}} - \beta \frac{\iota_b \vartheta^-_{b\,b'}}{\sum_b \iota_b \vartheta^-_{b\,b'}} \right) \mathrm{R}_{l_2\,b'}, \tag{2.99}$$

where $\vartheta^+_{b\,b'}$ denotes positive weights and $\vartheta^-_{b\,b'}$ denotes negative weights. The values of α and β are chosen such that $\alpha - \beta = 1$ with the constraint that $\beta \geq 0$.
The relevance values obtained by LRP were shown to be equivalent to the product of the inputs and the gradient up to a scaling value by Shrikumar et al. (2017b).

2.4.2.3 DeepLIFT

The above techniques suffer from two problems:

1. Saturation problem: After a neuron has reached saturation, the output of the neuron remains unchanged even when the input changes. This results in zero-valued gradients thereby misrepresenting the importance of the input signal.
2. Thresholding problem: nonlinearities such as ReLU can cause sudden jumps in the gradients resulting in discontinuous jumps in the signal importance.

Deep Learning Important FeaTures (DeepLIFT) (Shrikumar et al., 2017a) addresses these issues. Instead of explaining which inputs have led to the observed output, DeepLIFT tries to explain the difference between the observed output and some reference output in

terms of the difference between the inputs and some reference inputs. DeepLIFT assigns a responsibility term $C_{\Delta x_d \Delta \mathcal{F}}$ to each input feature x_d explaining the difference $\Delta \mathcal{F}$ in the model output which respect to a reference output value. The relation between the responsibility term and the difference Δx_d in the input values is linear and defined as

$$m_{\Delta x_d \Delta \mathcal{F}} = \frac{C_{\Delta x_d \Delta \mathcal{F}}}{\Delta x_d} \tag{2.100}$$

with the constraint

$$\sum_{d=1}^{D} C_{\Delta x_d \Delta \mathcal{F}} = \Delta \mathcal{F} \tag{2.101}$$

where $m_{\Delta x_d \Delta \mathcal{F}}$ is referred to as a multiplier. The multiplier can be considered as the contribution of Δx_d to $\Delta \mathcal{F}$ divided by Δx_d. The concept of multiplier is similar to that of partial derivative $\frac{\partial \mathcal{F}}{\partial x}$.
The input difference-from-reference Δx_d with respect to a reference input x_d^{r} is defined as

$$\Delta x_d = x_d - x_d^{\mathrm{r}} \tag{2.102}$$

and the output difference-from-reference $\Delta \mathcal{F}$ with respect to a reference output $\mathcal{F}^{\mathrm{r}} = \mathcal{F}(\mathbf{x}^{\mathrm{r}})$ is defined as

$$\Delta \mathcal{F} = \mathcal{F}(\mathbf{x}) - \mathcal{F}^{\mathrm{r}}\,. \tag{2.103}$$

Denoting the neurons in the hidden layers as b, the multiplier values can be obtained using the chain rule as

$$m_{\Delta x_d \Delta \mathcal{F}} = \sum_b m_{\Delta x_d \Delta \iota_b} m_{\Delta \iota_b \Delta \mathcal{F}}. \tag{2.104}$$

Like in LRP, it is useful to separate the positive and negative contributions as

$$\Delta \iota_b = \Delta \iota_b^+ + \Delta \iota_b^-, \tag{2.105}$$

which implies

$$C_{\Delta \iota_b \Delta \mathcal{F}} = C_{\Delta \iota_b^+ \Delta \mathcal{F}} + C_{\Delta \iota_b^- \Delta \mathcal{F}}. \tag{2.106}$$

The following rules govern the assignment of contribution scores for each neuron b.

1. The first rule is the linear rule, designed for network components containing only linear functions wherein the contribution $C_{\Delta x \Delta b}$ coming from intermediate neurons b is split into four components $C_{\Delta x_d^+ \Delta \iota_b^+}$, $C_{\Delta x_d^- \Delta \iota_b^+}$, $C_{\Delta x_d^+ \Delta \iota_b^-}$, $C_{\Delta x_d^- \Delta \iota_b^-}$, where $(.)^+$ and $(.)^-$ mean that the contribution is accounted only when $(.) > 0$ and $(.) < 0$, respectively. The corresponding multipliers can be obtained using Eq. (2.100).
2. The rescale rule is designed to handle discontinuities in the gradient due to nonlinearities such as sigmoid, ReLU or hyperbolic tangent activation functions. In the rescale rule, the positive and negative components are obtained as

$$\Delta \iota_b^+ = \frac{\Delta \iota_b}{\Delta x_d} \Delta x_d^+ = C_{\Delta x_d^+ \Delta \iota_b^+} \tag{2.107}$$

$$\Delta \iota_b^- = \frac{\Delta \iota_b}{\Delta x_d} \Delta x_d^- = C_{\Delta x_d^- \Delta \iota_b^-}. \tag{2.108}$$

We therefore obtain

$$m_{\Delta x_d^+ \Delta \iota_b^+} = m_{\Delta x_d^- \Delta \iota_b^-} = \frac{\Delta \iota_b}{\Delta x_d}. \tag{2.109}$$

As observed by Shrikumar et al. (2017a), $m_{\Delta x_d \Delta \iota_b}$ has nonzero values even when the gradients are zero.

A comparison of above mentioned feature attribution methods on the MNIST dataset of handwritten digits for a DNN containing two-layers of CNN followed by two layers of fully connected dense layers is shown in Fig. 2.5. DeepLIFT and the integrated gradients method provide clearer and sharper attribution values compared to the other methods.

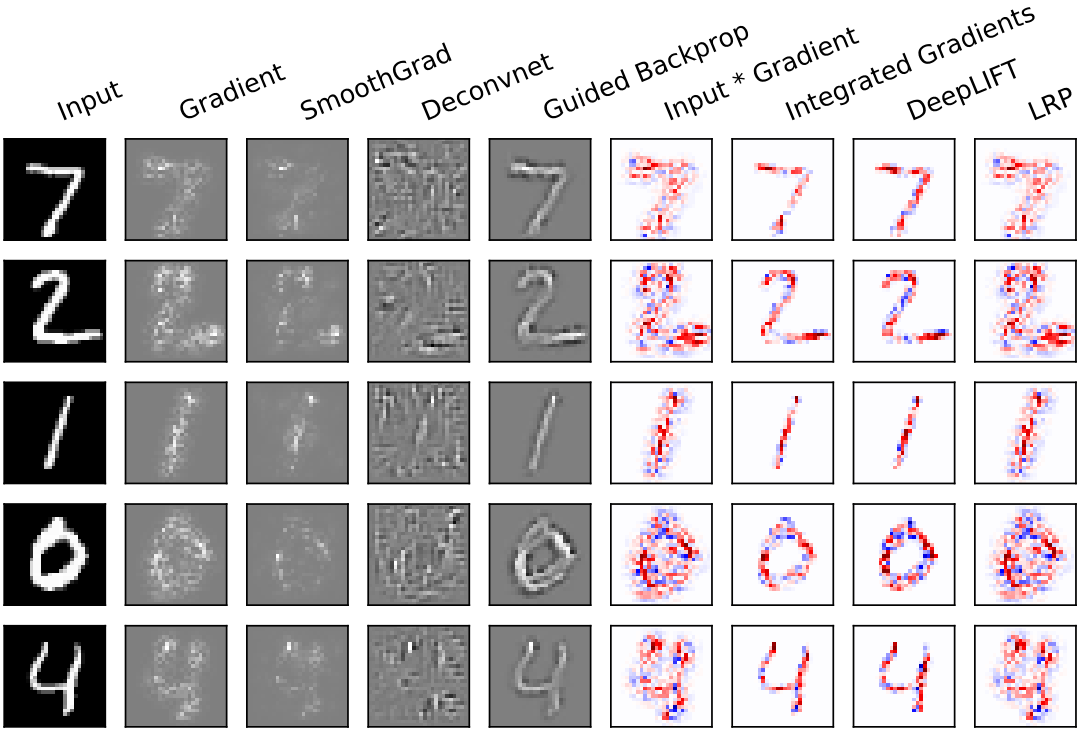


Figure 2.5: Comparison of different feature attribution method on the MNIST dataset. The iNNvestigate toolkit (Alber et al., 2019) was used to obtain the plot.

2.4.2.4 Shapley additive explanations

SHapley Additive exPlanations (SHAP) (Lundberg et al., 2020; Lundberg and Lee, 2017) uses Shapley values (Shapley, 1953) to explain model predictions. Shapley values were originally proposed in the field of game theory to distribute payout among the players in a game. SHAP tries to explain the model output by assuming that each input feature is a player in a game — the game being the computation of the model output. It was shown by Lundberg and Lee (2017) that all the above feature attribution methods are special cases of the SHAP framework.

Feature attribution based methods are designed with the idea that the absence of a particular feature in the inputs will impact the performance of the model. To find the feature attributions, the input features $\mathbf{x}$ are locally projected into a simplified input space $\mathbf{x}' = \{x'_1, \dots . x'_D\}$ of dimension D using an invertible function:

$$\mathbf{x} = h_{\mathbf{x}}(\mathbf{x}'). \tag{2.110}$$

Each simplified input $x'_d \in \{0, 1\}$ denotes the presence or absence of a certain feature. Different algorithms use different invertible functions. For example local interpretable model-agnostic explanations (LIME) (Ribeiro et al., 2016) converts an image into a set of superpixels. Each superpixel is assigned a presence (1) or absence (0) indicator value which acts as the simplified input x'_d. DeepLIFT converts the binary input x'_d into the original feature value when the indicator value is 1, otherwise a reference value is assigned. The simplified inputs are used to learn a linear model $\mathcal{G}(.)$ which takes $\mathbf{x}'$ as inputs and approximates the output of the original model $\mathcal{F}(.)$:

$$\mathcal{G}(\mathbf{x}') \approx \mathcal{F}(\mathbf{x}) = \mathcal{F}(h_{\mathbf{x}}(\mathbf{x}')) \tag{2.111}$$

where

$$\mathcal{G}(\mathbf{x}') = \phi_0 + \sum_{d=1}^{D} \phi_d \, x'_d \, . \tag{2.112}$$

A unique $\mathcal{G}$ can be obtained if ϕ has the following properties (Lundberg and Lee, 2017):

1. *Local accuracy:* the explanation model $\mathcal{G}$ must perfectly match the output of the original model, i.e.,
$$\mathcal{G}(\mathbf{x}') = \mathcal{F}(\mathbf{x}); \tag{2.113}$$
2. *Missingness:* If a feature x_d is absent, the explanation for that feature index must be zero.
$$\text{if } x'_d = 0 \quad \text{then} \quad \phi_d = 0; \tag{2.114}$$
3. *Consistency:* For two models $\mathcal{F}$ and $\mathcal{F}'$,
$$\text{if} \quad \mathcal{F}'(\mathbf{x}) - \mathcal{F}'(\mathbf{x} \setminus d) \geq \mathcal{F}(\mathbf{x}) - \mathcal{F}(\mathbf{x} \setminus d) \quad \text{then} \quad \phi_d(\mathcal{F}', \mathbf{x}) \geq \phi_d(\mathcal{F}, \mathbf{x}) \tag{2.115}$$
where
$$\mathbf{x} \setminus d = h_{\mathbf{x}}([x'_1, \dots, x'_{d-1}, 0, x'_{d+1}, \dots, x'_D]^T). \tag{2.116}$$

The unique $\mathcal{G}$ is obtained using :

$$\phi_d(\mathcal{F}, \mathbf{x}) = \sum_{\mathbf{z}' \subset \mathbf{x}'} \frac{|\mathbf{z}'|!(D - |\mathbf{z}'| - 1)!}{D!} [\mathcal{F}(h_{\mathbf{x}}(\mathbf{z}')) - \mathcal{F}(h_{\mathbf{x}}(\mathbf{z}' \setminus d))] \tag{2.117}$$

where $\mathbf{z}' \subset \mathbf{x}'$ represents all $\mathbf{z}'$ whose nonzero entries are a subset of the nonzero entries of $\mathbf{x}'$, and $|\mathbf{z}'|$ denotes the number of nonzero elements in $\mathbf{z}'$. ϕ_d measures the importance of the d-th feature and is called the Shapley value (Shapley, 1953) for that feature. DeepLIFT was shown to follow the SHAP formulation with $\phi_d = C_{\Delta x_d \Delta \mathcal{F}}$ by Lundberg and Lee (2017).

2.4.3 Application of feature attribution methods to speech

Many of the above feature attribution methods were designed for image or text classification problems. Few prior works can be found in the field of speech and audio, notably by Perotin et al. (2019b), Muckenhirn et al. (2019), Bharadhwaj (2018) and Becker et al. (2018).
LRP was applied to explain models across multiple tasks. Perotin et al. (2019b) used it in the context of single and multiple speaker localization to find the time-frequency bins responsible for the estimated DOAs. LRP was again used by Bharadhwaj (2018) to explain deep learning based ASR models and by Becker et al. (2018) to explain DNN-based models used for gender classification in audio. Guided backpropagation was used by Muckenhirn et al. (2019) to explain CNN-based phone recognition models.

3 Text-informed speaker localization

In this chapter, the new task of incorporating the text transcript of the speech uttered by a target speaker in order to improve speaker localization performance in adverse acoustic conditions is introduced. This kind of additional text information can be obtained in systems such as Google Assistant and Amazon Alexa which use wake-up words "OK Google" and "Alexa" respectively to activate automatic speech recognition (ASR). Here, we consider the situation where speech is recorded by a two-microphone array. In that situation, estimating the DOA of a speaker is equivalent to estimating the TDOA between the microphones. In this work, CSIPD features are used as input features. We found them to be particularly useful to incorporate the textual information into the localization system.

Multiple works have addressed the problem of multiple speaker localization using both signal processing based and learning based methods as detailed in Section 2.2. Nevertheless the task of localizing a particular target speaker in a multi-speaker environment has not been addressed before. This task raises two main challenges. The first challenge relates to the representation of the additional information needed to identify the target speaker and the second one concerns the incorporation of that target identifier into the localization system. In this work, the phonetic information extracted from the text spoken by the target speaker is used as the additional information to estimate the DOA of the target in the presence of an interfering speaker.

Considering that most time-frequency bins are dominated by a single source (Rickard and Yılmaz, 2002), a good target identifier should identify the time-frequency bins which are dominated by the target speaker. A time-frequency mask which represents the proportion of signal magnitude associated with the target in each time-frequency bin is therefore an ideal target identifier. Similarly the magnitude spectrum corresponding to the target signal provides information to identify the time-frequency bins related to the target speaker. We therefore use the masks or spectra corresponding to the target signal as identifiers to localize the target speaker. Estimating these masks or spectra from the observed mixture is non-trivial. To do so, the text information is first aligned with the signal. The corresponding sequence of mean phone spectra (Erdogan et al., 2015) and the spectrum of the mixture are appended as inputs to a DNN, which is trained to estimate the target identifier. Another DNN which uses the CSIPD features and the estimated target identifier is then trained to estimate the DOA.

Localization works best with long time frames in the order of 100 ms, which may be composed of several phones with different durations whereas as a typical ASR system assigns phonetic labels for shorter frames in the order of 25 ms (Povey et al., 2011). This introduces the additional issue of finding the right phonetic representation of the target speech segment in order to estimate the target identifier. One way to tackle this

problem is to select speech segments of at least 100 ms corresponding to a single phone while ignoring the remaining part of the segment. The selected segments represent a single phone, therefore the corresponding mean phone spectra can be used to estimate the target identifier. We refer to this method as *frame localization*. The disadvantage of the *frame localization* method is that we throw away chunks of speech which are useful for DOA estimation. This limitation can be overcome by exploiting all time frames and associating each frame with a sequence of several mean phone spectra. This method is referred to as *sequence localization*.

The following terminology is frequently used in the rest of the chapter:

- A *segment* refers to 0.5 s of speech corresponding to the wake-up word used for localization. The wake-up word duration is assumed to be 0.5 s at least and longer segments are trimmed to 0.5 s for DNN architecture simplicity.
- A *long frame* is a smaller part of a segment with 100 ms duration.
- A *short frame* is a smaller part of a long frame with 25 ms duration.

The rest of the chapter is organized as follows. The problem setup and an overview of the solution are discussed in Section 3.1. The estimation of the target identifier is presented in Section 3.2. DOA estimation using the target identifier is explained in Section 3.3. Section 3.4 describes the datasets used in this work and Section 3.5 details the experimental setup. The results are discussed and analyzed in Section 3.6. Section 3.7 concludes the chapter.

3.1 Problem setup

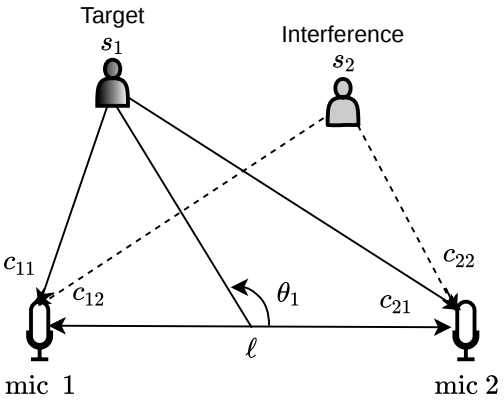


Figure 3.1: Problem setup.

The problem setup is shown in Fig. 3.1. A pair of microphones with distance ℓ from each other is placed at a random position and orientation in a room. A target speaker s_1 and an interfering speaker s_2 speaking simultaneously are recorded at both microphones. The goal is to estimate the DOA of the target speaker using the signals $\mathbf{x}$ captured at the microphones as well as the text spoken by the target speaker.

The signal captured at the i-th microphone is

$$x_i(n,f) = c_{i\,1}(n,f) + c_{i\,2}(n,f) + u(n,f).$$

The DOA θ_1 of the target speaker is to be estimated using $\mathbf{x}$ and the text spoken by that speaker.
As described in Chapter 2, the spatial image of a source with respect to a microphone, c_{ij}, contains the direct component, early echoes and late reverberation. Three different spatial components of a source with respect to a microphone can be envisaged, namely:

1. The direct-path component $c_{ij}^D(t)$ defined by

$$c_{ij}^D(t) = \sum_{\tau=0}^{\tau_D} a_{ij}(\tau)\, s_j(t-\tau) \tag{3.1}$$

   where a_{ij} is the RIR between microphone i and source j and τ_D is the time taken by the signal to reach the microphone. In practice, when the delay of arrival is fractional, the corresponding RIR samples are nonzero for several taps, and all those taps are included in τ_D.
2. The early component $c_{ij}^E(t)$ defined by

$$c_{ij}^E(t) = \sum_{\tau=0}^{\tau_E} a_{ij}(\tau)\, s_j(t-\tau) \tag{3.2}$$

   where τ_E is the time corresponding to the early echoes which is set to 50 ms after the arrival of the direct component.
3. The reverberated component or the full spatial image $c_{ij}^R(t)$ defined by

$$c_{ij}^R(t) = \sum_{\tau=0}^{\infty} a_{ij}(\tau)\, s_j(t-\tau). \tag{3.3}$$

Since we assume that the speaker is in the far field and the distance between the microphones is small ($\ell = 10$ cm), a representative magnitude spectrum $|\, c_j^D(n,f)|, |\, c_j^E(n,f)|$ or $|\, c_j^R(n,f)|$ for the j-th source can be computed by averaging the spectrum across the microphones. For example:

$$|\, c_j^E(n,f)| = \frac{1}{I} \sum_i |\, c_{ij}^E(n,f)|. \tag{3.4}$$

The overview of the proposed solution is illustrated in Fig. 3.2. The input CSIPD features are computed from the STFT of the captured multichannel signal. The CSIPD features along with the target identifier are used to train a neural network to estimate the DOA. The target identifier is in turn estimated with a convolutional neural network using the speech magnitude spectrum and mean phone spectrum corresponding to the text uttered by the target. The phonetic alignments needed to pick the mean phone spectrum are obtained using an automatic speech recognition (ASR) system. These blocks are detailed in the following sections.

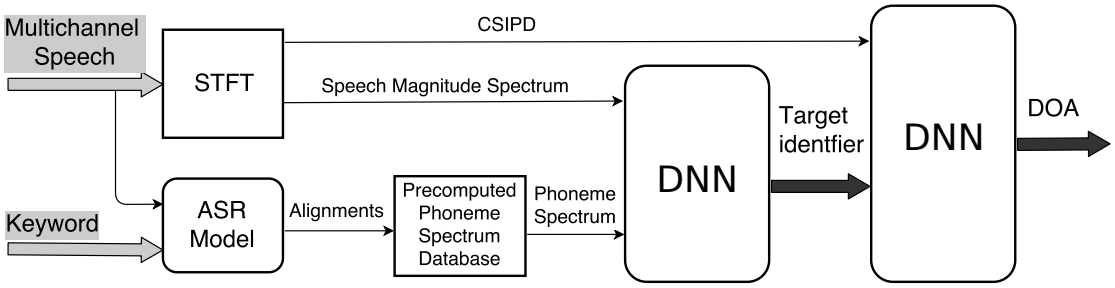


Figure 3.2: Overall structure of the proposed approach.

3.2 Target identifiers

3.2.1 Difference between *frame localization* and *sequence localization*

The computation of target identifiers differs for the *frame localization* and *sequence localization* systems. In the *frame localization* system, each speech segment containing a single phone can be used to estimate the target identifier. In the case of the *sequence localization* system, incorporating the phonetic information is not straightforward.

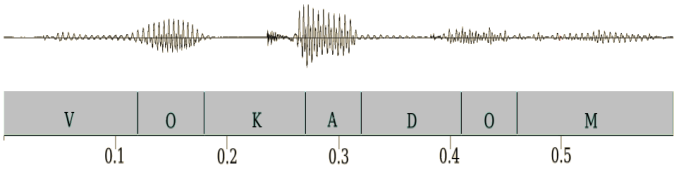


Figure 3.3: Phonetic boundaries of the word "Vocadom".

Figure 3.3 shows the waveform of an utterance of the word "Vocadom" along with the phonetic alignment obtained from an ASR system. There are 7 phones, out of which *V, K, D, M* are of at least 100 ms duration while the phones *O* and *A* are of shorter duration. If *frame localization* is used to estimate the DOA with 100 ms frames, only the phones *V, K, D, M* can be used while the speech signals corresponding to the phones *O* and *A* are ignored.
In *sequence localization* the whole signal is utilized by windowing it into 100 ms frames with 50 ms shift. A single phonetic representation cannot be obtained for each of the 100 ms frames. To handle this issue, we frame each 100 ms frame into shorter frames of 25 ms length with 10 ms shift. The ASR system assigns a phone to each short frame and the spectrum corresponding to the phone is picked from the phone spectrum database. This sequence of phone spectra represents the textual information spoken by the target which is then used to estimate the target identifier.

3.2.2 Computing the phone spectrum database

The database of phone spectra contains a representative spectrum for every phone. To compute these representative spectra, the magnitude spectra of all speech segments corresponding to a given phone in the training set are averaged in a similar fashion as by Erdogan et al. (2015) and Chen et al. (2015). Example representative spectra are shown in Fig. 3.4. Distinct patterns can be observed for each phone which provide clues to the DNN to identify the time-frequency bins dominated by the target speaker.

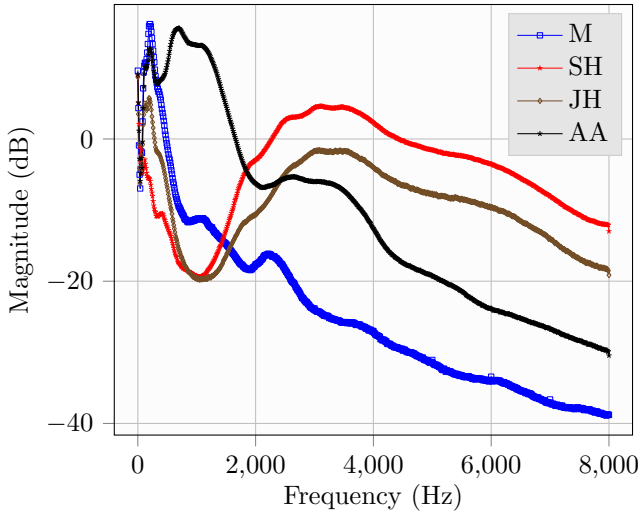


Figure 3.4: Mean phone spectra for the phones *M*, *SH*, *JH* and *AA* (in international phonetic alphabet notation: m, ʃ, ʤ, ɑ).

Two different phone spectrum databases were created. One database was created for the *frame localization* system where the phone spectra were computed from single-phone segments of 100 ms or more. Segments shorter than 100 ms were ignored and segments longer than 100 ms were trimmed to 100 ms before computing the spectrum. The other database was created for *sequence localization* where the spectra are computed in a similar fashion, but using 25 ms speech segments.

3.2.3 Types of target identifiers

We propose various target identifiers to identify the time-frequency bins which are useful to localize the target speaker. Such identifiers can be categorized into spectrum-based identifiers and mask-based identifiers. Spectrum-based identifiers directly relate to the spectrum of the target whereas mask-based identifiers measure the magnitude ratio associated with the target signal in each time-frequency bin. We explain the extraction of each of these identifiers in the ideal case when we assume the availability of the direct target signal as well as in real scenarios where the identifiers are to be estimated from the corrupted signals obtained at the microphones, $\mathbf{x}$.

3.2.3.1 Spectrum-based target identifiers

We define three possible spectrum identifiers corresponding to various amounts of reverberation:

1. *Direct spectrum identifier:* The direct component of the target signal $|\, c_j^D \,|$ is free from early echoes and late reverberation and enables highly accurate localization. We refer to $|\, c_j^D \,|$ as the *direct spectrum identifier*.
2. *Early spectrum identifier:* Blindly dereverberating the signal in order to estimate the direct spectrum is difficult. We therefore use the spectrum $|\, c_j^E \,|$ corresponding to the early component of the target signal as a target identifier. This identifier is presumably easier to estimate while still leading to good localization performance.
3. *Reverberated spectrum identifier:* The reverberated component of the target signal, $c_j^R(n, f)$ is corrupted by both early echoes and late reverberation but free from the interfering speech and noise. We hypothesize that it may be easier to estimate the reverberated spectrum than the direct or early spectrum of the target signal.

3.2.3.2 Mask-based target identifiers

Mask-based target identifiers represent the amount of target signal present in each time-frequency bin. Similar to spectrum-based identifiers, we can estimate the amount of direct, early and reverberated target speech in each time-frequency bin. To compute a mask for a particular target component, that component is first removed from the corrupted speech to obtain the remainder. The ratio of the component spectrum to the total spectrum is referred to as the mask identifier. The mask is therefore real-valued and lies in the range of $[0, 1]$.

Direct mask identifier: The direct mask identifier is obtained using the direct component of the target signal as

$$\mathcal{M}_1^D(n, f) = \frac{|\, c_1^D(n, f)|}{|\, c_1^D(n, f)| + |\delta_1^D(n, f)|} \tag{3.5}$$

where

$$|\delta_1^D(n, f)| = \frac{1}{I} \sum_i |\delta_{i1}^D(n, f)| \tag{3.6}$$

$$\delta_{i1}^D(n, f) = x_i(n, f) - c_{i1}^D(n, f). \tag{3.7}$$

Here δ_1^D represents the remainder of the signal obtained after removing the direct component of the target speech from the mixture signal.
It is important to compute the denominator as the sum of the component spectrum and the remainder spectrum instead of using the mixture spectrum $|\, \mathbf{x} \,|$ as shown in Eq. (3.5). This is because the mixture spectrum is influenced by destructive interference patterns arising due to reverberation and could result in mask values greater than 1.

Early mask identifier: This identifier is obtained using the early component of the target signal and is computed by

$$\mathcal{M}_1^E(n,f) = \frac{|\, c_1^E(n,f)|}{|\, c_1^E(n,f)| + |\delta_1^E(n,f)|} \tag{3.8}$$

where

$$|\delta_1^E(n,f)| = \frac{1}{I}\sum_i |\delta_{i1}^E(n,f)| \tag{3.9}$$

$$\delta_{i1}^E(n,f) = x_i(n,f) - c_{i1}^E(n,f). \tag{3.10}$$

Reverberated mask identifier: Similarly, the reverberated mask is computed by

$$\mathcal{M}_1^R(n,f) = \frac{|\, c_1^R(n,f)|}{|\, c_1^R(n,f)| + |\delta_1^R(n,f)|} \tag{3.11}$$

where

$$|\delta_1^R(n,f)| = \frac{1}{I}\sum_i |\delta_{i1}^R(n,f)| \tag{3.12}$$

$$\delta_{i1}^R(n,f) = x_i(n,f) - c_{i1}^R(n,f). \tag{3.13}$$

Figure 3.5 compares the ground truth direct, early and reverberated mask identifiers for the speech utterance containing the word "sure" spoken by the target and the word "ring" spoken by the interfering speaker. Not surprisingly, the direct component of the target signal dominates in fewer time-frequency bins and the reverberated component dominates in more bins.

3.2.4 Estimating the target identifiers

For *sequence localization*: The input features used to estimate the target identifier for *sequence localization* are computed as follows. The 0.5 s speech segment is padded on both sides with its own reflection before windowing it into 11 long frames of 100 ms each with a shift of 50 ms. Each long frame is further windowed into 8 short frames of 25 ms (400 samples) each with a shift of 10 ms. For each short frame, the STFT magnitude spectra for both microphones are appended with the corresponding mean phone spectrum and provided as inputs to a CRNN as shown in Fig. 3.6. The dimension of the input features is therefore $3 \times 8 \times 201 \times 11$ where 201 is the dimension of each short frame spectrum.

A three-dimensional CRNN is used to learn correlations along the long frame, short frame and frequency axes. Four convolutional layers are used with filters of shape $3 \times 3 \times 3$ resulting in $64, 32, 16$ and 4 filter maps, respectively. Batch normalization (Ioffe and Szegedy, 2015) and dropout (Srivastava et al., 2014) are used for all the convolutional layers. Max pooling is used in the second, third and fourth layers along the short frame

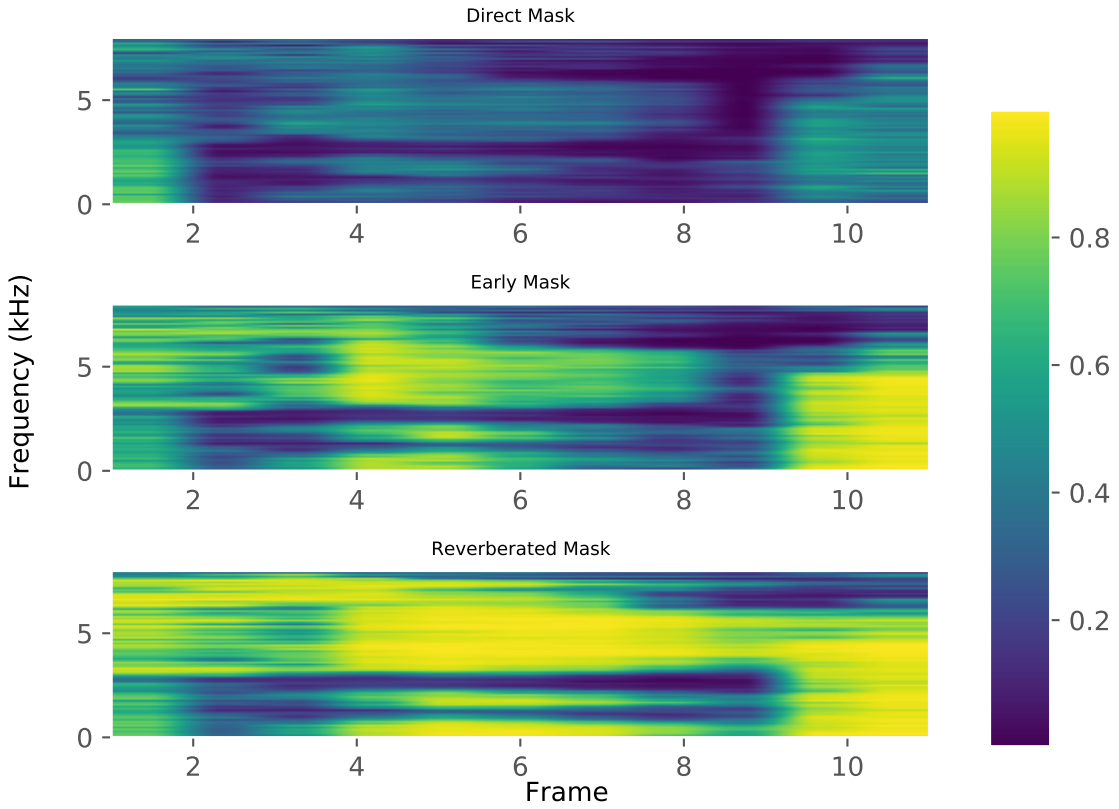


Figure 3.5: Example direct, early and reverberated target mask identifiers. The target speaker uttered the word *sure* and the interfering speaker uttered *ring*.

axis while retaining the frequency and the long frame axes. A bidirectional gated recurrent (GRU) layer (Cho et al., 2014a) is then used along the long frame axis followed by a time-distributed dense output layer. Rectified Linear Units (ReLU) are used as nonlinearities in all the convolutional layers. A sigmoid nonlinearity is applied at the output to learn mask-based target identifiers while a linear output layer is used to learn spectrum-based target identifiers. The network is trained with the Adam (Kingma and Ba, 2015) optimizer and mean squared error (MSE) is used as the cost function.
Figure 3.7 shows the inputs and outputs of the target identifier estimation network. The inputs are the magnitude spectra, shown in Fig. 3.7a for the signal captured at one of the microphones, and the mean phone spectra shown in Fig. 3.7b. The estimated outputs using the textual information (via phone spectra) are in Fig. 3.7e. The ground truth early target mask identifier and the mask corresponding to the interfering speaker are shown in Fig. 3.7c and 3.7d for comparison. As can be observed, the CRNN has been trained to use the text information, via the mean phone spectra, to estimate the relevant target time-frequency bins as highlighted in the white colored boxes in Fig. 3.7 and reject time-frequency bins dominated by the interfering speaker as highlighted in the red colored boxes.

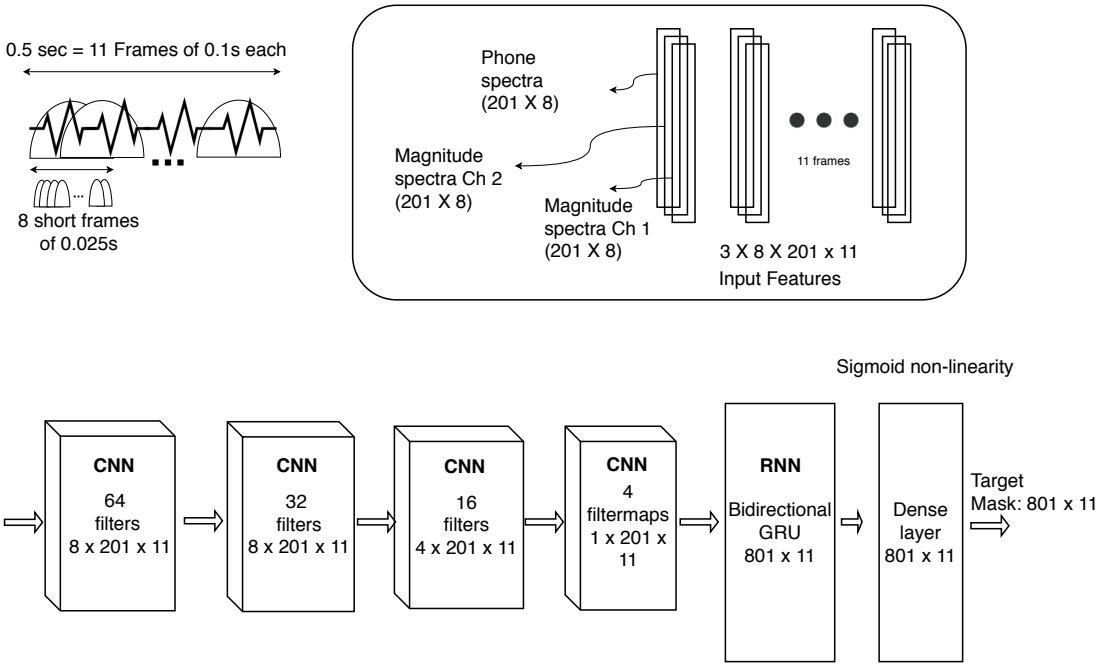
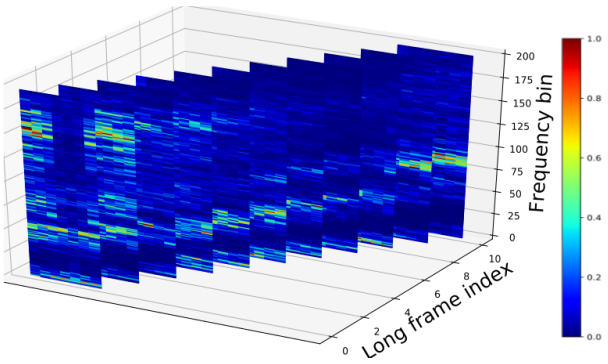


Figure 3.6: Network architecture for the estimation of the target mask identifier. To estimate spectrum identifiers, the last sigmoid layer is replaced by a linear layer. The upper part of the figure shows the construction of the input features and the lower part shows the CRNN architecture.

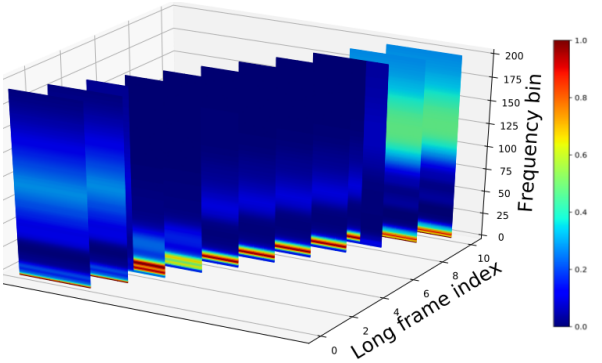
For *frame localization* : Target identifiers for *frame localization* are estimated in a similar fashion as *sequence localization* with small but important differences. In *frame localization* , DOA decisions are made for 100 ms long frames which have a single phonetic representation. The mean phone spectrum corresponding to each frame is appended with the magnitude spectrum of the signal captured at the two-microphones to estimate the identifier. The input features are therefore of dimension 3×801. The CNN architecture to estimate the identifier is similar to *sequence localization* as shown in Fig. 3.6 without the GRU layer, since there is no sequence information to be modeled.

3.2.5 Appending vs. multiplication of target identifiers

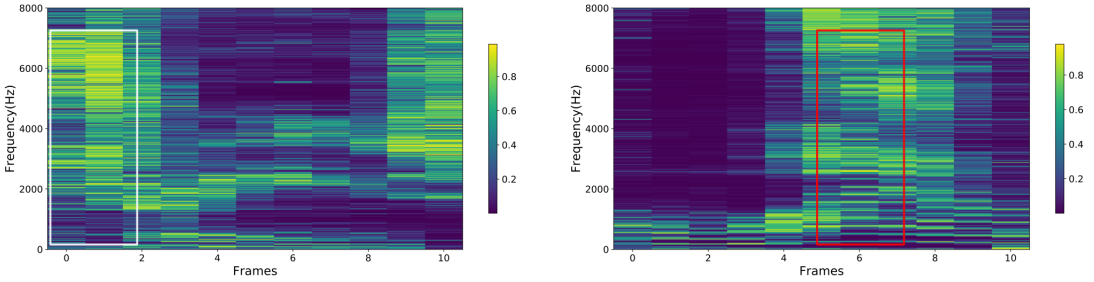
The target identifiers are appended to the CSIPD features and used as inputs to the CRNN or the CNN, both for training and testing. The network will learn to correlate the target identifier time-frequency bins with the CSIPD features while estimating the DOA. In the particular case of the target mask identifiers, this relationship can be directly imposed on the features by multiplying the CSIPD features with the mask in every time-frequency bin, thereby freeing the network from explicitly learning such a relationship. This would not be possible with other features such as phasemaps (Chakrabarty and Habets, 2017a) since multiplying the phase will disturb the phasemap patterns with-



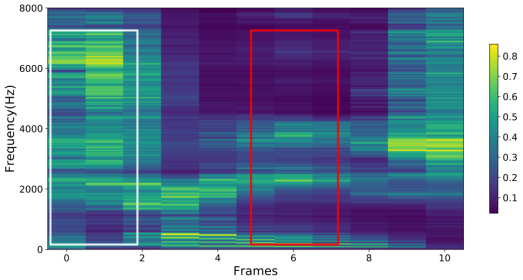
(a) Magnitude spectra.



(b) Sequence of mean phone spectra.



(c) Ground truth early mask of the target speaker. (d) Ground truth mask of the interfering speaker.



(e) Predicted early mask of the target speaker.

Figure 3.7: Target early mask estimation using the magnitude spectra and the mean phone spectra. Red and white rectangles in 3.7c, 3.7d and 3.7e correspond to the regions dominated by the interfering and target speakers, respectively.

out highlighting the target-dominated bins which are necessary to localize the target. Moreover multiplying CSIPD features with a mask is akin to weighting GCC which is a standard methodology used to achieve noise robustness for localization (Rascon and Meza, 2017).

3.3 DOA estimation

The CRNN architecture used to estimate the DOA in the case of *sequence localization* is shown in Fig. 3.8. The CSIPD features and the target identifiers are appended (or multiplied in the case of target mask identifiers) to form three-dimensional input features of shape $3 \times 801 \times 11$ (or $2 \times 801 \times 11$ if the mask identifier is multiplied). These input features are fed to a CRNN containing four convolutional layers with $64, 32, 16$ and 3 feature maps respectively. Batch normalization and Dropout are used for all layers. Max pooling is done for all layers except the first convolutional layer. The output of the last convolutional layer is fed to a bidirectional GRU layer. The outputs of the two directions are concatenated and fed to a time-distributed dense layer containing 512 units which is then fed to another time-distributed dense layer containing 181 units with softmax at the output. The ReLU nonlinearity is used in all the convolutional and dense layers.
For *frame localization* , the architecture is similar to Fig. 3.8 without the bidirectional GRU layer. The time-distributed dense layer is further replaced by a dense layer. Cross-entropy was used as cost function and Adam was used as the optimizer to train both DOA estimation networks.

3.4 Datasets

We use three different datasets to evaluate the performance of our system:

1. a simulated dataset consisting of artificially generated two-speaker mixtures,
2. the ANR Vocadom dataset consisting of real data recorded as part of the ANR Vocadom project,
3. the Inria two-speaker mixture dataset consisting of real data recorded specifically for this thesis at Inria Nancy – Grand Est.

Each of these datasets is described in detail below.

3.4.1 Simulated dataset

RIR simulation: Two microphones are used in all our experiments implying $I = 2$. The DOA space is quantized into 181 classes corresponding to the angles 0° to 180° with 1° step.
A method to generate the target and interference positions is shown in Fig. 3.9. For every configuration, a room with random dimensions in the range of $[3 - 8]$ m is chosen along with a random reverberation time in the range of $[0.3, 1]$ s. The first microphone is placed at a random point inside this room. The position of the second microphone is decided by selecting a random point on top of the sphere created using the first microphone as

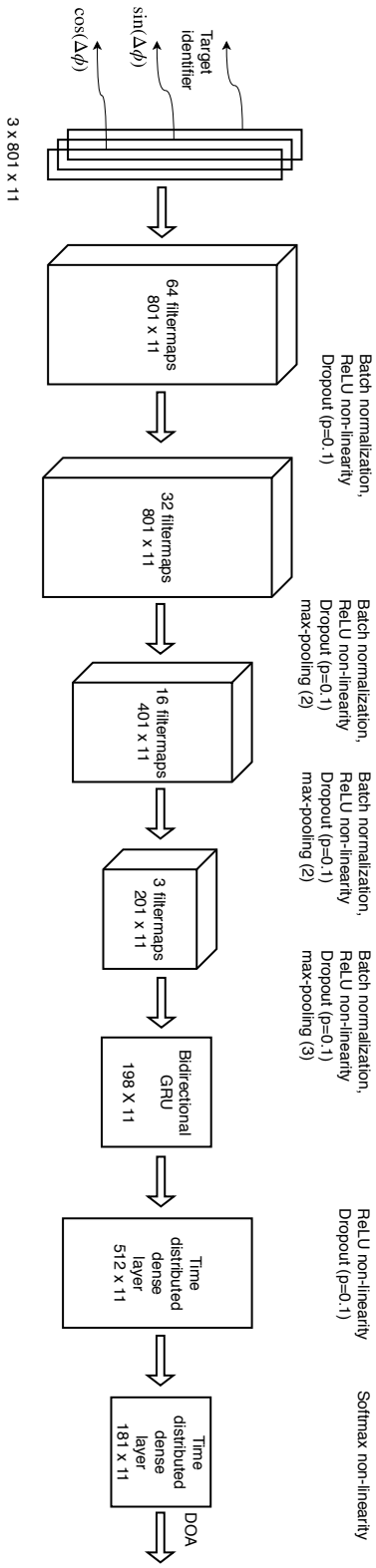


Figure 3.8: DOA estimation using *sequence localization*.

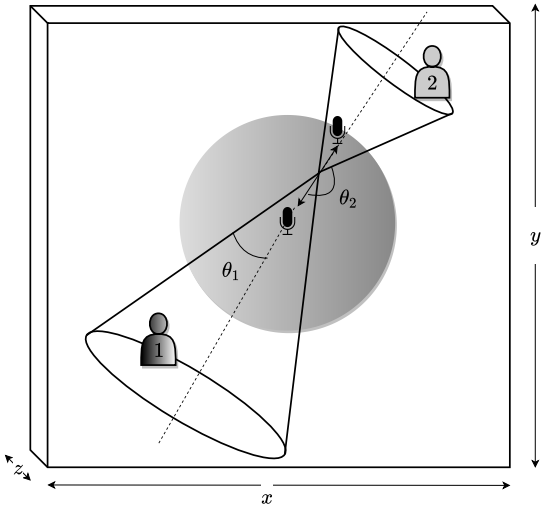


Figure 3.9: Generating random positions for microphones and speakers for RIR simulation.

the center with a radius of 10 cm. To decide the position of the target speaker, we pick a random point on a circular cone whose center lies on the the midpoint of the line joining the two microphones. The line joining the microphones is the cone axis and θ_1 is the cone angle. Similarly, the position of the interfering speaker is decided by choosing a random point on a circular cone of angle θ_2. Both speakers are at a distance of $[0.5 - 5.5]$ m from the microphone array. All microphones and speakers are at least 50 cm from the walls of the room.
All possible target DOA and interference DOA pairs $\{\theta_1, \theta_2\}$, $\forall\theta_1 \in [0, 180]$, $\forall\theta_2 \in [0, 180]$ with the constraint that $|\theta_1 - \theta_2| > 5°$ are created. 50, 1 and 2 such positions are created for every θ_1, θ_2 as part of the training, validation and test datasets. This results in 1,557,600, 31,152, and 62,304 configurations, respectively. To maximize the variability of the data, the room dimensions and the reverberation time are changed for every configuration. RIR-simulator (Habets, 2018) was used to simulate the RIRs using the microphone and speaker positions.

Generating speech mixtures: Speech signals from the Librispeech dataset (Panayotov et al., 2015) are used to create noisy speech. The dataset is divided into training, validation and test sets with no overlap. Two signals are randomly picked and convolved with the target and inference RIRs from a single room. The signal-to-interference-ratio (SIR) is randomly chosen in the range of $[0, 10]$ dB. We further add speech-shaped noise (SSN) (Pariente and Pressnitzer, 2017; Li et al., 2014; Valentini-Botinhao et al., 2016) to the mixtures in the training and validation sets with a signal-to-noise-ratio (SNR) in the range of $[0, 15]$ dB and $[0, 30]$ dB, respectively. The SSN is generated by applying a speech-shaped spectrum onto white noise in the STFT domain. The speech-shaped spectrum is computed by averaging the magnitude spectra of 3,000 STFT frames of randomly chosen speech signals. For every mixture, two single-channel SSN signals are

generated independently, and they are multiplied by the square root of $\mathbf{\Omega}(f)$ (as defined in Eq. (2.87)) such that the resulting two-channel SSN is spatially diffuse. Real ambient noise which was recorded as part of the Vocadom project is included in the test set instead of SSN, at an SNR range of $[0-30]$ dB. The noise was recorded using a microphone pair of 10 cm spacing in three different apartments in a similar fashion as Kinoshita et al. (2016).
To compute the early reverberated signals, speech is convolved with the first $\tau_E = 50$ ms of the RIR. The direct signal is obtained using Eq. (3.1).

3.4.2 ANR Vocadom dataset

The ANR Vocadom dataset contains real data recorded as part of the ANR Vocadom project (Vacher et al., 2018). It was recorded in multiple rooms in a single apartment. It contains 11 native french speakers who uttered voice commands corresponding to actions in a home automation system. The commands are composed of a wake-up word followed by the action. The recordings corresponding to speakers S07, S08 and S09 are used in this work.
Each participant was allowed to choose a keyword among the following: *Allô Cirrus, Allô Messire, Chanticou, Dis Bérénio, Dis Hestia, Dis Téraphim, Dis Vesta, Dis Vocadom, Hé Cirrus, Ichefix, Minouche, Téraphim, Ulysse, Vocadom.* Actions such as placing a telephone call and controlling different objects in a smart home like television, lights, windows and doors were allowed.
The data was recorded under different domestic conditions — referred to as phases — each of which allows for different degrees of spontaneity in the behavior of the participants. In this work we are interested in Phase 3 in which the participants had to read a list of voice commands at different positions in the apartment. The data was recorded with different noise conditions such as interfering speech, vacuum cleaner, television, ventilator and shower. The spatial positions of the speakers and the microphone arrays were also recorded.
The data was recorded using four arrays of 4 MEMS[1] microphones (Fig. 3.10) and encoded as 16-bit, 44.1 kHz WAV files. It was downsampled to 16 kHz before computing the features. The microphones are placed on the vertices of a rectangle with 10 cm diagonal length. The signals captured at both diagonal microphone pairs were used for localization. One microphone array was mounted on the ceiling of each room in the apartment. For every utterance, the microphone array which is closest to the speaker is used for localization. The data contains strong reverberation with audibly poor SNR and SIR values.

[1] MP34DT01 Digital MEMS by ST Microelectronics.

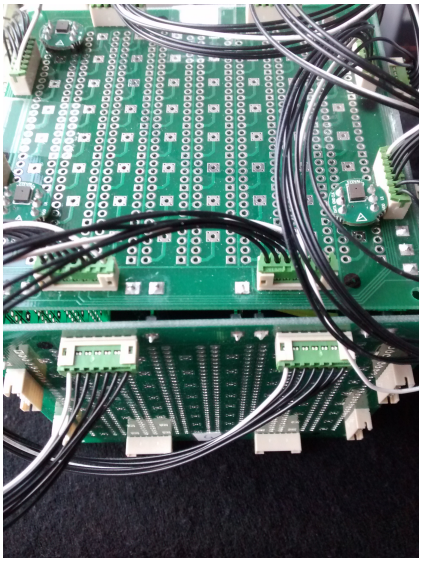


Figure 3.10: Microphone array used to record real data

3.4.3 Inria two-speaker mixture dataset

To have finer control on the positions of the speakers we recorded another real dataset[2]. Similar to the ANR Vocadom recording, the target speaker utters a sentence containing the keyword and an action.

Recording conditions: All the data was recorded in a single room. Two recording sessions were conducted, each with a different target speaker. The interference speaker remained the same for both sessions. The list of sentences to be spoken by the target speaker was displayed on a computer screen. The sentences are in French and a sample sentence list is shown in Table 3.1. The target speaker was given a keyboard to navigate the text sequence to be displayed. The timestamps of the key strokes were recorded in order to obtain the sentence boundaries. The interfering speaker read out a set of sentences from a book without stopping, ensuring the presence of interfering speech at all times. The first session was recorded using a single 4-MEMS microphone array and the second session was recorded with two such microphone arrays.

Speech recording protocol: Three positions were marked in the room at a distance of 2.16 m, 4.16 m and 3.94 m from the first microphone array. They were labeled as positions $1, 2$ and 3 respectively. These positions are at a distance of 3.59 m, 4.65 m and 3.59 m from the second microphone array. The target speaker uttered 10 sentences in each of these positions in the absence of the interfering speaker. These data are used to verify the ground truth localization results. Thereupon, the recordings were conducted

[2]Thanks to Elodie Gauthier, Manuel Pariente, Nicholas Furnon, Nicholas Turpault and Emmanuel Vincent for helping out with the recording.

Table 3.1: Example sentences spoken by the target speaker

Allô Cirrus, ouvre les rideaux de la salle de bain !
Il fait très beau dehors.
Allô Messire, augmentez la lumière du rez-de-chaussée !
Une ambulance, vite !
Vocadom, baissez le store de la cuisine !

with the following (target speaker, interference speaker) positions: $(1, 2), (2, 3)$ and $(3, 1)$. For each pair of positions, 50 sentences were recorded. The spatial coordinates of the microphones and the target and interfering speakers were recorded as well.
A total of 28 and 6.6 minutes of real data were recorded with and without an interfering speaker, respectively. To increase the number of segments available for testing, we do not restrict ourselves to the speech segments corresponding to the keyword. Instead, we collect all the speech signals containing the target speech (using ASR alignments) and chop them into non-overlapping segments of 0.5 s each. A total of $1,981$ such segments with an interfering speaker and 455 segments without an interfering speaker were obtained.

3.5 Experimental setup

3.5.1 Feature extraction

Features for *frame localization* : 100 ms segments corresponding to a single phone are picked from the reverberated target and interference speech. The duration and identity of the phone are obtained using the phonetic alignments obtained from the ASR system. It is ensured that the phones spoken by the target and interference are different. Delays arising due to the distance between the speakers and the microphones are incorporated in the alignments.
For every segment, a sinusoidal window is applied before computing a 1,600 point Fourier transform. The resulting CSIPD is 2-dimensional with shape 2×801. To compute the mask identifier, the magnitude spectra of the segments from both microphones are appended with the phone spectra to form input features of dimension 3×801.
A single segment from each noisy speech mixture is used for training and development in order to maximize the diversity of the training and development sets. For the test set, 15 such segments from the same mixture are taken and the DOA decision is made by pooling the DOAs across all segments.

Features for *sequence localization* : Speech segments corresponding to a word of duration 0.5 s at least are picked from both the target and interference signals. The respective signals are truncated if the duration of the word exceeds 0.5 s. The features for estimating the target identifiers are computed as detailed in Section 3.2.4.

3.5.2 ASR models

ASR alignments are computed using an HMM-GMM acoustic model with feature-space maximum likelihood linear regression (fMLLR) features (Gales, 1998). The model is trained on the clean training set of the Librispeech corpus, and used to align the training, validation and test datasets. The acoustic model contains 150,000 Gaussians and 5,751 senones.
Experiments are also conducted using noisy alignments to quantify the impact of bad alignment on the localization performance. These alignments are obtained by aligning a noisy ASR model on simulated noisy speech. Noisy speech is obtained in a similar fashion as described in Section 3.5.1.

3.5.3 Pooling DOA estimates

DOA estimates are provided for every frame in both the *sequence* and *frame localization* systems. These estimates are pooled to obtain a single DOA for an utterance. Multiple statistics were tried to pool the estimates, namely:

1. Median: The median of all the estimates is assigned as a DOA.
2. Maximum peak: Denoting by $p(n, \theta)$ the output of the DOA network (i.e., the posterior probability of DOA θ) in time frame n, the target DOA is found as

$$\widehat{\theta}_1 = \underset{\theta}{\operatorname{argmax}} \max_{n \in \{1, \cdots, N\}} p(n, \theta). \tag{3.14}$$

3. Weighted mean: The estimated DOAs per frame are weighted by the corresponding probabilities as

$$\widehat{\theta}_1 = \frac{1}{N} \sum_{n=1}^{N} \underset{\theta}{\operatorname{argmax}}\, p(n, \theta) \times \max_{\theta} p(n, \theta). \tag{3.15}$$

11 and 15 DOA estimates are pooled for the *sequence localization* and *frame localization* systems, respectively. The maximum peak estimate was found to work best for *frame localization* while the weighted mean estimate was preferred for *sequence localization* .

3.5.4 Experiments using real data

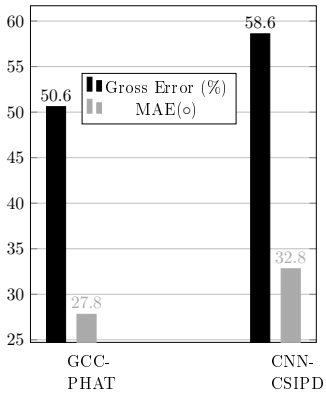
Since both real datasets contain speech in French language, all the models (ASR, mask estimation and DOA estimation models) are re-built using speech utterances from the Ester (Galliano et al., 2006) and Etape (Gravier et al., 2012) French speech corpora. The RIRs and noise described in Section 3.4.1 are reused to simulate the noisy speech.

3.6 Results and analysis

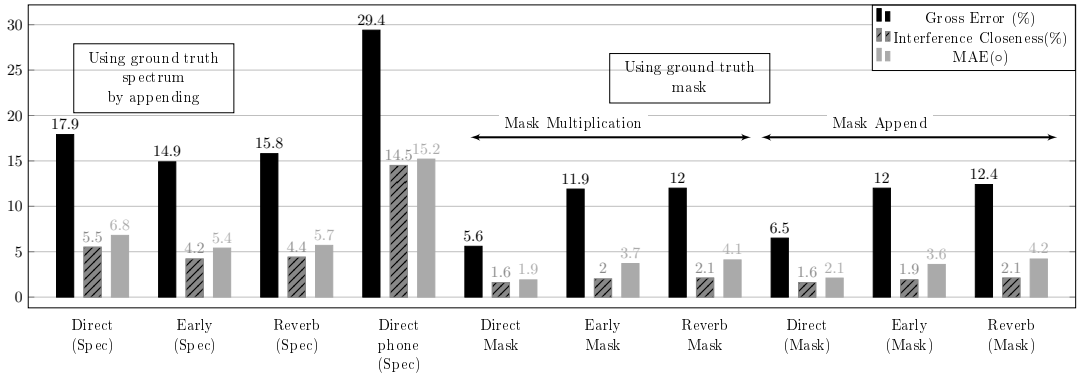
The three metrics described in Section 2.2.3 are used to evaluate the performance of the two localization systems, namely:

1. gross error rate, the threshold was set to $\theta_{\text{Th}} = 5°$, unless mentioned otherwise.

2. interference closeness rate,
3. mean absolute error.



(a) GCC-PHAT and *frame localization* system without target identifiers (CNN-CSIPD).



(b) *frame localization* system with ground truth target spectrum and mask identifiers.

Figure 3.11: Localization performance achieved by the *frame localization* system with ground truth target identifiers compared with GCC-PHAT and with the *frame localization* system without target identifiers on the simulated dataset.

3.6.1 Frame localization

We analyze the results of the *frame localization* system first. All results in this subsection are obtained on simulated data.

Baseline: The results obtained using the *frame localization* system without target identifiers are shown in Fig. 3.11a. This system obtained a gross error rate of 58.6%: it was able to find patterns in the CSIPD features to localize the two speech sources but unable to distinguish the target from the interference. For comparison, GCC-PHAT obtained a slightly lower (but still poor) target error rate of 50.6%, which is considered as the baseline in the following.

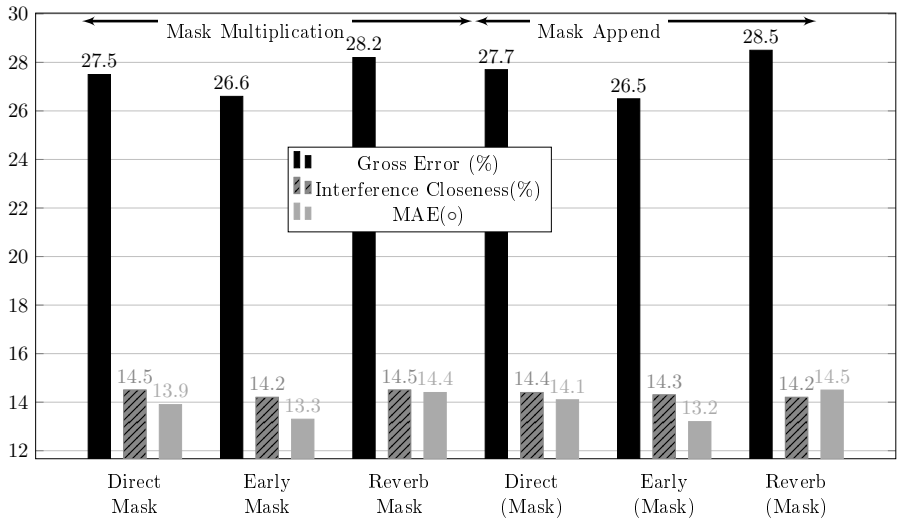


Figure 3.12: Localization performance achieved by the *frame localization* system with estimated target identifiers on the simulated dataset.

Using ground truth target identifiers: The results achieved by the *frame localization* system with ground truth target identifiers are shown in Fig. 3.11b. Appending the ground truth direct spectrum identifiers results in a drastic reduction in all the metrics. The low interference closeness rate of 5.5% shows that the model was able to use the spectral information to identify the relevant target time-frequency bins. A similar reduction in metrics was observed while using the ground truth early and reverberated target spectra. The early target spectrum outperformed the two other target identifier spectra. Appending the phone spectra, which does not assume the availability of the target signal, resulted in an error rate of 29.4% which is a 42% relative improvement over the baseline. The reduction in the interference closeness rate to 14.5% shows that the phonetic information of the speech spoken by the target can be directly used by the model to identify the target DOA.

Mask-based ground truth identifiers performed better than spectrum-based ground truth identifiers. Multiplying CSIPD features with the direct mask identifier gave an error rate of 5.6% which is a 69% relative improvement over the direct spectrum target identifier. The very low interference closeness rate of 1.6% further shows that the direct target mask removes any confusion caused by the interfering speech to estimate the target DOA. Multiplying the early or reverberated mask with CSIPD features resulted in increased error rates even though the change in the interference closeness rate was minimal. This is due to the echoes (both early and late) corrupting the masks. Appending the ground truth mask resulted in either equal or worse error rates compared to multiplying it with CSIPD features even though the interference closeness rates were similar. Since mask-based identifiers were found to perform better than spectrum-based identifiers, only the former are considered in the following experiments.

Using estimated target identifiers: The results achieved by the *frame localization* system with estimated target identifiers are shown in Fig. 3.12. The estimated early mask gave better performance than the estimated direct or reverberated masks. An error rate of 26.6% was obtained which is better than appending only the phonetic spectrum, by a relative margin of 9%. This is probably due to the fact that early mask estimation requires the removal of the late reverberation only, while direct mask estimation requires the removal of both the early echoes and the late reverberation. The results obtained by multiplying and appending the estimated mask were similar.

Ideal keyword for localization: Since the *frame localization* system estimates the DOA for every frame (of 100 ms duration) and since every frame is associated with a single phone, we can study the influence of the phone class on the localization performance. Table 3.2 shows the gross error rate achieved by the *frame localization* system with ground truth direct mask identifiers for the four best and four worst performing phones. The extensions {*_B, _I, _E*} describe the positions of a phone in a word, namely at the beginning, middle and end of the word, respectively.

Table 3.2: Gross error rate (%) achieved by the *frame localization* system by multiplying CSIPD features with the ground truth direct mask on the simulated dataset as a function of the phone class. Only the best and the worst performing phone classes are shown.

Phone	CH_I	CH_B	Z_B	SH_B	NG_E	N_E	M_E	B_B
Gross error rate	1.5	1.6	1.8	1.8	19.4	21.1	21.3	24.5

The best performance is observed for fricatives such as *CH, DH, F, TH, V, Z, ZH, S, SH, JH* with an average error rate of 3.8%. Nasal sounds such as *L, M, N, NG, R, ER* perform worst with an average error rate of 14.5%. Vowels (*AA, AE, AH, AO, AW, AY, OW, OY, UW, IH, IY, EY*) and plosives (*B, D, G, K, P, T*) averaged 6.7% and 9.9% respectively.
The better localization performance with fricatives can be attributed to their wideband nature which results in target speech dominance in a larger number of time-frequency bins. This contradicts the usual design of wake-up words where high-energy sounds such as vowels are preferred to low-energy sounds such as fricatives. A compromise would be to have a two-word wake-up phrase with the first word containing multiple vowels and the second word containing multiple fricatives.
Phones at the beginning of the word performed best with an average error rate of 5.6%, while phones at the end and in the middle had an average error rate of 10.7% and 6.7% respectively.

3.6.2 Sequence localization

Using ground truth ASR alignments: The results achieved by the *sequence localization* system on the simulated dataset are shown in Fig. 3.13. In this experiment the CSIPD features were multiplied with either ground truth masks or masks estimated using ground

truth ASR alignments obtained from the ground truth target speech signal s_1. Due to memory constraints, the models were trained using only 80% of the training data.
GCC-PHAT gave a gross error rate of 50.9%, an interference closeness rate of 29.2% and a mean absolute error of 27.5°. The *sequence localization* system without target identifiers reduced the error rate to 51.4% compared to 58.6% for *frame localization* . This shows that the *sequence localization* system is better at localizing a source than the *frame localization* system.

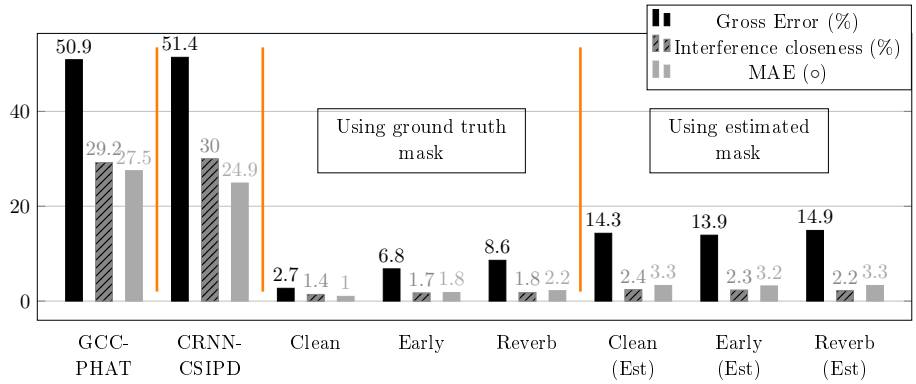


Figure 3.13: Localization performance achieved by the *sequence localization* system on the simulated dataset using multiplication with either ground truth masks or masks estimated using ground truth ASR alignments, compared with GCC-PHAT and with the *sequence localization* system without target identifiers (CRNN-CSIPD).

The error rates obtained using ground truth masks further validate this observation. The error rate of 2.7% obtained using the ground truth direct mask, which is a 52% relative improvement over a similar *frame localization* model, is of particular interest since the localization decisions of the *sequence localization* system are based on 0.5 s of speech only compared to 1.5 s (15 frames of 0.1 s each) for the *frame localization* system. Early and reverberated ground truth masks gave similar improvements compared to the respective *frame localization* results.
The impact of *sequence localization* is more apparent in the results with the estimated masks. An error rate of 13.9% was obtained using the estimated early mask which is a 48% relative improvement over *frame localization* . This improvement can be attributed to both the improvements in the mask estimation stage as well as the DOA estimation stage. Since the difference in the interference closeness rate between the estimated mask and the ground truth mask (2.3% vs. 1.7%) is very small, we can conclude that the estimated mask provides all required information needed to differentiate the target from the interference.

Impact of noisy alignments: Noisy phonetic alignments can also be obtained by aligning the mixture $\mathbf{x}$ using an HMM-GMM based ASR model as explained in Section 3.5.2. The obtained results using *sequence localization* are shown in Table 3.3. Since alignments impacts only the mask estimation, the results only show the DOA with the estimated

masks.

Table 3.3: Localization performance achieved by the *sequence localization* system on the simulated dataset using multiplication with masks estimated using noisy ASR alignments.

	Clean Mask	Early Mask	Reverb Mask
Gross error rate (%)	15.2	14.8	15.9
Interference closeness rate (%)	2.5	2.4	2.3
Mean absolute error (°)	3.8	3.6	3.9

A gross error rate of 14.8% was obtained by estimating an early mask with noisy alignments, a 6% relative degradation when compared to mask estimated with clean alignments. Similar degradations in the performance were observed when the direct and reverberated masks were estimated. However, the change in the interference closeness rate was negligible. This could be due to the robustness induced by the phone sequence information used by the *sequence localization* models.

3.6.3 Performance on real data

We now report the localization performance achieved by the *sequence localization* system on real data. All models were trained using simulated data as described in Section 3.5.4.

On the ANR Vocadom dataset: The mean absolute error obtained on the ANR Vocadom dataset for different noise conditions is shown in Table. 3.4. The error range of $[11 - 15]^\circ$ for both the *sequence localization* system and GCC-PHAT in noiseless conditions shows the difficulty of localizing the speaker in this dataset.

Table 3.4: Mean absolute error (°) achieved by the *sequence localization* system using multiplication with masks estimated using noisy ASR alignments (CRNN) and by GCC-PHAT on the ANR Vocadom dataset for different speakers in different noise conditions. The best system for each speaker and each noise condition is marked in bold.

Noise conditions	**S07**		**S08**		**S09**	
	CRNN	GCC-PHAT	CRNN	GCC-PHAT	CRNN	GCC-PHAT
No Noise	**11.9**	12.0	**14.4**	15.5	**11.25**	15.2
Radio	**26.4**	31.0	**15.9**	23.0	23.1	**22.6**
Interfering speaker	**15.2**	19.8	31.3	**27.4**	25.5	**23.7**
TV	**38.7**	44.3	36.9	**35.6**	**34.3**	35.7
Ventilator	**12.7**	21.6	**21.5**	29.2	**17.6**	22.8
Kitchen hood + TV	**28.8**	29.6	**34.2**	35.5	27.0	**19.9**

The proposed *sequence localization* system outperforms GCC-PHAT across most speaker and noise conditions. Nevertheless, it performs poorly for mixtures containing interfering speakers for speakers S08 and S09, possibly due to wrong ASR alignments. Note that

even though the models were trained to handle interfering speech only, they seem to generalize well across other noise conditions. The mean absolute error was the highest in the presence of TV noise. This is mainly due to the poor SNR of the signal.
The high error rate could either be due to the reverberation or to human errors while annotating the positions of the speakers and the microphone positions.

Table 3.5: Gross error rate (%) achieved by GCC-PHAT, the *sequence localization* system without target identifiers (CRNN-CSIPD), and the *sequence localization* system with estimated direct mask identifier on the subset of the Inria mixture dataset containing only the target speaker. A larger threshold of 10° was used to compute the error rate in order to account for the human errors in marking the speaker and microphone positions.

Speaker-to-microphone distance (m)	Sample count	GCC-PHAT	CRNN-CSIPD	Estimated direct mask
2.16	113	8.7	8.0	5.3
3.59	90	23.3	18.9	14.4
3.94	87	47.2	49.4	35.6
4.16	114	23.6	28.0	12.2
4.65	51	53.8	22.0	25.8

On the Inria two-speaker mixture dataset: Table 3.5 shows the gross error rate on the subset of the Inria real mixture dataset when only the target speaker spoke without the presence of the interfering speaker. For a small distance between the speaker and the microphone array (2.16 m), GCC-PHAT and *sequence localization* without target identifiers perform similarly, with a gross error rate in the order of 8%. However an improved error rate of 5.35% was observed by including the direct mask target identifier. A similar trend was observed for other distances as well. For a very large distance (4.65 m), an error rate reduction of 52% was observed with the usage of the direct mask target identifier. This shows the importance of the keyword for localization even in noiseless but reverberated speech signals.
Figure 3.14 shows the performance of the *sequence localization* system on the Inria two-speaker real mixture dataset in the presence of the interfering speaker. The results are displayed based on the distance between the target and the interfering speaker. The dataset contained 6 such distances as shown in Table 3.6.
Negative distances refer to the cases when the target speaker was closer to the microphone array compared to the interfering speaker. This can be seen as performance comparison in different SIR conditions. In general, the localization errors in real data were significantly worse than the errors with simulated data. Similar performance degradations with real data were reported by Perotin et al. (2018), who showed that the performance of a model trained with simulated RIRs degraded when tested with real data. Nevertheless, it was also shown that the localization performance achieved on data simulated with real RIRs and real recorded data was similar. To the best of our knowledge, all other previous

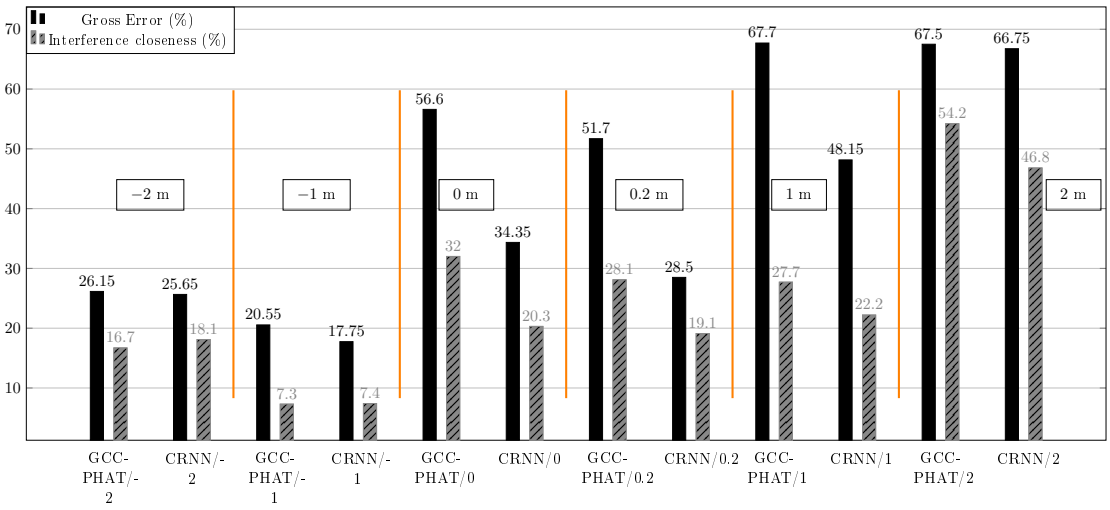


Figure 3.14: Localization performance achieved by GCC-PHAT and the *sequence localization* system with estimated direct mask identifier on the subset of the Inria mixture dataset containing two speakers as a function of the difference in distance between the target speaker and the interfering speaker. Negative distances indicate that the target is closer to the microphone pair than the interfering speaker.

works on learning-based localization have reported results on simulated data only.
In high SIR conditions when the target speaker is close to the microphone (< 0), GCC-PHAT and the proposed method perform equally well. When the distances are similar (0 and 0.2 m), which corresponds to an SIR in the order of 0 dB, a marked improvement in the localization performance can be seen with our proposed method. A reduction of both the gross error rate and the interference closeness rate in the order of 40% relative was observed. A similar trend can be observed with negative SIR (distance = 1 m). However, with very low SIR values (distance = 2 m), our proposed method reduced the interference closeness rate by 14% relative implying that the estimated mask was still useful to differentiate the target from the interfering speaker.

Table 3.6: Distance of the target and interfering speakers from the microphone array for the results shown in Fig. 3.14.

Difference (m)	Target speaker (m)	Interfering speaker (m)
-2	2.16	4.16
-1	3.59	4.59
0	3.59	3.59
0.2	4.16	3.96
1	4.65	3.65
2	3.94	1.94

3.7 Conclusion

In this chapter methods to exploit the text spoken by a target speaker to improve speaker localization performance in a multi-speaker environment were proposed. The spoken text is forced-aligned with the corrupted speech using ASR models and the obtained phonetic alignments are used to compute a target identifier. This target identifier is used along with CSIPD features to localize the target.
Two methods, namely *frame localization* and *sequence localization* were proposed. In the *frame localization* method, a DOA decision is made on a long-duration speech segment representing a single phone whereas the *sequence localization* method works on speech segments containing multiple phonetic labels. *sequence localization* was found to outperform *frame localization* in both idealistic conditions where the target identifiers are computed using the direct target speech and in real conditions where the target identifiers are computed using the corrupted speech and the phonetic alignments. The computed target identifier using sequence localization was found to be very effective at reducing the interference closeness rate. Testing the system on real data yielded better localization performance than GCC-PHAT. Furthermore, we found that localization works best with fricatives compared to other types of phones such as plosives and nasals.

4 Speech separation

4.1 Introduction

In this chapter, we consider the problem of multichannel speech separation. As we have seen in Chapter 2, the usual approach is to estimate the second-order statistics (covariance matrices) of all speech and noise sources and to derive a beamformer to separate the speakers. Different methods have been proposed to estimate the target speech and noise covariance matrices based on the magnitude spectra or the phase differences between the microphones. Explicit speaker location estimates have also been employed. Perotin et al. (2018) and Chen et al. (2018a) assume that the speaker location is known and beamform the microphone signal towards that speaker. Perotin et al. (2018) used the magnitude spectra of DS beamformed signals corresponding to the location of the speaker and the interfering speech as inputs to a DNN to estimate a mask whereas Chen et al. (2018a) used features derived from the phase information along with the magnitude spectrum as inputs. An initial study of estimating the speaker location and using it to obtain masks was conducted by Perotin (2019). A similar approach is proposed by Taseska and Habets (2017) where the so-called speech presence probability (SPP) is estimated using DOA information with a minimum Bayes risk detector. The speech and noise statistics are then derived from the SPP. No DNNs are used to estimate either the DOA or the mask. In this chapter, we focus on two tasks. The first task is speech extraction, where we extract a single speaker from a mixture containing two speakers and noise, using the DOA corresponding to that speaker. Experiments are conducted using ground truth DOAs as well as DOAs estimated with GCC-PHAT and evaluated using ASR metrics. A related study on analyzing the impact of localization errors on ASR metrics was done by Barfuss and Kellermann (2016), but under limited acoustic conditions and vocabulary size. The second task is speech separation where we propose to iteratively estimate multiple speech sources using a Speaker LOcalization Guided Deflation (SLOGD) approach. The concept of deflation was introduced in blind source separation and refers to the iterative estimation and removal of one source at a time (Delfosse and Loubaton, 1995). Our intuition is that the dominant sources are easier to estimate in the first few iterations and, once they have been removed from the mixture, it becomes easier to estimate the other sources. In order to implement this approach in a modern deep learning based multichannel signal processing framework, we estimate the DOA of a first speaker, compute the beamformed signal, and use features of that signal as inputs to a DNN to derive the corresponding mask. The DOA and the mask of the next source are estimated by removing the first speaker from the mixture's features using the first estimated mask, and so on. The mask estimation network is trained on beamformed signals computed from potentially

erroneous DOA estimates, which results in increased robustness to localization errors. A few earlier works have proposed to iteratively estimate the sources using DNN in a single-channel setting (Kinoshita et al., 2018; Takahashi et al., 2019). To the best of our knowledge, this is the first such study in a multichannel setting, where we estimate both the DOAs and the masks of all speakers. DOA estimation is crucial since location-based speech separation is the only method which works well in the presence of reverberation and noise as shown by Chen et al. (2018a).

To sum up, this chapter provides the following contributions:

1. We create a new multichannel, multispeaker, reverberated, noisy dataset which extends the original WSJ0-2mix single-channel, non-reverberated, noiseless dataset (Hershey et al., 2016) to the strong reverberation and noise conditions and the Kinect-like microphone array geometry used in CHiME-5 (Barker et al., 2018). This allows us to use the real noise captured as part of the CHiME-5 dataset, thereby making the simulated dataset quite realistic and challenging. The code to re-create the dataset is made publicly available at `https://github.com/sunits/Reverberated_WSJ_2MIX`.
2. On this dataset, we perform speech extraction using the ground truth location of the speakers and we evaluate the resulting ASR performance on the extracted speech. We also conduct experiments to evaluate the impact of localization errors on the speech extraction performance using either artificially induced errors or DOAs estimated by GCC-PHAT.
3. We propose an algorithm — which we refer to as SLOGD — to separate multiple speakers in the presence of localization errors.

The rest of the chapter is organized as follows. Section 4.2 introduces the proposed framework for speech separation using speaker localization information. Section 4.3 details the proposed SLOGD method. Section 4.4 describes the dataset used in this work, Section 4.5 describes the experimental setup and the obtained results are discussed in Section 4.6. We conclude in Section 4.7.

4.2 Speech extraction using location information

Recalling the signal mixing model, the multichannel signal

$$\mathbf{x}(t) = [x_1(t), \ldots, x_I(t)]^T \tag{4.1}$$

captured at I microphones can be expressed as:

$$\mathbf{x}(t) = \sum_{j=1}^{J} \mathbf{c}_j(t) \tag{4.2}$$

where

$$\mathbf{c}_j(t) = [c_{1\,j}(t), \ldots, c_{I\,j}(t)]^T \tag{4.3}$$

is the spatial image of source j, i.e., the signal emitted by the source and captured at the microphones. There are J sources out of which J' are speech sources and the remaining $J - J'$ sources are noise.

This general formulation is valid for both point sources as well as diffuse noise. For point sources such as human speakers, the spatial image can be expressed as the linear convolution of the room impulse response (RIR)

$$\mathbf{a}_j(t) = [a_{1\,j}(t), \ldots, a_{I\,j}(t)]^T \tag{4.4}$$

with a single-channel source signal $s_j(t)$:

$$\mathbf{c}_j(t) = \sum_{\tau=0}^{\infty} \mathbf{a}_j(\tau)\, s_j(t-\tau). \tag{4.5}$$

Under the narrowband approximation, this can be written in the time-frequency domain as

$$\mathbf{c}_j(n,f) = \mathbf{a}_j(f)\, s_j(n,f). \tag{4.6}$$

In speech extraction, we are interested in obtaining a single source only, say source $j = 1$, implying that the other sources can be considered as noise $\mathbf{v}$, i.e.,

$$\mathbf{x}(n,f) = \mathbf{c}_1(n,f) + \mathbf{v}(n,f) \tag{4.7}$$

$$\mathbf{v}(n,f) = \mathbf{u}(n,f) + \sum_{j=2}^{J'} \mathbf{c}_j(n,f). \tag{4.8}$$

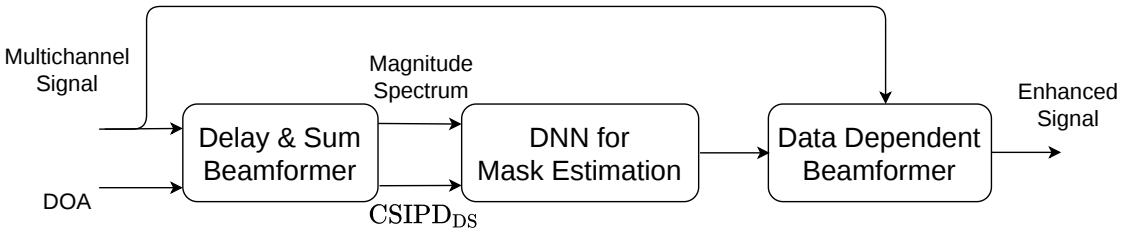


Figure 4.1: Speech separation pipeline using the DOA information.

Our objective is to estimate the spatial image of each source given its (known or estimated) DOA. An overview of our speaker location guided speech separation system is shown in Fig. 4.1. This system comprises three steps:

1. delay-and-sum (DS) beamforming,
2. mask estimation,
3. adaptive beamforming.

These steps are detailed in the subsections below.

4.2.1 DS beamforming

Given that the target source is located in the far field, the corresponding TDOA between microphones i and i' can be obtained as

$$\Delta_{i\,i'\,1} = \frac{\ell_{i\,i'} \cos(\theta_{i\,i'\,1})}{c} \tag{4.9}$$

where $\theta_{i\,i'\,1}$ is the DOA of the source with respect to the microphone pair (i, i'), $\ell_{i\,i'}$ is the distance between the two microphones, and c is the velocity of sound.
The relative steering vector with respect to a reference microphone (in the following, microphone 1) can be computed as

$$\tilde{\mathbf{d}}_1(f) = \begin{bmatrix} 1 \\ e^{-2j\pi\,\Delta_{211}\,\nu_\mathrm{f}} \\ \vdots \\ e^{-2j\pi\,\Delta_{I\,11}\,\nu_\mathrm{f}} \end{bmatrix}. \tag{4.10}$$

The output of the DS beamformer for source 1 can then be obtained as

$$\widehat{c}_{1,\mathrm{DS}}(n, f) = \tilde{\mathbf{d}}_1^H(f)\,\mathbf{x}(n, f). \tag{4.11}$$

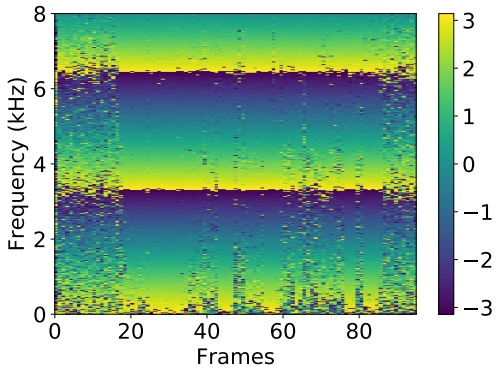
The number of time-frequency bins dominated by the localized speaker is higher in $\widehat{c}_{1,\mathrm{DS}}$ than in $\mathbf{x}$. We hence use $\widehat{c}_{1,\mathrm{DS}}$ to compute a time-frequency mask corresponding to the target speaker.

4.2.2 Time-frequency mask estimation

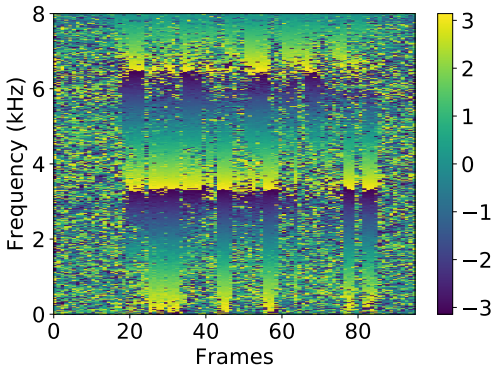
The magnitude spectrum of $\widehat{c}_{1,\mathrm{DS}}$ and the sine and cosine of its phase difference with respect to the reference microphone — denoted as $\mathrm{CSIPD}_\mathrm{DS}$ in Fig. 4.1 — are used as inputs to a DNN to estimate the time-frequency mask corresponding to the localized speaker.
The magnitude spectrum of $\widehat{c}_{1,\mathrm{DS}}$ has previously been used by Perotin et al. (2018) and Chen et al. (2018a) as input to the network, and Chen et al. (2018a) also concatenated features derived from the phase difference of each microphone pair. Using the cosine and sine of the phase difference between $\widehat{c}_{1,\mathrm{DS}}$ and a reference microphone, i.e

$$\phi_{\widehat{c}_{1,\mathrm{DS}}}(n, f) = \angle\,\widehat{c}_{1,\mathrm{DS}}(n, f) - \angle\,x_1(n, f), \tag{4.12}$$

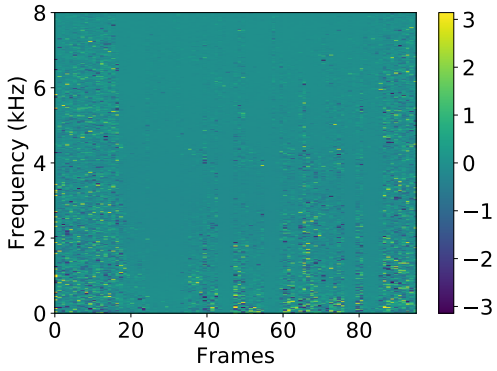
as a feature may not seem intuitive at first and requires further justification. Figure 4.2 shows the information captured by this phase difference. Fig. 4.2b shows the phase difference of the direct component (without reverberation) of a speech source between two microphones placed at a distance of 0.226 m in the presence of noise. The phase difference is perturbed in the time-frequency bins dominated by noise.
Fig. 4.2d shows the phase difference of the beamformed signal with respect to the signal at the reference microphone. The phase difference in the time-frequency bins dominated by speech is now close to 0, and a speech-like pattern can be observed in these bins. In the presence of reverberation, the speech pattern is less clearly visible before or after DS beamforming as shown in Fig. 4.3. Nevertheless, we argue that the phase difference contains useful information regarding the source which can be leveraged by a DNN in addition to the magnitude spectrum of the DS beamformer output in order to estimate a better time-frequency mask.



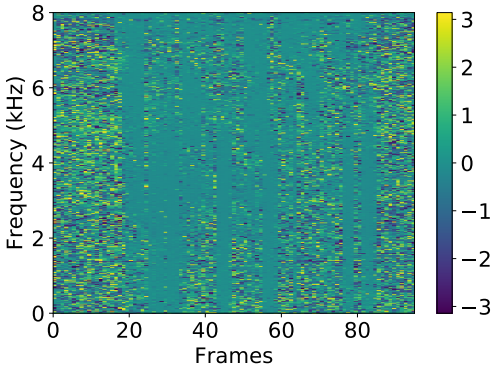
(a) Phase difference between the direct components of the signal in the absence of noise, i.e., $\angle \frac{c_{21}^D(n,f)}{c_{11}^D(n,f)}$.



(b) Phase difference between the direct components of the signal in the presence of noise, i.e., $\angle \frac{c_{21}^D(n,f)+\mathrm{u}_2(n,f)}{c_{11}^D(n,f)+\mathrm{u}_1(n,f)}$.



(c) Difference between the phase of the beamformed signal with respect to the phase of the signal of a reference microphone in the absence of noise, i.e., $\phi_{\hat{c}_{1,\mathrm{DS}}}(n,f)$ with $\mathbf{x}(n,f) = \mathbf{c}_1(n,f)$.

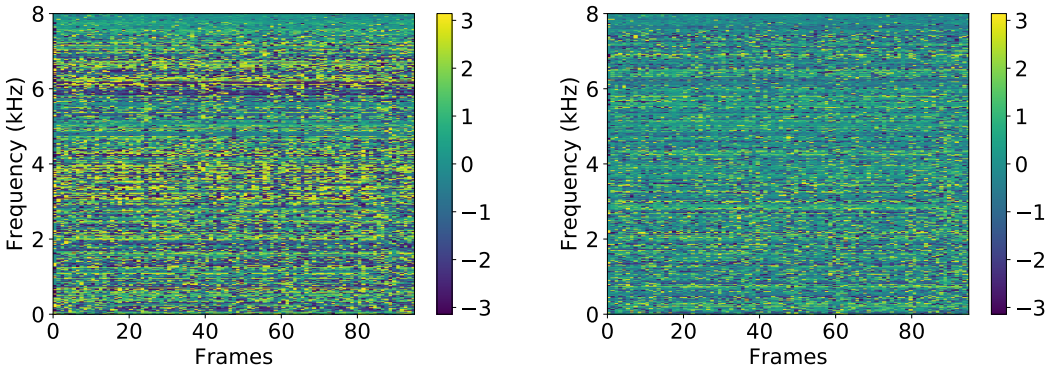


(d) Difference between the phase of the beamformed signal with respect to the phase of the signal of a reference microphone in the presence of noise, i.e., $\phi_{\hat{c}_{1,\mathrm{DS}}}(n,f)$ with $\mathbf{x}(n,f) = \mathbf{c}_1(n,f) + \mathbf{u}(n,f)$.

Figure 4.2: Impact of DS beamforming on the interchannel phase difference.

Since the phase difference is defined modulo 2π only, we use its cosine and sine as features. These features are given as inputs along with the magnitude spectrum of $\hat{c}_{1,\mathrm{DS}}$ to train a DNN to estimate the IRM corresponding to the spatial component of the localized speaker. A two-layer Bi-LSTM is used as the DNN with mean square error as the loss function. The DNN is optimized using Adam.

We highlight the fact that the dimension of the input features to train the mask estimation network does not depend on the number of microphones or the microphone array geometry.



(a) Before delay and sum beamforming.

(b) After delay-and-sum beamforming.

Figure 4.3: Phase patterns after including reverberation in Fig. 4.2.

4.2.3 Adaptive beamforming

The mask $\widehat{\mathcal{M}}_1$ output by the DNN for the target source can be used to estimate the covariance matrix of that source as

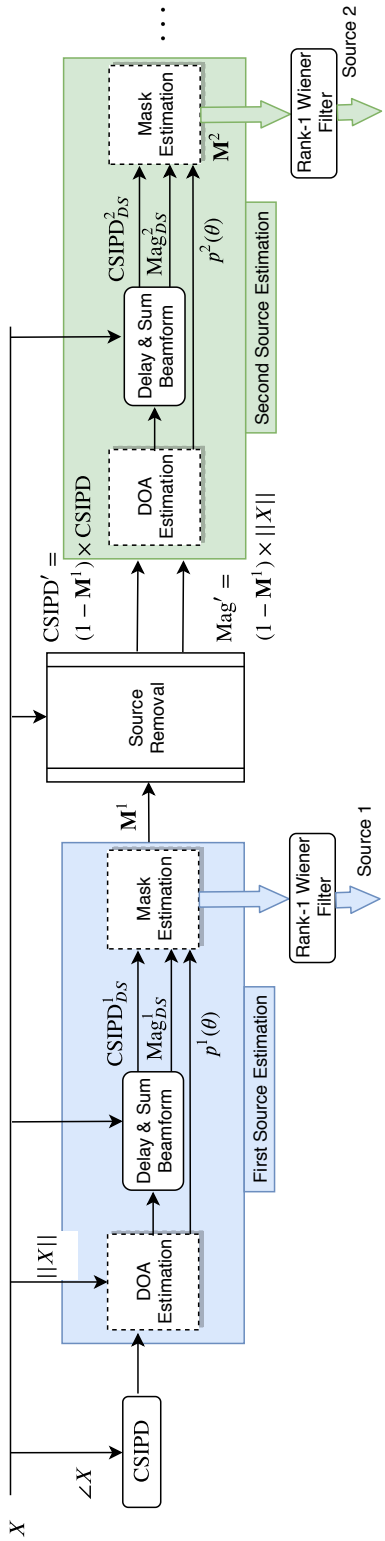
$$\mathbf{R}_{\mathbf{c}_1}(f) = \sum_n \widehat{\mathcal{M}}_1(n,f)\,\mathbf{x}(n,f)\,\mathbf{x}^H(n,f). \tag{4.13}$$

$$\mathbf{R}_{\mathbf{v}}(f) = \sum_n (1-\widehat{\mathcal{M}}_1(n,f))\,\mathbf{x}(n,f)\,\mathbf{x}^H(n,f). \tag{4.14}$$

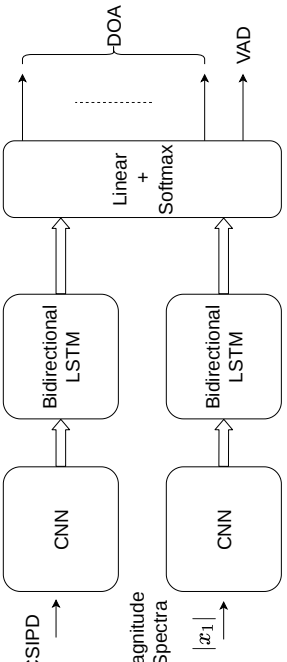
An adaptive beamformer, i.e., a beamformer depending on the above statistics rather than the spatial location, is applied to the mixture signal $\mathbf{x}(n,f)$ to recover the sources. The output of the beamformer is $\mathbf{w}^H(n,f)\mathbf{x}(n,f)$. Different beamformers can be defined based on the chosen optimization criterion (Wölfel and McDonough, 2009; Gannot et al., 2017). In this work we consider the generalized eigenvalue (GEV) beamformer (Warsitz and Haeb-Umbach, 2007), the speech distortion weighted multichannel Wiener filter (SDW-MWF) (Spriet et al., 2004), and the rank-1 constrained multichannel Wiener filter (R1-MWF) (Wang et al., 2018b) (See Section 2.3.3 for details).

4.3 Speech separation using location information

Localization errors corrupt both the magnitude and phase information of the DS beamformed signal $\widehat{c}_{j,\mathrm{DS}}(n,f)$. Even when the location estimates are perfect, the signal-to-interference ratio (SIR) has a big impact on the separation performance. A lower SIR makes it harder for the network to estimate the mask even when the true DOA is known. In this section we propose SLOGD — a deflation strategy where we iteratively estimate sources and remove them from the mixture before estimating the next source as shown in Fig. 4.4a and detailed below.



(a) Iterative estimation of DOA and mask.



(b) DOA estimation network of Fig. 4.4a.

Figure 4.4: SLOGD approach for speech separation.

4.3.1 Estimating the first source

Step 1: Estimating the DOA of the first speaker In the first step we estimate the DOA of the first speaker using a DNN.

The inputs to the network consist of 12 CSIPD feature vectors — 2 per pair of microphones between the 4 microphones in the array — as well as the magnitude spectrum of the first microphone.

Similar to Chapter 3, the DOA estimation network is trained as a classifier to estimate the DOA within a discrete grid of DOAs in every time frame. A non-speech class is also included at the output so that the network can classify a frame as non-speech if no speaker is active in that particular frame, thereby providing voice activity detection (VAD) information.

The network architecture to estimate the DOA is shown in Fig. 4.4b. It contains convolutional layers followed by a Bi-LSTM layer to process the CSIPD and the magnitude spectrum features independently. The phase and magnitude representations are fused together using a linear layer.

The intuition behind the architecture of Fig. 4.4b is that CSIPD features encode spatial information which is useful to estimate the DOA as shown in Chapter 3 whereas the magnitude spectrum provides additional speech activity information which is useful to perform VAD.

If the DOA grid is divided into P classes indexed by p, the output of the network for each time frame n is

$$\mathbf{p}_1(n) = \text{DOA\_DNN}_1([\mathbf{CSIPD}_{\text{DS}}(n), |\,\mathbf{x}_1(n)|]), \tag{4.15}$$

where

$$\mathbf{p}_1(n) = [\mathrm{p}_1(n,1), \ldots, \mathrm{p}_1(n,\mathrm{P}+1)] \tag{4.16}$$

$$\mathbf{CSIPD}_{\text{DS}}(n) = [\text{CSIPD}_{\text{DS}}(n,1), \ldots, \text{CSIPD}_{\text{DS}}(n,F)] \tag{4.17}$$

$$\mathbf{x}_1(n) = [x_1(n,1), \ldots, x_1(n,F)]. \tag{4.18}$$

The frame-level DOA probabilities estimated by the network are averaged across time, the DOA class corresponding to the highest probability is found, i.e.,

$$\widehat{p}_1 = \underset{p\in\{1,\ldots,\mathrm{P}\}}{\text{argmax}} \frac{1}{N}\sum_n \mathrm{p}_1(n,p) \tag{4.19}$$

and the corresponding $\widehat{\theta}_1 = \theta_{\widehat{p}}$ is used as the first speaker DOA where θ_p is the DOA value corresponding to the p-th class. The maximum in Eq. (4.19) is taken only along the output dimensions corresponding to the DOA classes while ignoring the dimension corresponding to VAD.

The cost function to train the DOA network contains two parts, namely

1. The cost associated with the VAD obtained using cross-entropy across frames as

$$\mathcal{L}_{\text{VAD}} = \sum_n -(1 - \text{VAD}(n)) \log \mathrm{p}_1(n, \mathrm{P}+1) \tag{4.20}$$

where $\text{VAD}(n) \in \{0, 1\}$ denotes the absence or presence of speech.

2. The cross-entropy cost associated with the speaker location pooled across time and computed as

$$\mathcal{L}_{\text{DOA}_1} = \min_j \sum_{p=1}^{\text{P}} -\log\left(\frac{1}{N}\sum_n \text{p}_1(n,\theta)\right)\mathbb{I}_{\theta_j}(p) \tag{4.21}$$

where $\mathbb{I}_{\theta_j}(p)$ is the one hot representation of the DOA defined as

$$\mathbb{I}_{\theta_j}(p) = \begin{cases} 1 & \text{if } \theta_p = \theta_j \quad \forall p \in \{1,\ldots,\text{P}\} \\ 0 & \text{otherwise.} \end{cases} \tag{4.22}$$

The intuition behind Eq. (4.21) is that the least cost will be associated with the speaker whose DOA is easier for the DNN to estimate. Similar ideas have been used by Kinoshita et al. (2018) to estimate masks in a single-channel multi-speaker speech mixture.
The total loss to train DOA_DNN$_1$ is therefore

$$\mathcal{L}_{\text{DOA\_DNN}_1} = \mathcal{L}_{\text{DOA}_1} + \mathcal{L}_{\text{VAD}}\,. \tag{4.23}$$

Step 2: Estimating the first mask Given the estimated DOA $\widehat{\theta}_1$, we compute the corresponding time-frequency mask $\widehat{\mathcal{M}}_1$ using another DNN as described in Section 4.2. Apart from the magnitude spectrum Mag^1_{DS} of the beamformed signal and its phase difference $\text{CSIPD}^1_{\text{DS}}$ with respect to the first channel, the output of the DOA network $\text{p}_1(n,p)$ is also concatenated and fed as input to the DNN, i.e,

$$\widehat{\mathcal{M}}_1(n) = \text{MASK\_DNN}_1([\mathbf{Mag}^1_{\text{DS}}(n), \mathbf{CSIPD}^1_{\text{DS}}(n), \mathbf{p}_1(n)]). \tag{4.24}$$

where MASK_DNN$_1$ is the DNN used to estimate the mask corresponding to the first speaker. We append the output of the DOA estimation network with the hope that it gives additional information regarding the variance of the estimated speaker DOA, which can potentially be exploited by MASK_DNN$_1$ for better mask estimation.

4.3.2 Estimating the second source

Step 3: Removing the estimated source There are multiple ways to remove the estimated source from the mixture. Kinoshita et al. (2018) computed a remainder mask after each iteration and appended it to the network inputs to estimate the next source. In this work we use a similar idea wherein we compute a remainder mask $(1-\widehat{\mathcal{M}}_1)$ but instead of appending it to the network inputs we multiply it with the magnitude spectrum and the CSIPD features before feeding them as inputs to the following DOA estimation and mask estimation stages. As shown in Chapter 3, mask multiplication performs better than mask concatenation for speaker localization.

Step 4: Estimating the second DOA Similarly to Step 1 in Section 4.3.1, we estimate the DOA of the second speaker using the CSIPD and the magnitude of the original signal after removing the estimated source in Step 3.

$$\mathbf{p}_2(n) = \text{DOA\_DNN}_2([(1-\widehat{\mathcal{M}}_1)(n) \times \mathbf{CSIPD}(n), (1-\widehat{\mathcal{M}}_1(n)) \times |\,\mathbf{x}_1(n)|]). \tag{4.25}$$

DOA_DNN_2 also estimates the VAD along with the probability for each DOA class. The estimated DOA class of the second speaker is

$$\widehat{p}_2 = \underset{p\in\{1,\ldots,\text{P}\}}{\text{argmax}} \frac{1}{N}\sum_n \text{p}_2(n,p). \tag{4.26}$$

Step 5: Estimating the second mask Similarly to Step 3, we apply DS beamforming using the estimated DOA and derive the mask for the second speaker as

$$\begin{aligned}\overline{\mathcal{M}}_2(n) = \text{MASK\_DNN}_2([(1-\mathcal{M}_1(n)) \times \mathbf{Mag}^2_{\text{DS}}(n),\\ (1-\mathcal{M}_1(n)) \times \mathbf{CSIPD}^2_{\text{DS}}(n,f), \mathbf{p}_2(n)]).\end{aligned} \tag{4.27}$$

Since this mask applies to $(1-\widehat{\mathcal{M}}_1)\times\mathbf{x}$, the equivalent mask to be applied to the original signal $\mathbf{x}$ is $\widehat{\mathcal{M}}_2 = \overline{M}_2 \times (1-\widehat{\mathcal{M}}_1)$.

4.3.3 Estimating the following sources

The proposed method can in theory estimate DOAs and masks for any number J' of sources using a source counting method (Kinoshita et al., 2018). In the following, we assume $J' = 2$. For the last source, since the remaining signal will contain a single speaker only, the network $\text{DOA\_DNN}_{J'}$ is trained using a normal cross-entropy loss.

4.4 Dataset

To evaluate the proposed speech extraction and separation methods, we introduce a multichannel , reverberated, noisy version of the WSJ0-2mix dataset which we refer to as Kinect-WSJ. The original WSJ0-2mix dataset introduced by Hershey et al. (2016) was created by mixing pairs of speakers from the WSJ corpus, and contains 20 k, 5 k, and 3 k training, development, and test mixtures, respectively. Each mixture contains two different speakers speaking for a variable duration. In this work, the "max" version of the dataset is used where the length of mixed signals is the maximum of the length of the individual signals. This version addresses some of the issues raised by Menne et al. (2019b) regarding the "min" version of the dataset. In particular, real scenarios rarely feature 100% overlap as in the "min" version but contain segments where a single speaker is speaking with short overlaps due to backchannel. This phenomenon can be observed in the CHiME-5 dataset which contains speech recorded in dinner parties. The proposed dataset simulated using the "max" version of WSJ0-2mix ensures that there are enough speech segments where a single speaker is speaking, but no effort is made to simulate realistic backchannel interference.

In our experiments we emulate the recording conditions of the CHiME-5 corpus which was recorded using Microsoft Kinect devices shown in Fig. 4.5. For each pair of speech signals in WSJ0-2mix, we simulate room impulse responses (RIRs) using RIR Simulator (Habets, 2018) for two distinct spatial locations with a minimum DOA difference of 5°. The room dimensions and the reverberation time (RT60) are randomly chosen in the

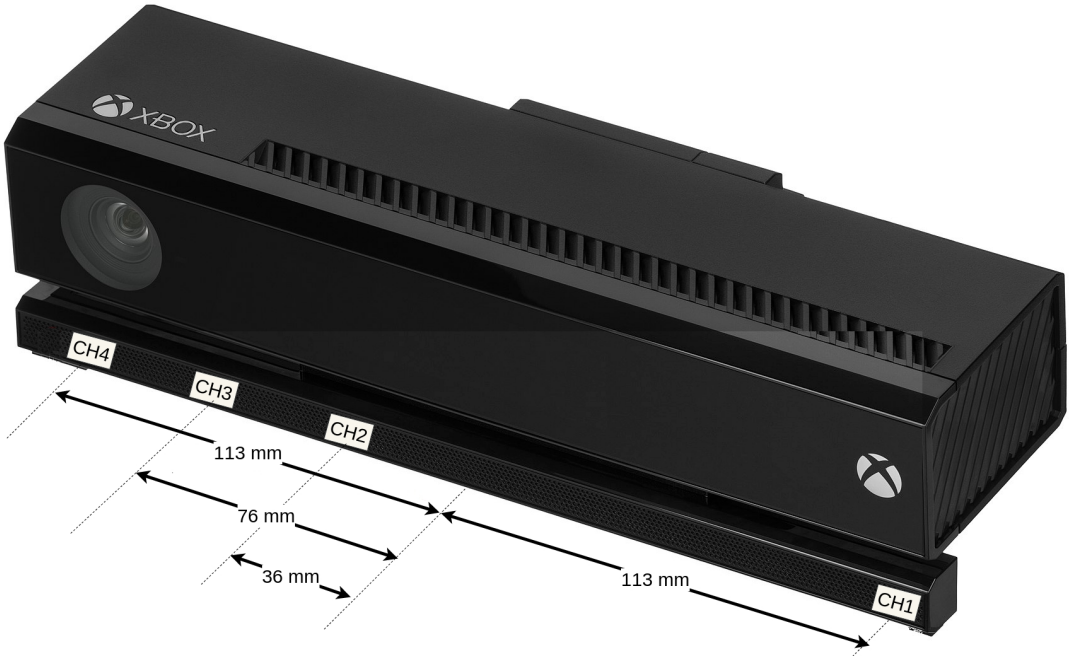


Figure 4.5: 4-channel microphone array in Microsoft Kinect. Original image credit: Wikipedia https://en.wikipedia.org/wiki/Kinect. Note that the channel indexing is the reverse of what is used in the CHiME-5 dataset.

range of $[3-9]$ m and $[0.3-1]$ s. The two speech signals are convolved with these RIRs and mixed at a random signal-to-interference ratio (SIR) in the range of $[0-10]$ dB. Real multichannel noise captured as part of the CHIME-5 dataset is then added with a random SNR in the range of $[0-10]$ dB. To obtain noise segments, the ground truth speech activity detection (SAD) labels from Track 3 of the DIHARD-II speaker diarization challenge (Ryant et al., 2019) are used, as these are more reliable than the SAD labels originally provided in CHiME-5. The noise signals in the training, development, and test sets are taken from different CHiME-5 sessions. The noise is realistic and non-stationary in nature and makes the speech separation task very challenging. A reverberated dataset based on WSJ0-2mix was created earlier by Wang et al. (2018d) but it does not contain any noise. Drude et al. (2019) also simulated a reverberated version of WSJ0-2mix with microphone noise but no ambient noise. The proposed simulated dataset contains 30, 10 and 5 h of data for training, development and test, respectively. The speech and noise signals used in training, development and test sets come from different speakers and different rooms.

4.5 Experimental setup

In this section we describe the setup of all the experiments done in this chapter. Section 4.5.1 describes the setup for speech extraction and Section 4.5.2 for speech separation. Section 4.5.3 describes the parameters used to train Conv-TasNet (Luo and Mesgarani, 2019), the state-of-the-art in speech separation, for comparison. Section 4.5.4 describes the evaluation metric.

4.5.1 Speech extraction using location information

Features to estimate the mask for speech separation Speech extraction and separation in this work is done in the STFT domain. The STFT was computed using 50 ms windows with 25 ms shift, resulting in 801 frequency bins. The 4 microphone channels in the dataset result in 6 microphone pairs. Since the CSIPD features consist of the sines and cosines of the phase differences between all pairs of microphones, 12 CSIPD feature vectors, each of dimension 801 were obtained for every time frame.

DNN to estimate the mask: The input to the mask estimation network in Fig. 4.1 was of dimension $3 \times 801 = 2403$: it comprises the magnitude spectrum of the DS signal, as well as the cosine and sine of the phase differences as detailed in Section 4.2.2, each of which is of dimension 801. A two-layer Bi-LSTM network containing 801 hidden units was trained to estimate the mask corresponding to the reverberated component of the localized speaker. Adam was used as the optimizer and MSE was used as cost function.

Estimating the location using GCC-PHAT: Experiments were conducted using both ground truth DOA values and DOA values estimated by GCC-PHAT for comparison. In the case of GCC-PHAT, peaks in the angular spectrum are assumed to correspond to the DOAs of the sources. The top two peaks are chosen and the peak which is closest to the true DOA is taken as the estimated DOA. Since GCC-PHAT works using 2 microphones, only the first and the last microphone of the array which are placed at a distance of 0.226 m are used.

4.5.2 Speech separation using location information

DOA estimation networks for SLOGD: The DOA estimation networks for the two sources contain a 2D convolutional neural network (CNN) which takes the 12 CSIPD channels as inputs and throws out a single-channel output using a 5×5 filter. This is followed by a rectified linear unit (ReLU) nonlinearity, a dropout layer, and a max pooling layer of kernel size 2×1 along the frequency dimension to obtain a phase embedding. Similarly, another 2D CNN is used to obtain a magnitude embedding with $|x_1(n, f)|$ as input. The magnitude and phase embeddings are concatenated and fed to a Bi-LSTM layer followed by a linear layer and a sigmoid nonlinearity. The DOA space is divided into $\mathrm{P} = 181$ discrete classes where each class corresponds to a DOA interval of 1°. Note that, since the array geometry is linear, the TDOAs corresponding to θ and $360 - \theta$ are

equal. Together with the non-speech class, this results in $181 + 1$ output classes with corresponding angles in the range $[0, 180]$. After source removal, only a single active speaker remains, hence we used cross-entropy as the cost function for DOA_DNN$_2$. VAD labels obtained from ASR alignments are used as targets while training the DOA estimation networks.

Mask estimation networks: The mask estimation network architecture for SLOGD used in Steps 2 and Steps 5 is similar to the network described in Section 4.5.1: a two-layer Bi-LSTM. The input dimension of the network is larger by 181, compared to the speech extraction mask estimation network, to accommodate the DOA information coming from the DOA estimation network as described in Section 4.5.1. The true reverberated mask corresponding to the localized speaker was used as the training target, with mean square error (MSE) as the cost function.
We trained the four networks one after another in the following order: DOA_DNN$_1$ $\Rightarrow$ MASK_DNN$_1$ $\Rightarrow$ DOA_DNN$_2$ $\Rightarrow$ MASK_DNN$_2$. We also tried to train the DOA and mask networks jointly which yielded poor results. This is because of the softmax nonlinearity between the DOA and mask estimation networks which prevents the gradients from the mask network from updating the weights of the DOA network. All networks were trained using Adam as the optimizer. After obtaining the masks, the rank-1 constrained MWF beamformer is used to separate speech from the mixture.

4.5.3 Conv-TasNet

For comparison, we trained the state-of-the-art Conv-TasNet (Luo and Mesgarani, 2019)[1] separation method on our dataset. It is trained on 4 s long sentences as reported in the original work. The parameters of the network are shown in Table 4.1. The converged network (after 147 epochs) was used to evaluate the source separation performance.

Table 4.1: Parameters used to train Conv-TasNet. The symbols shown in the Table are as reported by Luo and Mesgarani (2019).

Description of the parameters	Symbol	Value
Encoder filter count	N	512
Filter length	L	16
Bottleneck channel count	B	128
Number of channels in convolution block	H	512
Kernel size	P	3
Number of convolutional blocks per repeat	X	8

[1] https://github.com/kaituoxu/Conv-TasNet

4.5.4 Evaluation metric

The proposed speech extraction and separation techniques are evaluated using ASR metrics. The ASR system was trained on the enhanced training set using accurate senone alignments obtained from the underlying clean single-speaker utterances. The acoustic model (AM) was a 15-layer time-delayed neural network (TDNN) trained using the lattice-free maximum mutual information criterion (Povey et al., 2016). 40 dimensional Mel frequency cepstral coefficients along with 100-dimensional i-vectors were used as input features. All experiments are conducted under matched training and test conditions. No dereverberation was performed. A 3-gram language model was used while decoding.

4.6 Results

4.6.1 Speech extraction using true location information

Baseline results: Table 4.2 shows the baseline ASR performance before separation. A WER of 12.5% was obtained using speech containing a single speaker with reverberation. The WER increased from 12.5% to 25.5% with the addition of noise — a relative degradation of 104%. A relative drop of 160% was further observed when the mixture contained overlapping speech in addition to noise and reverberation. This shows that background noise and overlapping speech have a huge impact on the ASR performance and specific methods are required to deal with these distortions.

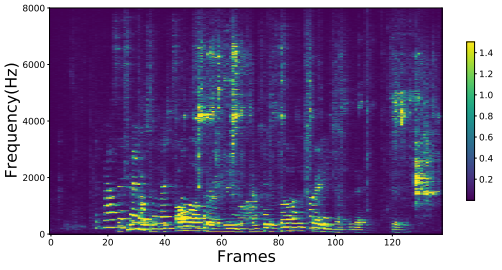
Table 4.2: Baseline WER (%) achieved on single-speaker or two-speaker mixtures before enhancement/separation. All results reported are with reverberated speech.

Input	**Single speaker**	**Single speaker + noise**	**2 speakers + noise**
WER	12.5	25.5	66.5

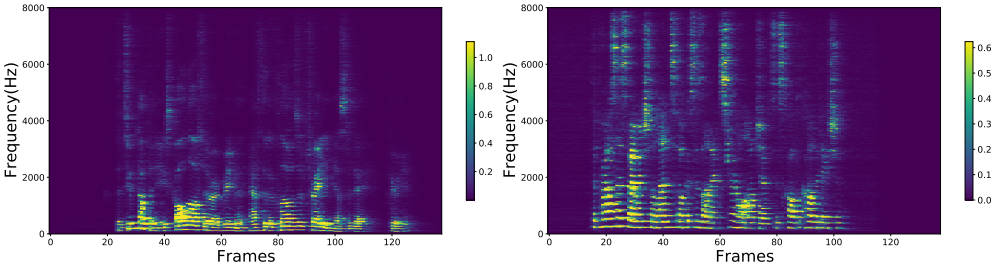
Results after speech extraction: Table 4.3 shows the ASR results obtained on noisy two-speaker mixtures after speech separation with different beamformers. A WER of 35.1% was obtained using the ground truth DOA with the R1-MWF beamformer, a relative improvement of 47.2% with respect to the system without source separation. This is relatively close to the ASR performance for a single speaker with noise (25.5%) as shown in Table 4.2. In all our experiments, the R1-MWF beamformer outperformed the widely used GEV beamformer. The SDW-MWF beamformer gave comparable performance to the R1-MWF.

Table 4.3: WER (%) achieved on two-speaker + noise mixtures after speech separation using true DOAs with different beamformers.

GEV	**R1-MWF**	**SDW**
36.5	35.1	35.5

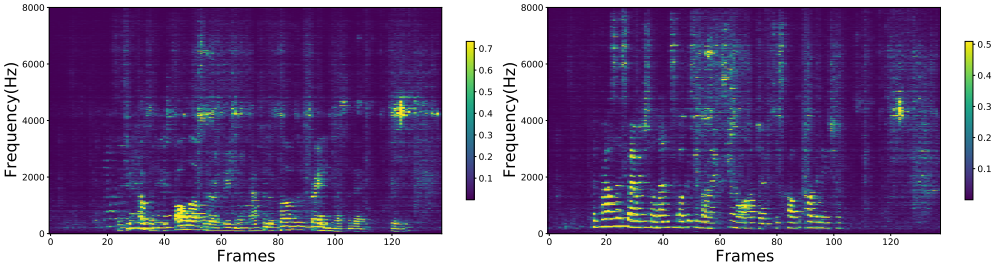


(a) Spectrogram of a two-speaker mixture.



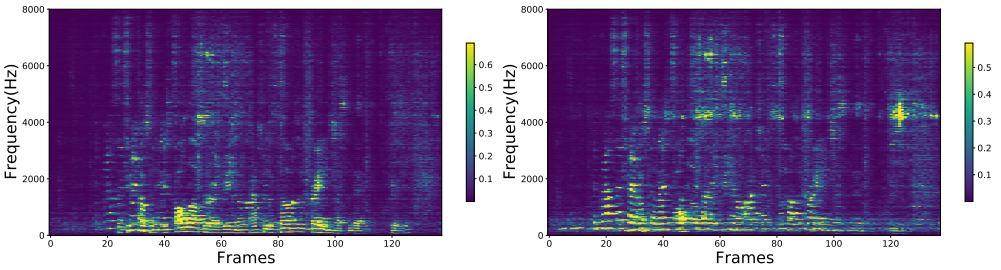
(b) Ground truth spectrum of c_1.

(c) Ground truth spectrum of c_2.



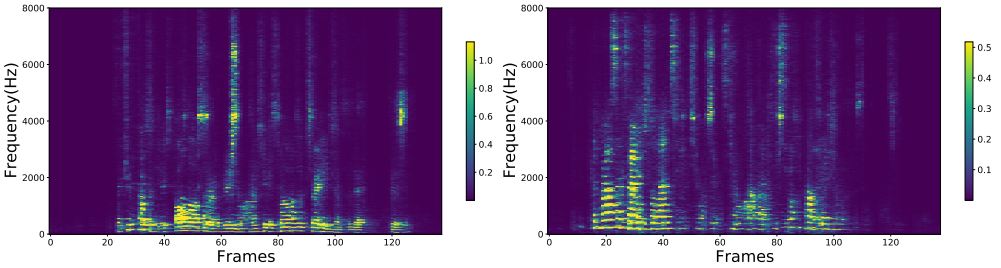
(d) Estimated spectrum of $\widehat{c}_1$.

(e) Estimated spectrum of $\widehat{c}_2$.



(f) Estimated spectrum of $\widehat{c}_1$ using SLOGD.

(g) Estimated spectrum of $\widehat{c}_2$ using SLOGD.



(h) Estimated spectrum of $\widehat{c}_1$ using Conv-TasNet.

(i) Estimated spectrum of $\widehat{c}_2$ using Conv-TasNet.

Figure 4.6: Speech separation with the R1-MWF using true speaker location information.

Figure 4.6a shows the spectrogram of a signal containing 2 speakers. The spectrum of the separated speech in Fig. 4.6d and Fig. 4.6e can be observed to match the spectrum of the respective speakers in Fig. 4.6b and Fig. 4.6c while ignoring the bins corresponding to the interfering speakers.

Impact of spatial distance on speech separation: The impact of the SIR and the DOA difference between the speakers on the ASR performance is shown in Table 4.4. Better performance is consistently observed for signals with higher SIR and also for mixtures containing speakers who are well separated in space ($> 50°$). It is interesting to note that better performance can be obtained for mixtures with a relatively lower SIR if the speakers are well separated in space compared to mixtures containing speakers who are spatially close to each other. For example, a WER of 21.7% was obtained for speech mixtures with SIR values in the range of $[0 - 5]$ dB and a DOA difference $> 50°$, that is a relative improvement of 17% compared to mixtures with a SIR > 5 dB and a DOA difference $< 10°$.

Table 4.4: WER (%) achieved on two-speaker + noise mixtures after separation using true DOAs as a function of the SIR and the DOA difference between the two speakers.

DOA diff vs SIR	< -5 **dB**	$[-5 : 0]$ **dB**	$[0 : 5]$ **dB**	> 5 **dB**
$< 10°$	67.0	43.2	25.7	26.3
$[10 : 30]°$	58.3	32.6	24.7	20.5
$[30 : 50]°$	60.0	32.0	23.4	22.2
$> 50°$	56.6	29.2	21.7	19.4

Impact of localization errors on speech extraction: As shown in Table 4.5, the performance dropped to 54.2% when the DOA was estimated using GCC-PHAT (compared to 35.1% when using the true DOA), indicating that erroneous DOA estimates decrease the separation quality. A similar performance degradation was observed when DOA errors were artificially induced. With $> 10°$ error in the speaker location, the performance was worse than baseline results without separation. This is probably because of enhancing noise or the interfering speaker in the direction corresponding to the wrong DOA.

Table 4.5: WER (%) achieved on two-speaker + noise mixtures after separation by inducing artificial DOA errors or due to the DOA estimation errors by GCC-PHAT. The models corresponding to the true DOAs are used. Matched condition models further deteriorated the performance.

Error (°)	WER (%)
GCC-PHAT	54.2
5	55.9
10	73.6
15	75.6

4.6.2 Speech separation results using SLOGD

ASR performance Table 4.6 shows the ASR performance after speech separation using SLOGD and Conv-TasNet. We obtain a WER of 44.2 % using our proposed method, that is a 34% relative improvement over the baseline without separation and a 18% relative improvement over GCC-PHAT based DOA estimation. In comparison, Conv-TasNet gave a WER of 53.2% on our dataset. These results are illustrated in Fig. 4.6.

Table 4.6: WER(%) results after speech separation using SLOGD and Conv-Tasnet.

Method	WER
SLOGD	44.2
Conv-TasNet	53.2

Analysis of the SLOGD network The outputs of two localization networks after pooling across time $\sum_n \mathrm{p}_1(n,\theta)$ and $\sum_n \mathrm{p}_2(n,\theta)$ are shown in Fig. 4.7. The blue curve shows the outputs of DOA_DNN$_1$ and the orange curve those of DOA_DNN$_2$. The true DOAs of the speakers are 55° and 25°, respectively. A peak corresponding to speaker c_1 can be observed at 52° in the blue curve. A peak at 25° can also be observed. The orange curve shows the output of DOA_DNN$_2$ after removing $\widehat{c}_1$ from the mixture $\mathbf{x}$. The peak at 52° has disappeared and a peak at around 30° can be seen, demonstrating the usefulness

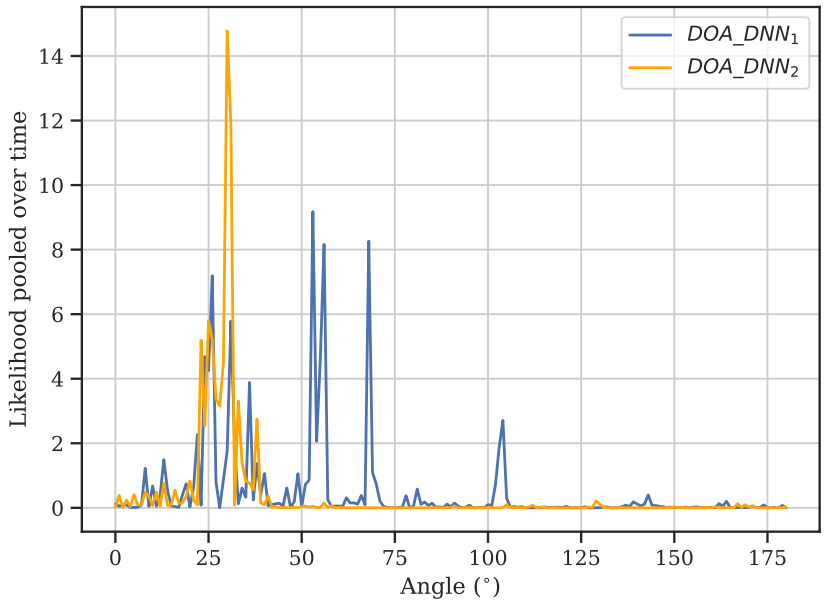
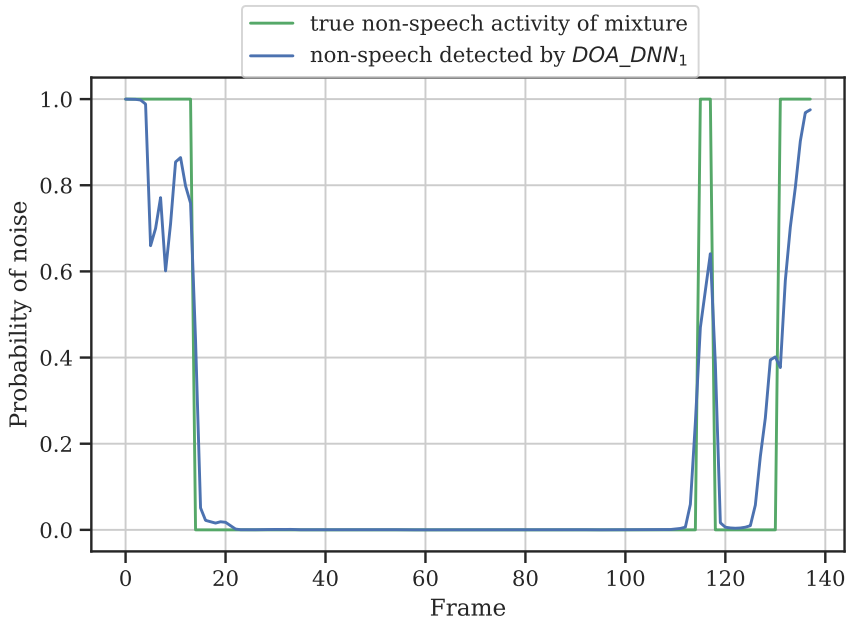


Figure 4.7: DOA estimates pooled across time for DOA_DNN$_1$ and DOA_DNN$_2$ on a given mixture. The true speaker DOAs are 55° and 25°.

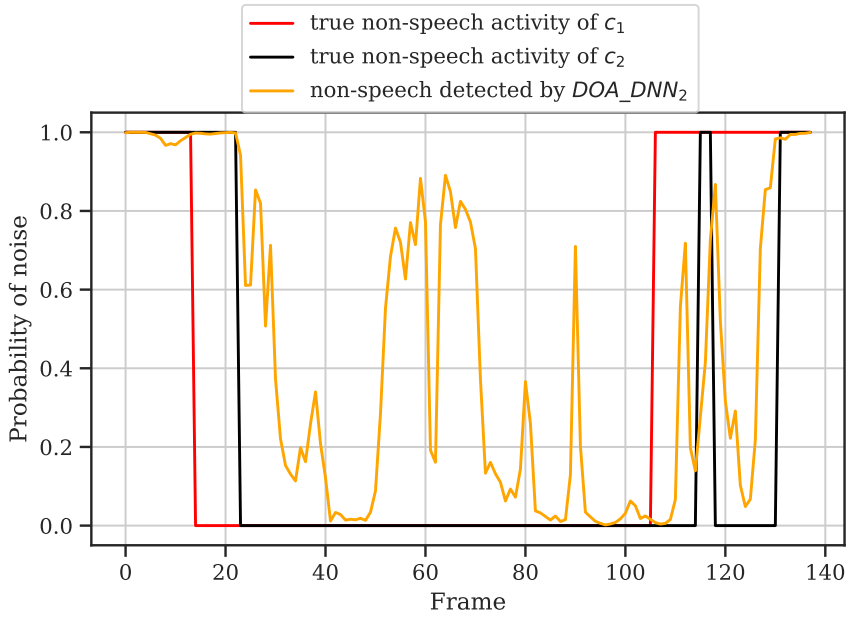
of source removal for estimating $\widehat{\theta}_2$.
Figure 4.8 shows the non-speech activity probability output of the DOA networks over-time. The blue curve in Fig. 4.8a for DOA_DNN$_1$ faithfully follows the curve corresponding to the ground truth non-speech labels of the mixture. Fig. 4.8b shows the same for DOA_DNN$_2$. The red and black curves show the ground truth labels for c_1 and c_2, respectively. The orange curve represents the noise activity probabilities obtained using DOA_DNN$_2$ after removing $\widehat{c}_1$ from the mixture $\mathbf{x}$. Multiple parts of the speech segment corresponding to the removed speech are marked as noise, further showing the effectiveness of the source removal step in Section 4.3.2.
On analyzing the errors made by the proposed SLOGD approach, we found that, for around 7% of the test dataset, the masks estimated in the two iterations were both closer (in terms of MSE with respect to the true masks) to the same speaker rather than two distinct speakers as expected. This suggests potential for further improvement either in the first mask estimation step or in the source removal stage.

4.7 Conclusion

In this chapter we approached the problem of distant speech extraction and separation in challenging noise and reverberation conditions. We conducted the first analysis of the impact of speaker localization accuracy on speech extraction performance as measured by the resulting ASR performance. To do so, we created a new dataset by reverberating WSJ0-2mix and mixing it with real CHiME-5 noise, and made the corresponding code publicly available. We found that the ASR performance strongly depends on the SIR of the speakers, with lower WERs for signals with higher SIR. The angular distance between the DOAs of the speakers was also found to have an impact, with better WERs for signals whose speakers exhibit a larger difference in DOAs.
Further, we proposed a deflation-based strategy for localization-guided speech separation. For each iteration, we train a network to estimate the location of the speaker and use it to estimate a time-frequency mask corresponding to the speaker. The estimated mask is used along with a rank-1 constrained MWF to extract the signal. The estimated source is then removed by masking the signal features before extracting the next source. Using this approach, we obtain a WER of 44.2 % compared to the WER of 53.2 % obtained by Conv-TasNet. Although the proposed method gave large improvements, the problem remains very difficult.



(a) DOA_DNN$_1$.



(b) DOA_DNN$_2$, after removal of $\widehat{c}_1$.

Figure 4.8: Noise activity detection by the DOA networks.

5 Explaining speech enhancement deep learning models

5.1 Introduction

Speech enhancement models are often trained in a supervised fashion using simulated data. The simulated data is generated by mixing speech and noise at different signal-to-noise ratios (SNR) and the DNN is trained to either estimate the speech and noise spectra or a time-frequency mask. Different kinds of noises have been used to simulate noisy speech. In the CHIME-4 evaluation challenge (Barker et al., 2017) for example, real noises recorded using 6 different microphones were used to simulate noisy speech. Pandey and Wang (2019) used a commercially available sound-effects library. Synthetically generated noise such as white or pink noise is also often used (Abdulaziz and Kepuska, 2017). This raises the question of what kind of noise is best suited to train the network. Real noise matching the conditions in which the speech enhancement model is to be deployed is a good choice. Yet recording noise scenes that cover all these conditions is expensive and often infeasible. An alternative is to use synthetically generated noise, provided that the impact on the enhancement performance is not drastic.
In this chapter we show that a speech enhancement model trained with synthetically generated speech-shaped noise (SSN) greatly improves the ASR performance on the CHiME-4 dataset but slightly less than a speech enhancement model trained with matched CHiME-4 noise. We focus on explaining this result, as a first step towards predicting the generalization ability of speech enhancement models and choosing optimal training noises in the future. To do so, we use a feature attribution method to quantify the importance of each input time-frequency bin in the estimation of the output mask. There are multiple feature attribution methods as detailed in Chapter 2. Many of these methods, such as deconvolution networks and grad-cam (Selvaraju et al., 2017), are designed for a particular DNN architecture, such as a CNN. Others such as DeepLIFT are designed for a wider range of architectures. DeepSHAP combines ideas from SHAP and DeepLIFT and we use it in this work to explain the performance of speech enhancement models. Existing studies on the explanation of neural network models for speech processing have focused on classification tasks (see Section 2.4.3). To the best of our knowledge, this is the first study on feature attribution for speech enhancement — a sequence-to-sequence regression task.
The rest of the chapter is organized as follows. Section 5.2 provides an overview of DeepSHAP. Section 5.3 describes the application of DeepSHAP to speech enhancement models. Section 5.4 proposes an objective measure to evaluate the obtained feature attribution values. Section 5.5 details our experimental setup and Section 5.6 discusses

the obtained results. We conclude in Section 5.7.

5.2 DeepSHAP

DeepSHAP (Lundberg and Lee, 2017) combines the ideas of DeepLIFT and SHAP. Following the notation in Section 2.4, we temporarily denote the input feature vector as

$$\mathbf{x} = [x_1, \ldots, x_D] \tag{5.1}$$

where D is the number of features, and we assume that the output $\mathcal{F}(\mathbf{x})$ is scalar.
SHAP computes the relevance of a particular feature x_d by observing the change in the output with respect to the presence vs. absence of that feature. To avoid retraining the network for every combination of present vs. absent features, the absence of a feature is approximated by replacing it by its expected value. This can be represented as locally projecting the input features $\mathbf{x}$ into a simplified input vector $\mathbf{x}' = \{x'_1, \ldots . x'_D\}$ of same dimension D using a function $h_{\mathbf{x}}(.)$ such that

$$[h_{\mathbf{x}}(\mathbf{x}')]_d = \begin{cases} x_d & \text{if } x'_d = 1 \\ \mathbb{E}(x_d) & \text{if } x'_d = 0 \end{cases} \tag{5.2}$$

where each simplified input $x'_d \in \{0, 1\}$ denotes the presence or absence of the corresponding feature. SHAP approximates the network output as a linear combination of the simplified inputs:

$$\mathcal{F}(h_{\mathbf{x}}(\mathbf{x}')) \approx \phi_0 + \sum_{d=1}^{D} \phi_d \, x'_d \,. \tag{5.3}$$

Each weight ϕ_d is referred to as a SHAP value and it directly quantifies the relevance of the corresponding feature.
Let us now recall the principle of DeepLIFT, where the feature attributions $C_{\Delta x_d \Delta \mathcal{F}}$ sum up to the difference between the output $\mathcal{F}(\mathbf{x})$ and the output $\mathcal{F}(\mathbf{x}^{\mathrm{r}})$ for a reference input vector $\mathbf{x}^{\mathrm{r}} = [x^{\mathrm{r}}_1, \ldots, x^{\mathrm{r}}_D]$:

$$\sum_{d=1}^{D} C_{\Delta x_d \Delta \mathcal{F}} = \Delta \mathcal{F} = \mathcal{F}(h_{\mathbf{x}}(\mathbf{x}')) - \mathcal{F}(\mathbf{x}^{\mathrm{r}}) \tag{5.4}$$

where

$$\Delta x_d = x_d - x^{\mathrm{r}}_d. \tag{5.5}$$

From Eqs. (5.3) and (5.4), by setting $\mathbf{x}^{\mathrm{r}} = \mathbb{E}(\mathbf{x}) = h_{\mathbf{x}}(\mathbf{0})$ and $\phi_0 = \mathcal{F}(\mathbf{x}^{\mathrm{r}})$, we obtain

$$C_{\Delta x_d \Delta \mathcal{F}} = \phi_d \,. \tag{5.6}$$

Therefore SHAP is conceptually equivalent to DeepLIFT when $\mathbf{x}^{\mathrm{r}} = \mathbb{E}(\mathbf{x})$ (Lundberg and Lee, 2017). The differences are that, while DeepLIFT's backpropagation rules presented in Section 2.4.2.3 are heuristic and the choice of the reference value is up to the

user, SHAP values are the only attribution values that satisfy the desirable consistency property stated in Section 2.4.2.4.
In practice, the replacement of every absent feature by its expected value is a poor approximation when applied to the whole network. DeepSHAP combine efficient, analytical computation of SHAP values for simple network modules (linear, maxout, activation) with DeepLIFT's mutiplier composition rule to backpropagate these attribution values down to the input layer.

5.3 Computing SHAP values for speech enhancement models

In this Section, we revert back to the classification notations used in the thesis, i.e., the multichannel signal is defined as

$$\mathbf{x}(n,f) = [x_1(n,f), \ldots, x_I(n,f)]^T, \tag{5.7}$$

where I is the number of microphones. Since the focus of this chapter is on speech enhancement there is a single source $J' = 1$. The mixture can be expressed as

$$\mathbf{x}(n,f) = \mathbf{c}_1(n,f) + \mathbf{u}(n,f). \tag{5.8}$$

A DNN is trained to estimate the IRM $\mathcal{M}$ using a single-channel input magnitude STFT — say channel 1, i.e.,

$$\widehat{\mathcal{M}} = \mathcal{F}(|\mathbf{X}_1|). \tag{5.9}$$

The most natural way of using DeepSHAP is to assume that each input time-frequency bin $|x_1(n',f')|$ is a feature and to compute the contribution of that feature to every time-frequency bin of the mask $\widehat{\mathcal{M}}(n,f)$. We therefore obtain $N \times F$ relevance matrices $\mathbf{\Phi}^{\mathrm{TF}}(n,f)$ of size $N \times F$ each, which we refer to as time-frequency SHAP. In order to reduce the number of matrices to be computed and analyzed, an alternative is to sum the attribution values per frame as

$$\mathbf{\Phi}^{\mathrm{T}}(n) = \sum_f \mathbf{\Phi}^{\mathrm{TF}}(n,f). \tag{5.10}$$

This doesn't require the computation of every $\mathbf{\Phi}^{\mathrm{TF}}(n,f)$. Instead, the SHAP values are summed at the output layer, and a single backpropagation to the inputs is performed. The F resulting matrices $\mathbf{\Phi}^{\mathrm{T}}(n)$ — referred to as time SHAP — are also $N \times F$ matrices, showing the relevance for every time frame of the output mask. Similarly, attribution values can also be summed over the whole utterance as

$$\mathbf{\Phi}^{\mathrm{U}} = \sum_n \mathbf{\Phi}^{\mathrm{T}}(n). \tag{5.11}$$

We refer to $\mathbf{\Phi}^{\mathrm{U}}$ as utterance SHAP. It can be observed that

$$\sum_{n,f} \mathbf{\Phi}^{\mathrm{TF}}(n,f) = \sum_n \mathbf{\Phi}^{\mathrm{T}}(n) = \mathbf{\Phi}^{\mathrm{U}}\,. \tag{5.12}$$

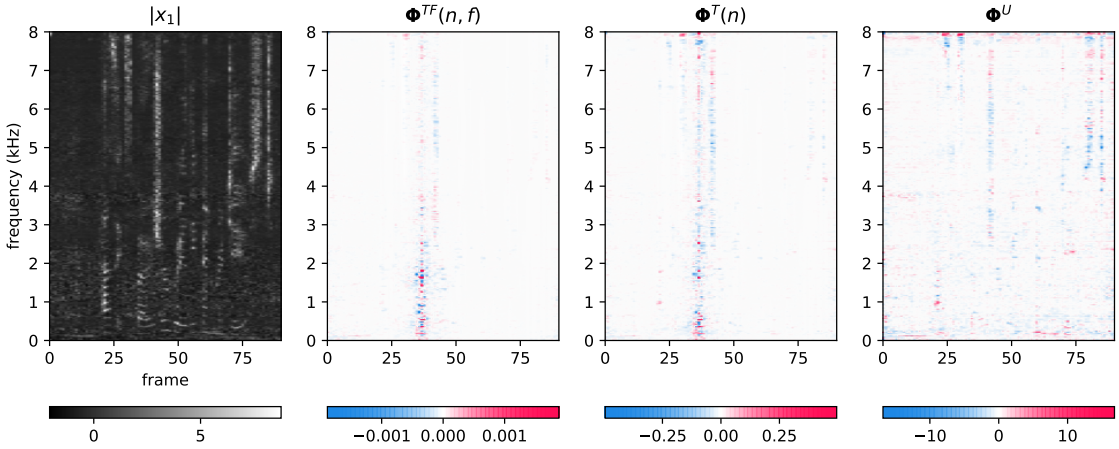


Figure 5.1: Example SHAP values computed for a noisy speech mixture with $n = 36$ and $f = 1635$ Hz. The input spectrogram has negative values since we use per-utterance spectral mean and variance normalization before feeding it to the network.

Figure 5.1 shows the obtained SHAP values for specific values of n and f. $\mathbf{\Phi}^{\mathrm{TF}}(n, f)$ gives the highest possible granularity of relevance while $\mathbf{\Phi}^{\mathrm{U}}$ is the lowest possible granularity. We have observed that the relevance maps $\mathbf{\Phi}^{\mathrm{TF}}(n, f)$ for different frequency bins f in a single frame n (not shown here) are similar to each other. We therefore choose $\mathbf{\Phi}^{\mathrm{T}}(n)$ for our following analysis.
As observed in the figure, SHAP attributions can either be positive or negative. The negative values also point to useful information. In this case, we observe that both the positive and negative values point to time-frequency bins belonging to speech and therefore use the absolute value of the SHAP attributions in the rest of this work. Negative values were also shown to be useful across multiple other feature attribution frameworks such as LRP and DeepLIFT.

5.4 Measure of SHAP relevance

Evaluating feature attributions is non-trivial. This is usually done using human visualization which is subjective by nature. In this section we propose an objective measure to evaluate the SHAP values obtained for speech enhancement models.
The job of a speech enhancement model is to remove the time-frequency bins associated with noise while retaining the time-frequency bins associated with speech. We argue that a well-trained speech enhancement model, with good generalization capability, should mostly look at time-frequency bins belonging to speech. This is particularly true while evaluating the model in unseen noise conditions, where a speech enhancement model will only have access to speech patterns learned from the training dataset with no prior knowledge about the noise spectra.
Based on this assumption we propose to use the following measure, which we refer to as

the speech relevance score (η), to summarize the estimated SHAP values:

$$\eta = \frac{\sum_{n \in \text{speech}} \#\{\phi_{>T+\text{IBM}}(n)\}}{\sum_{n \in \text{speech}} \#\{\phi_{>T}(n)\}}, \tag{5.13}$$

where $\#\{\phi_{>T}(n)\}$ represents the number of time-frequency bins in $\mathbf{\Phi}^{\text{T}}(n)$ whose absolute value is greater than a threshold T and $\#\{\phi_{>T+\text{IBM}}(n)\}$ represents the number of such bins identified as speech in an ideal binary mask. The measure is computed only for frames containing speech. The threshold T denotes the T-th percentile of the absolute SHAP values in each frame. The number of time-frequency bins with large enough SHAP value $\#\{\phi_{>T}(n)\}$, increases with decreasing T as shown in Fig. 5.2.

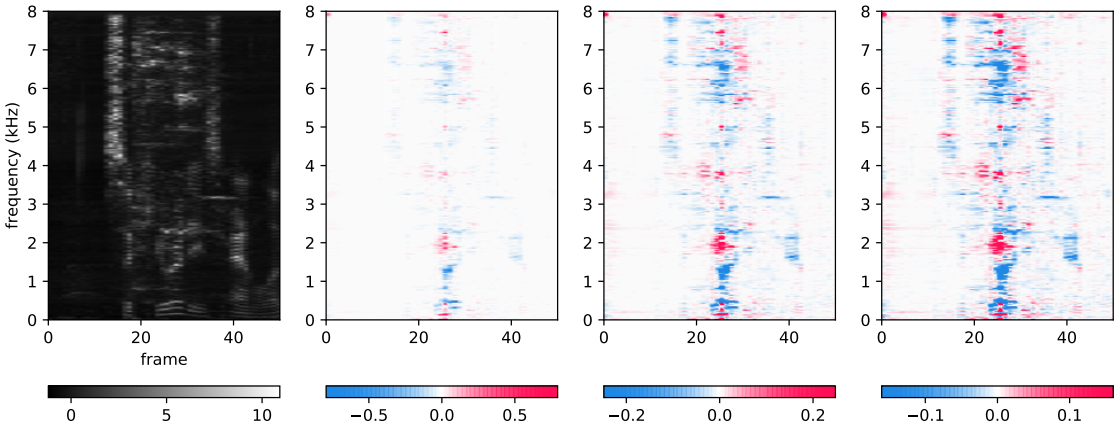


Figure 5.2: Input noisy speech spectrogram and SHAP values $\mathbf{\Phi}^{\text{T}}(n)$ for $n = 26$ plotted with three color scales associated with the thresholds $T = 99.9, 99.0$ and 98.0, respectively. The SHAP values above the threshold correspond to the reddest or the bluest color in each plot.

The speech relevance score only computes the proportion of selected SHAP bins which belong to speech, that is the true positive rate. We might also want to measure the false positive rate, that is the number of selected SHAP bins which do not belong to speech, which we refer to as the speech irrelevance score. In the case of speech enhancement where the mixture contains a single speaker plus noise, the speech irrelevance measure is simply $1 - \eta$. For other applications containing multiple sources, such as speech separation, the speech irrelevance score could be computed using Eq. (5.13) by replacing the IBM with the IBM of the interfering speaker and noise. Note that an F1 metric would be a better metric than η, but the computation of the false negative rate is non-trivial because the time-frequency bins in $\mathbf{\Phi}^{\text{T}}(n)$ whose absolute value is large tend to cluster around time frame n rather than spanning the whole time axis.

5.5 Experimental setup

5.5.1 Dataset

Experiments are conducted using the CHiME-4 dataset (Barker et al., 2017), which consists of Wall Street Journal corpus sentences spoken by talkers situated in challenging noisy environments recorded using a 6-channel tablet-based microphone array. The original dataset considers four different categories of environments: bus, cafe, pedestrian area, and street junction. It comes with a data simulation tool, which mixes original non-reverberated WSJ0 utterances with background noise, ensuring the same SNR distribution as real noisy recordings on every channel.
In order to train the speech enhancement network, we generate three different training and validation datasets corresponding to three different noise conditions, namely

1. CHiME: real noise recordings from the CHiME-4 dataset,
2. SSN: artificially generated noise (see Section 5.5.2),
3. Network: sound files from the Network Sound Effects library[1] as used by Pandey and Wang (2019), containing sounds from various categories such as music, weather, rail, etc.

Each condition involves $7,138$ utterances for training and $1,640$ for validation. The distribution of SNR values is the same across all conditions. Since the mask estimation network is trained using a single channel, only the first channel of each simulated mixture is used.

5.5.2 Generating speech-shaped noise

SSN is a form of synthetically generated noise which is generated by applying a "speech-like" filter to white noise. The speech-like filter is obtained by averaging the magnitude spectrum of several clean speech utterances as

$$\text{filter}(f) = \frac{1}{N_s} \sum_{k=1}^{N_s} \frac{1}{N_k} \sum_{n=1}^{N_k} |s(k, n, f)| \tag{5.14}$$

where N_s is the total number of speech utterances and N_k is the length of the k-th utterance.
In this work, for every mixture to be generated, we take $N_s = 6$ other clean speech utterances from the CHiME-4 corpus to create the filter. The magnitude response of one filter and the spectrogram of the resulting noise signal are shown in Figs. 5.3 and 5.4, respectively. The intuition behind the use of SSN is that it has high energy in time-frequency bins which are typically dominated by speech, thereby making it harder for the network to differentiate between speech and noise and resulting in a more accurate decision boundaries.

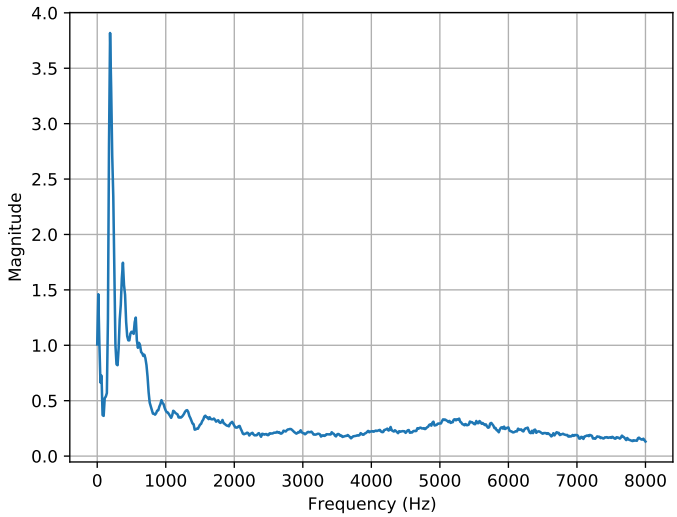


Figure 5.3: Example speech-like filter used to create SSN.

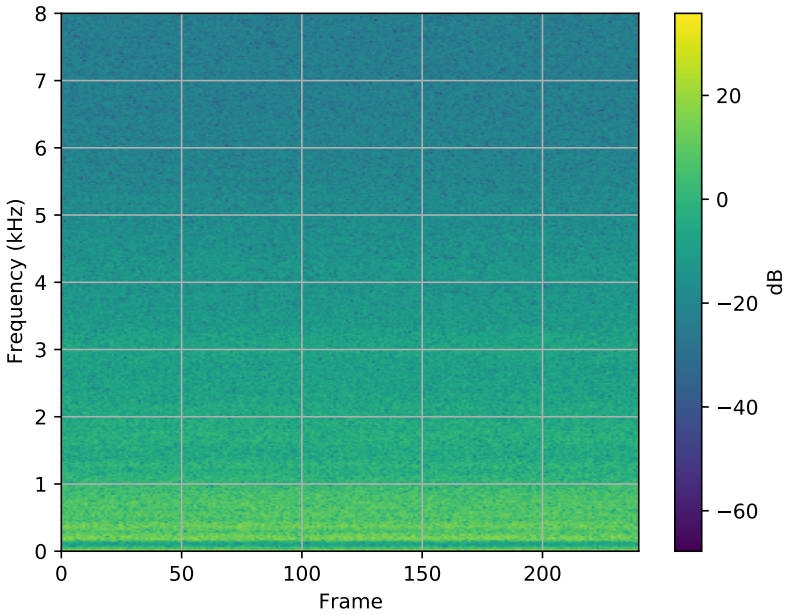
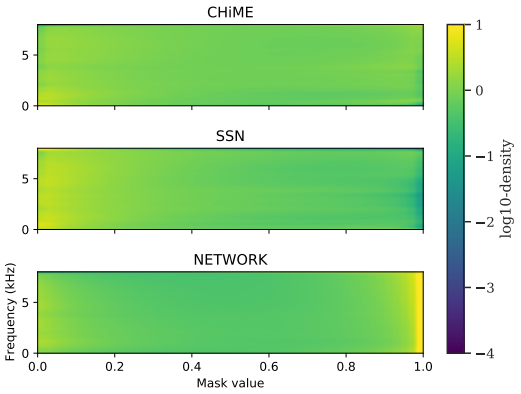


Figure 5.4: Example spectrogram of an SSN signal.

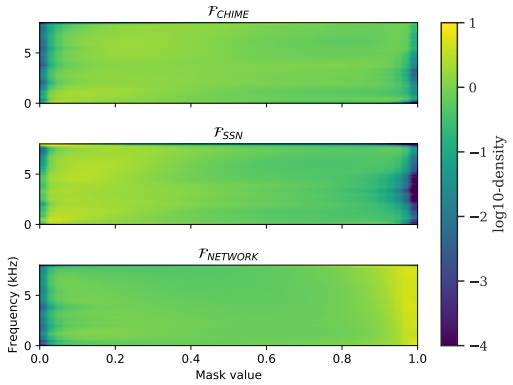
5.5.3 DNN architectures for speech enhancement

The DNNs for speech enhancement are trained to estimate the IRM using the STFT magnitude spectra of the mixture x_5 (i.e., channel 5 of the multichannel signal) as inputs.

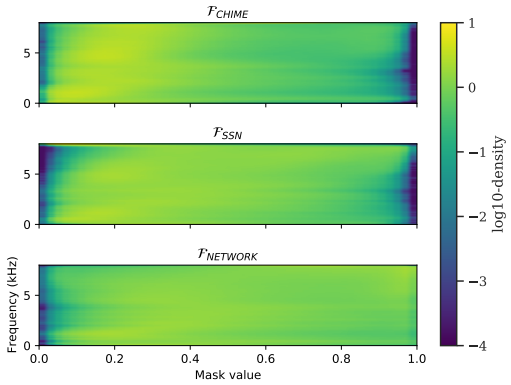
[1] https://www.sound-ideas.com/Product/199/Network-Sound-Effects-Library



(a) Histogram of the true IRM values on the simulated training data.



(b) Histogram of the mask values estimated by each model on the simulated training data.



(c) Histogram of the mask values estimated by each model on the real CHiME-4 evaluation data.

Figure 5.5: Histogram of the mask values per frequency bin.

The window size of the STFT was 50 ms with a 25 ms shift. The input dimension of the network was 401. The DNN architecture contained two Bi-LSTM layers followed by layer normalization (Ba et al., 2016) and a linear layer containing 401 hidden units. The output was constrained to lie in the range $[0-1]$ using a sigmoid nonlinearity. The speech enhancement models trained on the three different noise conditions are denoted as $\mathcal{F}_{\text{CHIME}}$, $\mathcal{F}_{\text{SSN}}$ and $\mathcal{F}_{\text{NETWORK}}$.

The histograms per frequency bin of the true IRM values, the mask values estimated by each network on its respective simulated training data and the mask values estimated by each network on the real evaluation data are shown in Fig. 5.5. The bias introduced by different noises can be observed in Fig. 5.5a. Network noise results in a large number of time-frequency bins with IRM values close to 1. This is due to the sparse nature of the noises, whose energy often concentrates in a small number of frames. By contrast, IRM values are more evenly distributed when using CHiME or SSN noise. This bias also reflects in the outputs of the trained models on the training set and to a lesser extent

on the evaluation set as seen in Fig. 5.5b and Fig. 5.5c, respectively. These histograms show that the DNNs trained with different noises tend to look at the input mixture x in different ways, which is noticeable when the same input mixtures are given to all the networks in Fig. 5.5c.

5.5.4 ASR evaluation

We evaluate the ASR performance resulting from speech enhancement on the real evaluation set (`et05_real`) of the original CHiME-4 dataset. Since the true IBM values are required to compute the speech relevance score, we also use simulated evaluation set (`dt05_simu`) in our experiments. Channel 5 of the recordings is used as input to the speech enhancement networks to estimate the mask. The estimated mask along with multichannel signals from all channels (with the exception of channel 2) is used to compute a rank-1 constrained multichannel Wiener filter (Wang et al., 2018b) which is then used to obtain the enhanced speech. The baseline ASR system provided as part of the CHiME-4 challenge was used to evaluate the quality of enhanced speech. This system follows the `nnet1` recipe of the Kaldi ASR toolkit (Povey et al., 2011), involving a 7-layer MLP-based acoustic model and a 3-gram language model. It was trained on both real and simulated noisy speech.

5.5.5 DeepSHAP

DeepSHAP requires the computation of the expectation of the input vector at every layer. Samples to compute these expectations are taken from the dataset associated with the input mixture $\mathbf{x}$ whose feature attribution is being computed. For example, if $\mathbf{x}$ is an utterance from `et05_real`, then a part of the `et05_real` dataset is kept aside as samples to compute these expectations. In our experiment, the sample size is set to 40. The SHAP toolkit[2] was used to compute the SHAP values. We use the DeepExplainer component of the toolkit[3], wherein the SHAP values are computed analytically for simple DNN modules (linear, sigmoid) or using the standard gradient for complex modules (Bi-LSTM) and backpropagated with DeepLIFT's mutiplier composition rule. A threshold value of $T = 99.9$ is used throughout our experiments unless mentioned otherwise. A high value of T results in lower number of time-frequency bins to compute η. But, since we are computing SHAP values per time frame, i.e., $\mathbf{\Phi}^T(n)$, the relevance is expected to be restricted to a small neighborhood around n.

5.6 Results

5.6.1 ASR

Table 5.1 shows the ASR results on `et05_real` dataset using different speech enhancement models. The same ASR model was used for all the experiments. A baseline WER

[2] https://github.com/slundberg/shap
[3] https://github.com/slundberg/shap/blob/master/shap/explainers/deep/deep_pytorch.py

Training Noise	ASR input	`et05_real` (%)	`dt05_simu`(%)
Baseline (No Enhancement)	x_5	25.9	12.7
CHiME	$\mathcal{F}_{\text{CHIME}}(x_5)$	11.7	6.7
SSN	$\mathcal{F}_{\text{SSN}}(x_5)$	14.0	7.3
Network	$\mathcal{F}_{\text{NETWORK}}(x_5)$	15.1	7.7

Table 5.1: WER (%) on the CHiME-4 real evaluation (`et05_real`) and simulated development data (`dt05_simu`).

Model	$T = 99.9$	$T = 99.0$	$T = 98.0$
$\mathcal{F}_{\text{CHIME}}$	94.8	92.2	90.5
$\mathcal{F}_{\text{SSN}}$	89.6	87.2	85.4
$\mathcal{F}_{\text{NETWORK}}$	90.3	89.5	88.7

Table 5.2: Speech relevance score (η) (%) values using different thresholds on `dt05_simu`.

of 25.9% was obtained when no speech enhancement was performed. The WER improved to 11.7% by enhancing speech using the $\mathcal{F}_{\text{CHIME}}$ model. The WERs obtained using the speech enhancement models trained with SSN and Network noise are 14.0% and 15.1%, respectively. The improved performance with the speech enhancement model trained using CHiME noise can be attributed to the matched condition between the training and evaluation data. Nevertheless, the results obtained using speech enhancement models trained with SSN and Network noise are significantly better than the baseline showing the usefulness of these noises for training a speech enhancement model. Similar gains in the ASR performance can be observed on the `dt05_simu` dataset.

5.6.2 Speech relevance score

Table 5.2 shows the speech relevance score obtained on `dt05_simu` for all the speech enhancement models with different threshold values. The results are obtained using a total of 300 utterances. A speech relevance score of 94.8% was obtained using $\mathcal{F}_{\text{CHIME}}$, meaning that for a threshold $T = 99.9$, 94.8% of the time-frequency bins in the input spectrogram which were used to explain the output mask were dominated by speech. The speech relevance score values in Table 5.2 follow the trends observed in the ASR results of Table 5.1. Better η values are seen for the $\mathcal{F}_{\text{CHIME}}$ model, which gave the best ASR performance. The negligible difference between the speech relevance score values for $\mathcal{F}_{\text{SSN}}$ and $\mathcal{F}_{\text{NETWORK}}$ reflects the difference in the ASR results of the corresponding models on `dt05_simu`, albeit in favor of $\mathcal{F}_{\text{NETWORK}}$. The speech relevance score varies with respect to the threshold T, indicating that the time-frequency bins with lower SHAP values are not dominated by speech. We can therefore conclude that $\mathcal{F}_{\text{CHIME}}$ works better than other models because it relies on speech-dominated time-frequency bins to estimate the mask.

Experiment setup	$\mathcal{F}_{\text{NETWORK}}$ (%)	$\mathcal{F}_{\text{SSN}}$ (%)
Train	82.5	81.7
Test	58.6	74.4

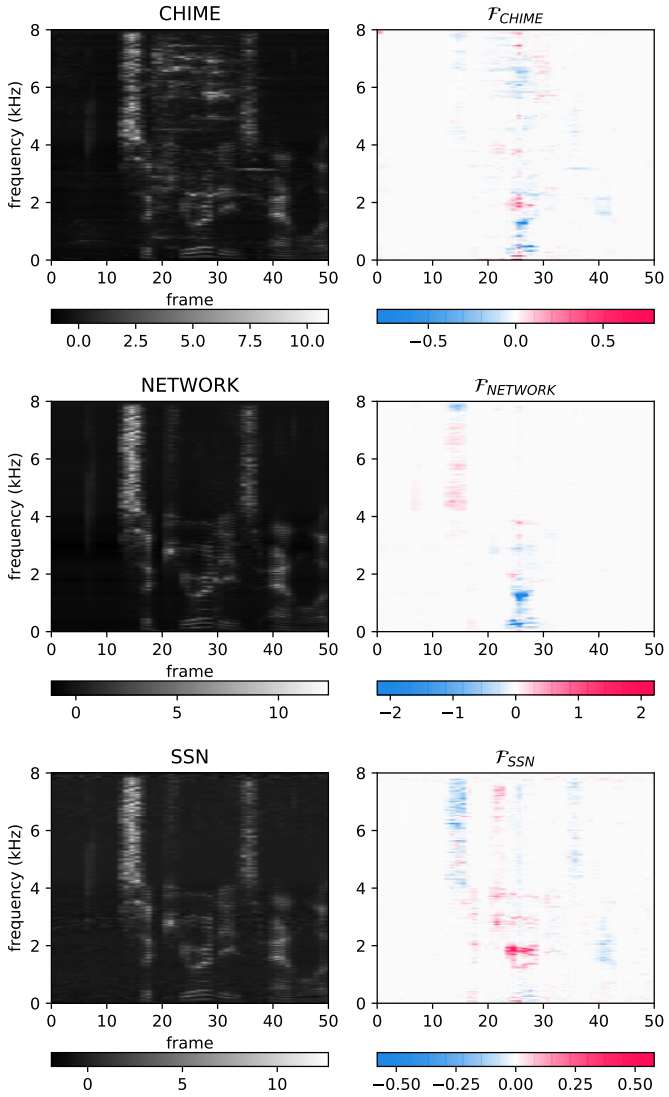
Table 5.3: Average speech relevance score obtained on 28 random utterances in the training and test setups. The speech relevance score for $\mathcal{F}_{\text{CHIME}}$ was 81.7%.

5.6.3 Generalization capability of speech enhancement models

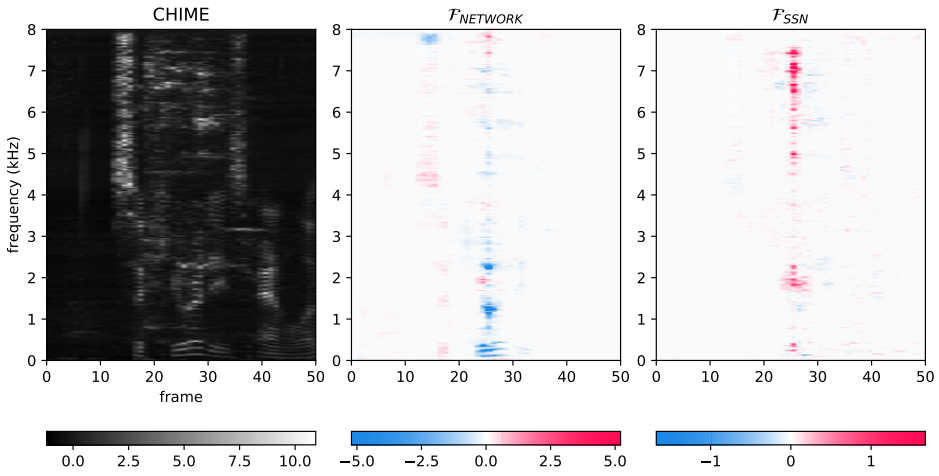
To better understand how the models generalize, we visualize SHAP values for several training and test signals sharing the same underlying clean speech signal. We refer to these two setups as training and test, respectively. In the training setup, the SHAP values are computed for each model by mixing the clean speech signal with noise from the same type as the one used to train the model. In the test setup, the SHAP values are computed for all models except $\mathcal{F}_{\text{CHIME}}$ by mixing the clean speech signal with CHiME noise. For a model that generalizes well, the relevant time-frequency bins should be the same across the training and test setups.
Figures 5.6a and 5.6b show the SHAP values obtained in the training and test setups, respectively. In this example, the SHAP values for $\mathcal{F}_{\text{NETWORK}}$ seems to match better across the two conditions when compared to $\mathcal{F}_{\text{SSN}}$, which can also be observed in the obtained speech relevance score values shown in the figure.
Table 5.3 shows the average speech relevance scores computed for the training and test setups on a larger number of utterances. It can be observed that the speech relevance score is higher in the training setup in $\mathcal{F}_{\text{NETWORK}}$ compared to $\mathcal{F}_{\text{SSN}}$ but is lower in the test setup indicating that $\mathcal{F}_{\text{SSN}}$ has better generalization capability than $\mathcal{F}_{\text{NETWORK}}$.

5.7 Conclusion

In this chapter we addressed the problem of explaining the predictions of a speech enhancement model. DeepSHAP, a feature attribution method, is employed to figure out which time-frequency bins of the input spectrogram are used by the DNN to estimate the mask. Based on the idea that a well-trained model should look at time-frequency bins dominated by speech instead of those dominated by noise, we proposed speech relevance score — a measure to evaluate feature attributions. We showed that speech enhancement models having a higher speech relevance score give better ASR performance. We also showed that the generalization capability of a speech enhancement model trained using synthetically generated SSN is better than that of a speech enhancement model trained using Network noise.



(a) $\mathbf{\Phi}^T(n)$ in the train setup for $\mathcal{F}_{\text{CHIME}}$, $\mathcal{F}_{\text{NETWORK}}$ and $\mathcal{F}_{\text{SSN}}$. The speech relevance score values were 70.2% , 95.2% and 90.3% , respectively.



(b) $\mathbf{\Phi}^T(n)$ in the test setup for $\mathcal{F}_{\text{NETWORK}}$ and $\mathcal{F}_{\text{SSN}}$. The speech relevance score values were 90.0% and 61.3%, respectively.

Figure 5.6: Input noisy speech spectrogram and SHAP values $\mathbf{\Phi}^{\text{T}}(n)$ for $n = 25$ in the training and test setups.

6 Conclusion and future research directions

The thesis was conducted in the context of the ANR Vocadom project. The methods proposed herewith can be used in commonly occurring problems across the speech industry. In this chapter, we conclude the thesis with a summary of our proposed methods in Section 6.1. Perspectives and future directions are presented in Section 6.2.

6.1 Summary

In Chapter 3, we addressed the problem of localizing the target speaker who uttered a known text such as the wake-up word of a voice assistant system in a reverberant, noisy, multispeaker acoustic environment. This is a new task compared to other works on multispeaker localization where all the speakers are localized instead. Our proposed method has applications in hands-free voice assistants which are activated by a known wake-up word — the desired text which can be used to localize the target speaker.
Our proposed method is done in two steps. In the first step a time-frequency mask corresponding to the target speaker — referred to as a target identifier — is estimated using the magnitude spectrum of the signal and a sequence of phoneme spectra corresponding to the phonemes of the words, with a DNN. To obtain the phoneme spectra, the speech mixture $\mathbf{x}$ is first forced-aligned with the text using an ASR model. A sequence of representative spectra corresponding to the sequence of phonemes obtained from ASR alignments is appended along with the input magnitude spectrum to estimate the mask. The representative spectrum corresponding to each phoneme is precomputed by taking the average of the magnitude spectra for that phoneme in the training dataset. In the second step, the estimated mask is used along with CSIPD features to estimate the DOA of the speaker using a DNN.
Experiments are conducted on both simulated and real data to verify the effectiveness of the proposed approach. Mixtures containing two speakers and noise are simulated by convolving two speech signals with simulated room impulse responses. Synthetic noise was included in the training set and real noise in the evaluation set to ensure that the model does not overfit to simulated noise. We recorded real data with two human speakers speaking from two different positions in a room. The positions of the speakers are set to ensure that the target speech signals are recorded at various SIRs. Three metrics are used to evaluate the system, namely, the gross error rate, interference closeness rate and mean average error. The objective was to obtain the lowest possible values for these metrics. When compared to the classical GCC-PHAT algorithm, our proposed method

gave a 71% relative reduction in the gross error rate, a 92% relative reduction in the interference closeness rate and an 87% reduction in the mean average error on simulated data. On real data, the proposed method gave a 39% relative reduction in the gross error rate over GCC-PHAT when the SIR between speakers was 0 dB. This shows that textual information is useful to improve speaker localization performance.

In Chapter 4, the problem of speech extraction and speech separation using location information is addressed. Localization guided speech extraction is done by first applying a delay-and-sum beamformer to the multichannel mixture using the location information. The magnitude spectrum and the phase difference of the beamformed signal with respect to a reference microphone are used to train a DNN to estimate a time-frequency mask. The estimated mask is used to extract speech from a mixture using a mask-dependent beamformer. We create a reverberated, noisy version of the WSJ0-2mix dataset containing two speakers and real noises extracted from the CHiME-5 dataset to test the performance of our proposed method. We obtain a WER of 35.0% on this dataset using the true speaker location. It was observed that the SIR of the mixture plays a bigger role compared to the angular distance between the speakers while separating speech and that a large angular distance (in the order of $> 50°$) can compensate for a low SIR. Further experiments with speaker locations estimated using GCC-PHAT, where we obtain a WER of 54.2%, showed that erroneous localization estimates impact the speech separation performance. We further propose a Speaker LOcalization Guided Deflation (SLOGD) approach for speech separation. SLOGD is an iterative approach which estimates a single source for every iteration until all the sources have been estimated. At each iteration we estimate the DOA of one speaker using the CSIPD features between all microphone pairs and the magnitude spectrum of the mixture. The DOA network estimates either the speaker location or outputs a non-speech label if no speaker is active. After estimating the mask corresponding to the localized speaker, the speaker is removed from the mixture by multiplying the CSIPD features and magnitude spectrum with a remainder mask obtained after subtracting the estimated mask from an all-one mask. The proposed SLOGD approach gave a WER of 44.2%. For comparison we trained a state-of-the-art speech separation method called Conv-TasNet (Luo and Mesgarani, 2019) on the same dataset and obtained a WER of 53.2%. This shows the importance of location information for far-field speech separation.

In Chapter 5, we proposed methods to explain the inner working of DNN-based speech enhancement models. In particular we explain the observation that a speech enhancement model trained on mixtures containing speech and artificially generated noise gave an ASR performance of 14.0% WER, comparable to a speech enhancement model trained on mixtures containing speech and Network noise which gave an ASR performance of 15.1% WER on the real CHIME-4 real evaluation set.
We use a feature attribution method called Deep SHapley Additive exPlanations (DeepSHAP) (Lundberg and Lee, 2017) to provide importance values to each time-frequency bin of the input spectrogram, thereby giving visual cues as to what aspect of the noisy speech

was useful for the DNN to predict the mask. DeepSHAP uses Shapley values (Shapley, 1953) which were originally proposed in game theory to distribute payout among the players in a game, to explain model output. It explains a model output by assuming that each feature value i.e., each time-frequency bin in the context of speech enhancement, is a player in a game — the game being the estimation of a mask by the model.
Based on the idea that a well-trained speech enhancement model with good generalization capabilities should mostly rely on time-frequency bins corresponding to speech rather than noise, we propose a metric called speech relevance score. The speech relevance score computes the percentage of input time-frequency bins having SHAP values above a threshold which belong to speech using an ideal binary mask as a reference. We showed that better ASR performance is observed for speech enhancement models with a higher speech relevance score. The generalization capability of a speech enhancement model trained using synthetically generated SSN as noise was shown to be better than that of a speech enhancement model trained using Network noise.

6.2 Perspectives and future directions

The work done in this thesis opens doors for further interesting areas of research. Some of these ideas are listed below.

6.2.1 Localization

6.2.1.1 End-to-end localization using textual information

In Chapter 3, the network that estimates the mask given the text and the network that estimates the DOA given the mask were trained independently of each other. These networks can be trained together instead, thereby updating the mask network with not only the gradients from the loss function associated with the mask but also the gradients from the loss function associated with the DOA as well. A similar approach was used by Zhang et al. (2019) but without the use of textual information. This idea can further be extended to train the localization network in an end-to-end fashion.
Neural networks are good at learning representations from raw signals. They have been used to replace signal processing based handcrafted representations such as the STFT and wavelets and are trained to learn filterbanks which adapt to different tasks at hand such as speech separation (Luo and Mesgarani, 2019), ASR (Sainath et al., 2017) or single-speaker binaural sound localization (Vecchiotti et al., 2019). A similar approach could also be used to incorporate textual information into the localization pipeline, by providing word or phone embeddings as contextual information in the architecture.

6.2.1.2 Target speaker identifiers

To localize a specific speaker in a mixture with multiple speakers, we estimate the mask of the target speaker first and use it to localize the speaker — as opposed to the localization-guided speech separation process where we estimate the location first and then estimate

the mask. As mentioned in Chapter 4, apart from the text spoken by the speaker, there are multiple other pieces of information which could be used to extract the target speaker from a mixture. Speaker identification information was used in SpeakerBeam (Žmolíková et al., 2019) to estimate the mask. The text-based mask estimation in Chapter 3 could be replaced or combined with speaker identification based mask estimation and used subsequently for localization.

6.2.1.3 Multi-task learning

As observed in Chapter 4, training a network to jointly perform localization and voice activity detection resulted in a good performance for both tasks. This was also shown in other multi-task learning scenarios such as joint speech enhancement and ASR (Chen et al., 2015). Further experiments need to be conducted to understand the symbiotic relationship associated with multi-task learning, i.e., how the two tasks benefit each other exactly. Further, the performance of the VAD network must be evaluated in real conditions such as on the CHIME-5 dataset.

6.2.1.4 Localization of moving sources

In a home environment we cannot expect users to stay at a single place. This is not a an issue for short utterances such as voice commands. For a localization-guided speech separation system to operate successfully on longer utterances, we need to track moving speakers. Localization for moving sources relies on good initial estimates, which can be estimated well using our text-informed localization approach since the wake-up word of the system is known a priori. An LSTM-based architecture can then be used to track the movements of the speaker. In addition, we may want to associate each trajectory with the corresponding speaker (Li et al., 2019).

6.2.1.5 Diarization

Diarization is the process of segmenting an audio file into smaller units, each of which is homogeneous, and marking the active speaker for each segment, i.e., figuring out who spoke when (Ryant et al., 2019). Diarization is an important component of a far-field ASR system since it is required to obtain speech segment boundaries and ensure that the ASR model receives audio segments corresponding to a single speaker.
In meeting scenarios, under the reasonable assumption that speakers do not move, localization can be used to improve the speaker diarization performance (Vijayasenan and Valente, 2012). Even in other scenarios when speakers move, we can assume that the speaker position is fixed for a short duration of time. Initial work done as part of the JSALT 2019 workshop using CHiME-5 data, where the speakers were assumed to stay in a position for a short duration of time (2.5 s), showed improved diarization performance compared to the baseline, using estimated speaker locations. More work needs to be done to understand how to use localization information when the speakers are moving.

6.2.2 Speech separation

Though the proposed SLOGD method improved the speech separation performance, error rates are still high and the speech separation models cannot be deployed in real situations. The problem of speech separation is still considered to be open. There are multiple approaches which need to be investigated to improve the performance as listed below.

6.2.2.1 Speech separation from the raw waveform

Speech separation from the raw waveform has given impressive performance in the absence of reverberation and noise. As shown by Heitkaemper et al. (2020), the major improvement comes from the use of the SI-SDR as a cost function instead of the MSE, with an acceptable degradation in the SDR (from 14.7 to 12.8 dB) when using the STFT instead of a learned filterbank. This gives rise to the possibility of using beamforming-based methods for multichannel speech separation from raw waveforms.
Replacing the R1-MWF beamformer by a differentiable beamformer as shown by Boeddeker et al. (2017), along with STFT-based filterbanks and the SI-SDR cost function could lead to improved speech separation performance. Speaker location information using the SLOGD approach could also be integrated into the pipeline.
These ideas can also be extended to train an end-to-end ASR model for speech separation. Menne et al. (2019a) used differentiable parametric Wiener filters and jointly trained a speech enhancement model along with the ASR acoustic model. This approach can be continued with the above speech separation methods using the raw waveform to obtain a truly end-end ASR system for speech separation.

6.2.2.2 Using visual cues

Multiple recent works have shown the use of video recordings to aid speech enhancement and separation systems (Ephrat et al., 2018; Sadeghi et al., pear; Sadeghi and Alameda-Pineda, 2020; Yu et al., 2020; Gao et al., 2020). The general idea is to learn embeddings for both the audio and video corresponding to the mixture $\mathbf{x}$ and fuse the embeddings to estimate the sources $\widehat{s}$. The gain in speech separation comes from lip reading and a drop in the performance was reported (Gao et al., 2020) when the lips are occluded. Occlusions often happen when the speaker turns away from the camera or another object comes in between the camera and the speaker. The method proposed by Gao et al. (2020) to overcome this issue is to use an attention mechanism to fuse the audio and video embeddings thereby training the network to weight the audio and video embeddings differently.
Apart from lip reading, visual cues also help in speaker localization (Alameda-Pineda and Horaud, 2015). With improved speaker localization and video embeddings and due to the fact that noise in the audio and the video are independent from each other, fusing information from video for localization can help localization-guided speech separation.

6.2.2.3 Informed speech separation

Speaker identity has been shown to provide useful information to separate speech from a mixture. In SpeakerBeam (Žmolíková et al., 2019), the network was informed about the speaker of interest using an embedding learned from a separate speaker embedding network. The speaker embedding was either concatenated to the inputs or used to scale certain weights of the network, similar to ASR acoustic model adaptation techniques (Swietojanski et al., 2016). The speaker identity can also help in localization as discussed in Section 6.2.1. Different architectures, using deflation-based strategies for example, could be designed to incorporate the localization and the speaker embedding to improve the speech separation performance.

6.2.2.4 Bayesian beamforming

As seen in Chapter 4, localization-based speech separation is sensitive to localization errors. Our approach to make the speech separation DNN robust to DOA errors was to train it along with the estimates of the DOA estimation network. We could not train the DOA and mask estimation networks jointly since the argmax operator, which is needed to obtain the DOA estimate, is not differentiable. Therefore we estimate the speaker as

$$\widehat{c}_j(\widehat{\theta}_j) = \mathbb{E}\{c_j \,|\, \mathbf{x}, \widehat{\theta}_j\}. \tag{6.1}$$

Another approach to incorporate the uncertainty in the DOA estimates is Bayesian beamforming (Lam and Singer, 2006), where the speech signals are estimated as

$$\widehat{c}_j = \sum_{k=1}^{\mathrm{P}} p(\theta_k \,|\, \mathbf{x})\, \widehat{c}_j(\theta_k). \tag{6.2}$$

A related work is reported by Chen et al. (2018b) who compute beams in all directions and optimize a DNN to choose the best beam for mask estimation. In this case the speech separation DNN is forced to learn $p(\theta_k \,|\, \mathbf{x})$ indirectly. We can instead replace the DS beamformed signal $\widehat{c}_{j,\mathrm{DS}}$ — the input to the mask estimation networks MASK_DNN$_1$ and MASK_DNN$_2$ — with the Bayesian beamformed estimates of Eq. (6.2). The estimates of $p(\theta_k \,|\, \mathbf{x})$ can be obtained using the DOA estimation networks DOA_DNN$_1$ and DOA_DNN$_2$. This is expected to lead to improved performance. Moreover, with this approach, we can jointly train the mask and DOA estimation networks which can further improve the robustness of the DNNs.

6.2.3 Interpreting predictions made by a DNN

In Chapter 5, we proposed a method to explain which components of the input spectrum were used to estimate the mask given the network architecture and the training data. An alternate approach could be to modify the architecture to incorporate an explainable component as demonstrated for a NLP task by Liu et al. (2019). Another important open question regarding model interpretation methods is how to exploit their outcomes to improve the model architecture and performance.

7 Résumé étendu

Les humains sont des êtres sociaux. Nous interagissons les uns avec les autres à l'aide d'un ensemble d'indices verbaux et visuels. Au fil du temps, nous avons développé des sons sophistiqués qui, lorsqu'ils sont liés les uns aux autres, ont un sens. Nous utilisons ces sons pour communiquer entre nous et transmettre des connaissances à travers les générations. Ces sons, appelés parole, ont été au cœur de notre évolution et sont une partie essentielle de notre vie quotidienne.
Avec l'avènement de l'industrialisation, nous avons développé des machines devenues omniprésentes. Il est naturel de vouloir interagir avec ces machines en utilisant la parole. La recherche de telles interactions a conduit à ce que l'on appelle communément les assistants vocaux et, avec l'amélioration de leur qualité et de leur robustesse, ces assistants ont fait des percées dans notre vie quotidienne. Des exemples typiques de ces assistants disponibles dans le commerce sont Google Assistant et Amazon Alexa. Ils se trouvent sur de nombreux appareils tels que les téléphones portables, les haut-parleurs intelligents, les réfrigérateurs, les téléviseurs, les télécommandes et les robots, et permettent une interaction plus facile. La demande pour de tels appareils à commande vocale a augmenté de façon exponentielle au fil des ans et son nombre devrait tripler, passant de 2,5 à 8 milliards d'ici 2023 (Perez, 2019). Le signal de parole capté par ces appareils est déformé en raison de la réverbération, de la parole interférente et du bruit (Loizou, 2013; Virtanen et al., 2012; Wölfel and McDonough, 2009). Ces distorsions réduisent l'intelligibilité de la parole et affectent les performances des assistants vocaux. L'objectif de cette thèse est d'estimer la position spatiale du locuteur et de l'utiliser pour extraire la parole cible en présence de telles distorsions.
Ce résumé étendu est organisé comme suit. La Partie 7.1 traite de la localisation du locuteur, la Partie 7.2 de la séparation de la parole et la Partie 7.3 de l'explication des résultats de modèles d'apprentissage automatique utilisés pour la séparation.

7.1 Localisation informée par le texte

Dans le Chapitre 3 de ce manuscrit, une nouvelle tâche de localisation spatiale exploitant la transcription textuelle de la parole prononcée par le locuteur cible est introduite afin d'améliorer la performance de localisation dans des conditions acoustiques défavorables contenant de la réverbération, du bruit et des locuteurs interférents. Une telle information textuelle peut être obtenue dans les systèmes tels que Google Assistant et Amazon Alexa qui utilisent un mot-clé ("OK Google" ou "Alexa") pour activer le système de reconnaissance automatique de la parole. Ici, nous considérons une situation de commande vocale mains-libres, où la parole est enregistrée par une antenne compacte constituée de

deux microphones. Dans cette situation, estimer la direction d'arrivée de la voix équivaut à estimer le délai d'arrivée entre les microphones. Pour cela, un réseau de neurones est entraîné à transformer les descripteurs d'entrée en directions d'arrivée discrétisées.
La localisation de plusieurs locuteurs simultanés introduit des difficultés supplémentaires. Plusieurs travaux ont attaqué ce problème en utilisant des méthodes basées sur le traitement du signal (Schmidt, 1986; Nakamura et al., 2013; Rascon and Meza, 2017) ou sur l'apprentissage (Chakrabarty and Habets, 2017b). Néanmoins, la tâche de localiser un locuteur cible particulier dans un environnement multi-locuteurs n'a pas été abordée auparavant. Cette tâche pose deux défis principaux. Le premier défi concerne la représentation des informations nécessaires pour identifier le locuteur cible sous forme d'un identifiant numérique et le second concerne l'incorporation de cet identifiant dans le système de localisation. Dans ce travail, les informations phonétiques extraites du signal capté sachant le mot-clé prononcé par le locuteur cible sont utilisées comme identifiant pour estimer la direction d'arrivée de ce locuteur en présence d'un locuteur interférent.
Le reste de cette partie est organisé comme suit. La définition du problème et l'aperçu de la solution proposée sont décrits dans la Partie 7.1.1. L'estimation de l'identifiant cible est expliquée dans la Partie 7.1.2. La Partie 7.1.3 détaille la configuration expérimentale. Les résultats sont discutés et analysés dans la Partie 7.1.4. La Partie 7.1.5 résume les conclusions.

7.1.1 Définition du problème

Une paire de microphones espacés d'une distance ℓ est placée dans une position et une orientation arbitraires dans une pièce. Un locuteur cible s_1 et un locuteur interférent s_2 parlant simultanément sont enregistrés par les deux microphones. Le but est d'estimer la direction d'arrivée θ du locuteur cible en utilisant le signal $\mathbf{x}(t) = [x_1(t), \ldots, x_I(t)]^T$ captés par les microphones et le mot-clé (texte) prononcé par le locuteur cible. Le signal capté par le microphone i est

$$x_i(t) = c_{i\,1}(t) + c_{i\,2}(t) + \mathrm{u}_i(t) \tag{7.1}$$

où $c_{i\,j}$ est l'image spatiale de la source de parole j et u est le bruit ambiant.

7.1.1.1 Composantes directe, précoce et réverbérée

Trois composantes spatiales différentes d'une source par rapport à un microphone peuvent être envisagées, à savoir

1. la composante directe $c_{i\,j}^{D}(t)$ définie par

$$c_{i\,j}^{D}(t) = \sum_{\tau=0}^{\tau_D} a_{i\,j}(\tau)\, s_j(t-\tau) \tag{7.2}$$

où τ_D est le temps de propagation du son en ligne droite entre la source et le microphone et $a_{i\,j}$ est la réponse impulsionnelle de salle entre la source j et le microphone i,

2. la composante précoce $c_{ij}^{E}(t)$ définie par

$$c_{ij}^{E}(t) = \sum_{\tau=0}^{\tau_E} a_{ij}(\tau)\, s_j(t-\tau) \tag{7.3}$$

où τ_E est le temps correspondant aux échos précoces fixé à 50 ms,

3. la composante réverbérée ou l'image spatiale complète $c_{ij}^{R}(t)$ définie par

$$c_{ij}^{R}(t) = \sum_{\tau=0}^{\infty} a_{ij}(\tau)\, s_j(t-\tau). \tag{7.4}$$

Nous supposons que les locuteurs sont en champ lointain et que la distance entre les microphones est petite ($\ell = 10$ cm). Dans le domaine temps-fréquence, un spectre d'amplitude représentatif de la source j peut donc être calculé en moyennant les spectres d'amplitude de l'image spatiale de cette source sur tous les microphones. Nous définissons ainsi le spectre d'amplitude $|\, c_j^D(n,f)|$ de la composante directe de la source j à la trame n et dans la bande de fréquence f comme

$$|\, c_j^D(n,f)| = \frac{1}{I} \sum_i |\, c_{ij}^D(n,f)| \tag{7.5}$$

et de même les spectres d'amplitude $|\, c_j^E(n,f)|$ et $|\, c_j^R(n,f)|$ des composantes précoce et réverbérée.

7.1.1.2 Aperçu de la méthode proposée

La Fig. 7.1 fournit un aperçu de la solution proposée. Après transformation des signaux dans le domaine temps-fréquence par le biais de la transformée de Fourier à court terme, le cosinus et le sinus des différences de phase entre les signaux captés par les deux microphones (appelés descripteurs CSIPD par la suite) sont calculés. Un identifiant du locuteur cible est obtenu par un réseau de neurones convolutif prenant en entrée le spectre d'amplitude du signal capté et le spectre moyen correspondant à chaque phone prononcé. L'alignement phonétique nécessaire pour sélectionner le spectre moyen correspondant au phone prononcé par le locuteur cible à chaque instant est obtenu par un système de reconnaissance automatique de la parole informé par le texte prononcé. Les descripteurs CSIPD et l'identifiant ainsi obtenu sont utilisés pour entraîner un réseau de neurones à estimer la direction d'arrivée de ce locuteur. Chacun de ces blocs de traitement est détaillé ci-dessous.

7.1.2 Identifiant du locuteur cible et localisation

7.1.2.1 Calcul du spectre moyen de chaque phone

Les spectres moyens de tous les phones sont précalculés. Pour cela, les spectres d'amplitude de tous les segments de parole correspondant à un phone donné dans l'ensemble d'apprentissage sont moyennés (Erdogan et al., 2015; Chen et al., 2015). Chaque phone a un

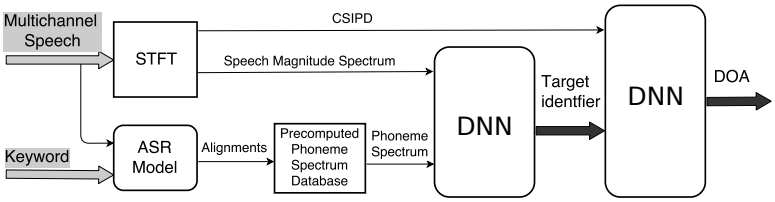


Figure 7.1: Schéma global de la méthode proposée de localisation informée par le texte.

spectre moyen distinct qui guide l'identification des points temps-fréquence dominés par le locuteur cible.

7.1.2.2 Identifiant du locuteur cible

L'identifiant du locuteur cible est un masque temps-fréquence représentant la proportion de parole cible présente en chaque point temps-fréquence. En pratique, nous pouvons estimer la proportion de parole cible directe, précoce ou réverbérée. Pour calculer le masque correspondant à une composante cible particulière, cette composante est d'abord soustraite du signal capté pour obtenir le reste. Le rapport entre le spectre de la composante cible et la somme des spectres de la composante cible et du reste est appelé masque identifiant. Le masque a donc des valeurs réelles dans l'intervalle $[0, 1]$.
Le masque identifiant correspondant à composante directe est par exemple défini comme

$$\mathcal{M}_1^D(n,f) = \frac{|\, c_1^D(n,f)|}{|\, c_1^D(n,f)| + |\delta_1^D(n,f)|} \tag{7.6}$$

où

$$|\delta_1^D(n,f)| = \frac{1}{I}\sum_i |\delta_{i1}^D(n,f)| \tag{7.7}$$

$$\delta_{i1}^D(n,f) = x_i(n,f) - c_{i1}^D(n,f). \tag{7.8}$$

Ici δ_1^D représente le reste obtenu après soustraction de la composante directe du locuteur cible du signal total capté.
Il est important de calculer le dénominateur du rapport comme la somme des spectres de la composante cible et du reste au lieu d'utiliser le spectre du mélange capté. En effet, le spectre du mélange $|\, x\,|$ est sujet à des interférences destructives qui entraîneraient des valeurs de masque supérieures à 1. Les masques identifiants précoce et réverbéré sont définis de la même manière.

7.1.2.3 Estimation de l'identifiant

Un segment de $0,5$ s de parole est fenêtré en 11 trames longues de 100 ms chacune avec un décalage de 50 ms. Chaque trame longue est ensuite fenêtrée en 8 trames courtes de 25 ms chacune avec un décalage de 10 ms. Les descripteurs d'entrée pour l'estimation de l'identifiant du locuteur cible comprennent les spectres d'amplitude des trames courtes et

les spectres moyens des phones prononcés. La dimension de ces descripteurs est donc $3 \times 8 \times 201 \times 11$ où 201 est la dimension des spectres des trames courtes. Un réseau de neurones convolutif et récurrent est utilisé. Il comprend 4 couches convolutives composées de 64, 32, 16 et 4 filtres de taille $3 \times 3 \times 3$. La normalisation par lots (Ioffe and Szegedy, 2015) et le *dropout* (Srivastava et al., 2014) sont utilisés pour toutes les couches convolutives. Le *max pooling* est utilisé dans les deuxième, troisième et quatrième couches le long de l'axe des trames courtes tout en conservant les axes des fréquences et des trames longues. Une couche récurrente seuillée bidirectionnelle (Cho et al., 2014a) est ensuite utilisée le long de l'axe des trames longues suivie d'une couche de sortie dense distribuée dans le temps. La fonction de rectification (ReLU) est utilisée comme non-linéarité dans toutes les couches convolutives. Une non-linéarité sigmoïde est appliquée à la sortie pour apprendre les masques identifiants. Les paramètres du réseau sont appris grâce à l'optimiseur Adam (Kingma and Ba, 2015) avec l'erreur quadratique moyenne pour fonction de coût.

7.1.2.4 Estimation de la direction d'arrivée

Les descripteurs CSIPD et les spectres d'amplitude sont multipliés point-à-point par le masque identifiant estimé pour former des descripteurs de taille $3 \times 801 \times 11$. Ces descripteurs sont traités par un réseau de neurones composé de quatre couches convolutives contenant respectivement 64, 32, 16 et 3 filtres. La normalisation par lots et le *dropout* sont utilisés pour toutes les couches. Le *max pooling* est utilisé pour toutes les couches, à l'exception de la première couche convolutive. La sortie de la dernière couche convolutive est traitée par une couche récurrente seuillée bidirectionnelle de 198 unités. La sortie des deux directions est concaténée et envoyée à une couche dense distribuée dans le temps de 512 unités qui alimente une autre couche dense distribuée dans le temps contenant 181 unités avec une non-linéarité *softmax* afin d'obtenir les probabilités *a posteriori* de l'ensemble de directions d'arrivée discrètes possibles. La non-linéarité ReLU est utilisée dans toutes les couches convolutives et denses.

7.1.3 Protocole expérimental

Nous avons mené des expériences sur des données réelles et simulées.
En ce qui concerne les données simulées, le logiciel RIR Simulator (Habets, 2018) a été utilisé pour simuler les réponses de salle correspondant aux positions des microphones et des locuteurs. Toutes les combinaisons possibles de directions d'arrivée de la source cible et de la source interférente sont considérées sous la contrainte d'une distance angulaire minimale de 5°. Pour chaque paire d'angles d'arrivée, 50, 1 et 2 configurations de salles sont ainsi simulées dans les ensembles d'apprentissage, de validation et de test, ce qui correspond à un total de 1 557 600, 31 152, et 62 304 configurations, respectivement. Pour maximiser la variabilité des données, les dimensions de la pièce et le temps de réverbération varient à chaque configuration.
Les données réelles ont été enregistrées dans le cadre du projet ANR Vocadom. Plusieurs mots-clés sont utilisés dans le projet, à savoir: « Allô Cirrus », « Allô Messire », « Chanticou », « Dis Bérénio », « Dis Hestia », « Dis Téraphim », « Dis Vesta », « Dis Vocadom »,

« Hé Cirrus », « Ichefix », « Minouche », « Téraphim », « Ulysse » et « Vocadom ». Les données ont été enregistrées à l'aide d'un réseau de 4 microphones MEMS sous forme de signaux à 16 bits et 44,1 kHz. Ces signaux sont sous-échantillonnés à 16 kHz avant de calculer les descripteurs. Les microphones sont placés sur les sommets d'un carré dont la diagonale est de 10 cm. Les données captées par les paires de microphones sur l'une ou l'autre diagonale sont utilisées pour la localisation.

7.1.4 Résultats

Trois métriques sont utilisées pour évaluer la performance de localisation.

1. Le taux d'erreur grossière mesure le pourcentage de directions d'arrivée estimées dont la différence en valeur absolue par rapport à la direction réelle du locuteur cible est supérieure à un certain seuil.
2. Le taux de proximité à l'interférence measure le pourcentage de directions d'arrivée estimées dont la différence en valeur absolue par rapport à la direction réelle du locuteur interférent est inférieure à ce même seuil.
3. L'erreur absolue moyenne est la moyenne de la différence en valeur absolue en degrés entre la direction d'arrivée estimée et la direction réelle du locuteur cible.

La Fig. 7.2 montre les résultats obtenus par notre méthode sur l'ensemble de test simulé. À titre de comparaison, l'approche classique GCC-PHAT a un taux d'erreur grossière de $50,9\%$, un taux de proximité à l'interférence de $29,2\%$ et une erreur absolue moyenne de $27,5\%$. Une réduction significative de toutes ces métriques est obtenue en appliquant le masque identifiant « vérité terrain », ce qui montre que ce masque est un bon identifiant du locuteur cible. Un taux d'erreur grossière de $13,9\%$ a été obtenu en utilisant le masque identifiant précoce estimé, ce qui représente une amélioration relative de $72,7\%$ par rapport à GCC-PHAT. Étant donné que les taux de proximité à l'interférence du masque estimé et du masque « vérité terrain » ($2,3\%$ et $1,7\%$) sont très similaires, nous pouvons conclure que le masque identifiant précoce estimé fournit toutes les informations nécessaires pour différencier la cible de l'interférence.

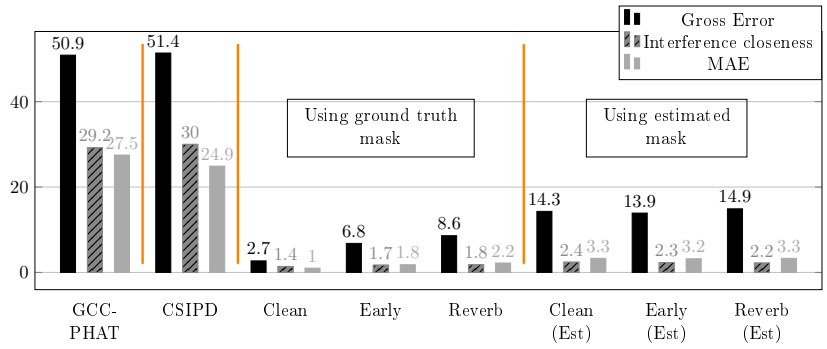


Figure 7.2: Performance de localisation de la méthode proposée comparée à GCC-PHAT sur l'ensemble de test simulé.

Des améliorations similaires ont été observées lors de l'expérimentation avec des données

réelles lorsque le rapport signal-à-interférence est de l'ordre de 0 dB.

7.1.5 Conclusion

Dans cette partie, nous avons proposé une méthode pour exploiter le texte prononcé par un locuteur cible afin d'améliorer la localisation de ce locuteur dans des environnements à plusieurs locuteurs. Le texte prononcé est aligné temporellement avec le signal capté par reconnaissance automatique de la parole et l'alignement phonétique obtenu est utilisés pour calculer un identifiant du locuteur cible. Cet identifiant est utilisé avec les descripteurs CSIPD pour localiser le locuteur cible. Des expériences sur des données réelles et simulées montrent l'efficacité de la méthode proposée.

7.2 Séparation de sources de parole

Le problème de la séparation de sources de parole a été étudié à la fois dans un contexte monocanal et multicanal. Les approches monocanales récentes incluent des méthodes basées sur le *clustering* telles que le *deep clustering* (Hershey et al., 2016) et les réseaux attracteurs profonds (Chen et al., 2017) où un réseau de neurones apprend à regrouper les points temps-fréquence dominés par le même locuteur. Une autre approche consiste à estimer les locuteurs de manière itérative (Kinoshita et al., 2018) en utilisant des réseaux de neurones avec des critères d'apprentissage invariants par permutation (Kolbæk et al., 2017).

Cette partie, qui résume le Chapitre 4 du manuscrit, fournit les contributions suivantes:

1. Nous créons un nouveau corpus multicanal, multi-locuteur, réverbéré et bruité qui étend le corpus originel WSJ0-2mix monocanal, non réverbéré et non bruité (Hershey et al., 2016) aux conditions de réverbération et de bruit puissantes et aux antennes de microphones de type Kinect considérées dans le défi CHiME-5 (Barker et al., 2018). Cela nous permet d'utiliser le bruit réel du corpus CHiME-5, ce qui rend notre corpus simulé particulièrement réaliste et difficile.
2. Sur ce corpus, nous séparons les sources de parole en utilisant soit la position réelle des locuteurs soit la position estimée par l'algorithme GCC-PHAT (Knapp and Carter, 1976), et nous évaluons le taux d'erreur sur les mots (WER) obtenu par reconnaissance automatique de la parole sur les signaux séparés. Une analyse similaire de l'impact des erreurs de localisation sur le WER a été réalisée par Barfuss and Kellermann (2016), mais dans des conditions acoustiques différentes et avec une taille de vocabulaire limitée.
3. Nous proposons un algorithme de séparation des sources de parole par déflation guidée par la localisation (SLOGD) conçu pour être robuste aux erreurs de localisation.

Le reste de cette partie est organisé comme suit. La Partie 7.2.1 présente le cadre proposé pour la séparation de la parole utilisant la position spatiale des locuteurs. La Partie 7.2.2 détaille la méthode SLOGD proposée pour améliorer la robustesse du réseau de séparation de la parole. La Partie 7.2.3 décrit la configuration expérimentale et les

résultats obtenus sont discutés dans la Partie 7.2.4. Nous résumons les conclusions dans la Partie 7.2.5.

7.2.1 Séparation de sources de parole guidée par la localisation

Notre objectif est d'estimer l'image spatiale de chaque source de parole compte tenu de sa position spatiale (connue ou estimée). Une vue d'ensemble de notre approche est présentée dans la Fig. 7.3. Cette approche comprend trois étapes:

1. formation de voie *delay-and-sum* (DS),
2. estimation du masque temps-fréquence,
3. formation de voie adaptative.

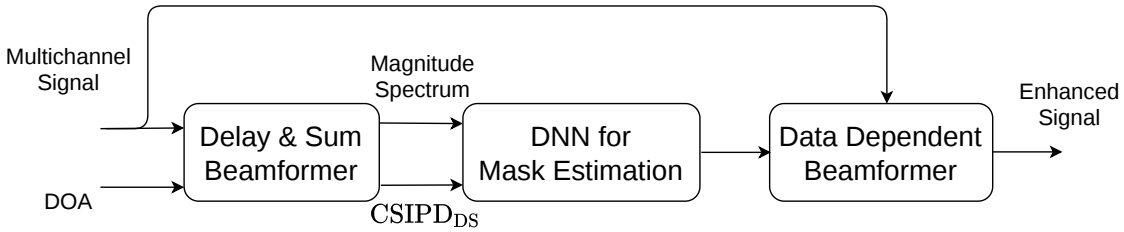


Figure 7.3: Schéma de séparation de sources guidée par la direction d'arrivée.

Formation de voie DS Étant donnée la position spatiale de la source j en champ lointain, la différence de temps d'arrivée correspondante entre deux microphones i et i' est égale à

$$\Delta_{i\,i'\,j} = \frac{\ell_{i\,i'}\cos(\theta_{i\,i'\,j})}{c} \tag{7.9}$$

où $\theta_{i\,i'\,j}$ est la direction d'arrivée de la source par rapport à l'axe des microphones, $\ell_{i\,i'}$ est la distance entre les microphones et c est la célérité du son.
Le vecteur directionnel relatif à un microphone de référence (par la suite, le microphone 1) est défini par $\tilde{\mathbf{d}}_j(f) = [1, e^{-2j\pi\,\Delta_{21\,j}\,\nu_{\mathrm{f}}}, \ldots, e^{-2j\pi\,\Delta_{I\,1\,j}\,\nu_{\mathrm{f}}}]^T$, où ν_{f} est la fréquence en Hertz correspondant à la bande de fréquence f. La formation de voie DS pour la source j consiste à calculer

$$\widehat{\mathbf{c}}_{j,\mathrm{DS}}(n,f) = \tilde{\mathbf{d}}_j^H(f)\,\mathbf{x}(n,f) \tag{7.10}$$

où ^{H} est la transposition hermitienne.
Le nombre de points temps-fréquence dominés par le locuteur j est plus élevé dans $\widehat{\mathbf{c}}_{j,\mathrm{DS}}(n,f)$ que dans $\mathbf{x}$. Nous utilisons donc $\widehat{\mathbf{c}}_{j,\mathrm{DS}}$ pour calculer le masque temps-fréquence correspondant au locuteur j.

Estimation du masque temps-fréquence Le spectre d'amplitude de $\widehat{\mathbf{c}}_{j,\mathrm{DS}}$ et sa différence de phase par rapport au microphone de référence sont utilisés comme entrées d'un réseau de neurones qui estime le masque temps-fréquence $\mathcal{M}_j(n,f)$ correspondant au locuteur j. Le spectre d'amplitude de $\widehat{\mathbf{c}}_{j,\mathrm{DS}}$ a déjà été utilisé par Perotin et al. (2018)

et Chen et al. (2018a). La différence de phase entre $\widehat{\mathbf{c}}_{j,\mathrm{DS}}$ et le microphone de référence s'exprime comme

$$\phi_{\widehat{\mathbf{c}}_{j,\mathrm{DS}}}(n,f) = \angle \widehat{\mathbf{c}}_{j,\mathrm{DS}}(n,f) - \angle x_1(n,f). \tag{7.11}$$

Formation de voie adaptative Le masque estimé $\widehat{\mathcal{M}}_j(n,f)$ est à son tour utilisé pour estimer la matrice de covariance multicanale de la source j

$$\mathbf{\Sigma}_j(f) = \frac{1}{N}\sum_n \widehat{\mathcal{M}}_j(n,f)^2\,\mathbf{x}(n,f)\,\mathbf{x}^H(n,f). \tag{7.12}$$

De même, la matrice de covariance multicanale du bruit, qui inclut les statistiques correspondant à toutes les autres sources et au bruit de fond, peut être estimée comme

$$\mathbf{\Sigma}_{\mathbf{u}}(f) = \frac{1}{N}\sum_n (1-\widehat{\mathcal{M}}_j(n,f))^2\,\mathbf{x}(n,f)\,\mathbf{x}^H(n,f). \tag{7.13}$$

Une formation de voie adaptative, c'est-à-dire qui dépend des statistiques ci-dessus plutôt que de la position spatiale, est appliquée au signal de mélange $\mathbf{x}(n,f)$ pour estimer les signaux sources. La sortie de la formation de voie est $\mathbf{w}^H(n,f)\mathbf{x}(n,f)$. Différentes formations de voies peuvent être appliquées en fonction du critère d'optimisation choisi (Wölfel and McDonough, 2009; Gannot et al., 2017). Dans ce travail, nous considérons la formation de voie par valeur propre généralisée (GEV) (Warsitz and Haeb-Umbach, 2007), le filtre de Wiener multicanal pondéré (SDW-MWF) (Spriet et al., 2004), et le filtre de Wiener multicanal avec contrainte de rang 1 (R1-MWF) (Wang et al., 2018b).

7.2.2 Séparation par déflation guidée par la localisation

Une localisation erronée affecte à la fois l'amplitude et la phase du signal $\widehat{\mathbf{c}}_{j,\mathrm{DS}}(n,f)$ estimé par formation de voie DS. Pour traiter ce problème et permettre la séparation de toutes les sources, nous proposons une stratégie de déflation où nous estimons de manière itérative les sources et les supprimons du mélange avant d'estimer la source suivante comme le montre la Fig. 7.4.

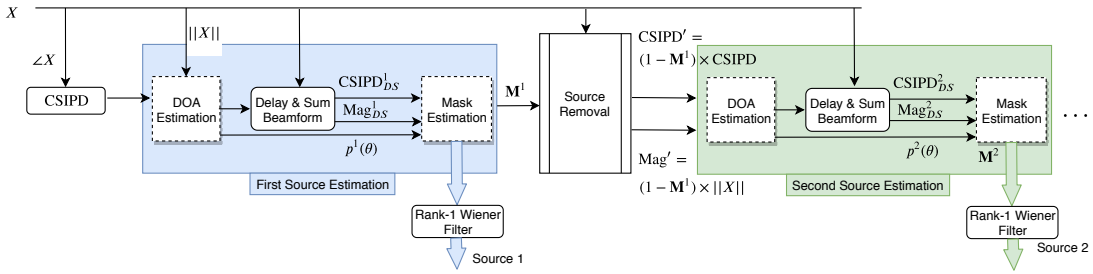


Figure 7.4: Estimation itérative de la direction d'arrivée et du masque.

Estimation de la première source Dans la première étape, nous estimons la direction d'arrivée de l'un des locuteurs à l'aide d'un réseau de neurones. Les descripteurs CSIPD de toutes les paires de microphones et le spectre d'amplitude à court terme de l'un des microphones (par exemple, le microphone 1) sont utilisés comme entrées. La probabilité a posteriori $p_1(n, \theta)$ de chaque direction d'arrivée θ estimée par le réseau sur chaque trame n est moyennée sur l'ensemble des trames, et la direction d'arrivée correspondant à la probabilité la plus élevée est sélectionnée, c'est-à-dire $\widehat{\theta}_1 = \text{argmax}_{\theta \in \{1,\dots,\text{P}\}} \sum_n p_1(n, \theta)$. Pour que la direction d'arrivée estimée corresponde à l'un des locuteurs, l'apprentissage repose sur un critère invariant par permutation.
Dans la deuxième étape, étant donnée la direction d'arrivée $\widehat{\theta}_1$ estimée, nous calculons le masque temps-fréquence correspondant en utilisant un autre réseau de neurones comme décrit ci-dessus. Outre le spectre d'amplitude $\mathbf{Mag}^1_{\text{DS}}$ du signal après formation de voie et sa différence de phase $\mathbf{CSIPD}^1_{\text{DS}}$ par rapport au premier canal, la sortie $\mathbf{p}_1(n)$ du réseau de localisation est également utilisée comme entrée du réseau.

Estimation de la deuxième source Dans la troisième étape, nous retirons le locuteur estimé du mélange. Kinoshita et al. (2018) calculent le masque restant après chaque itération et le concatènent aux entrées du réseau pour estimer la source suivante. De façon similaire, nous calculons le masque restant $(1 - \widehat{\mathcal{M}}_1)$ mais, au lieu de le concaténer aux entrées du réseau, nous multiplions point-à-point le spectre d'amplitude $\mathbf{Mag}^1_{\text{DS}}$ et les descripteurs CSIPD $\mathbf{CSIPD}^1_{\text{DS}}$ par ce masque avant de les utiliser comme entrées pour les étapes de localisation et d'estimation de masque suivantes. Comme indiqué ci-dessus, la multiplication par le masque fonctionne en effet mieux que la concaténation pour la localisation du locuteur.
Dans la quatrième étape, similaire à la première, nous estimons la direction d'arrivée du deuxième locuteur à l'aide d'un réseau de neurones prenant en entrée les descripteurs CSIPD et le spectre d'amplitude du signal d'origine après multiplication par le masque restant.
Enfin, la cinquième étape consiste à appliquer la formation de voie DS en utilisant la direction estimée et à en déduire le masque pour le deuxième locuteur.
La méthode proposée peut en théorie estimer les directions d'arrivée et les masques pour n'importe quel nombre J' de locuteurs en utilisant une méthode de comptage des sources (Kinoshita et al., 2018). Dans ce qui suit, nous supposons $J' = 2$.

7.2.3 Protocole expérimental

Corpus L'approche proposée est évaluée sur un nouveau corpus, qui constitue une version multicanale, réverbérée et bruitée du corpus originel WSJ0-2mix[1]. Nous émulons les conditions d'enregistrement du corpus CHiME-5 qui a été acquis à l'aide d'antennes de 4 microphones Microsoft Kinect. Pour chaque paire de signaux de parole dans WSJ0-2mix, nous simulons les réponses impulsionnelles de la pièce en utilisant RIR Simulator (Ha-

[1]Le code pour recréer le corpus est distribué publiquement: https://github.com/sunits/Reverberated_WSJ_2MIX

bets, 2018) pour deux positions spatiales distinctes avec une différence d'angle d'arrivée minimale de 5°. Les dimensions de la pièce et le temps de réverbération sont tirés aléatoirement dans les plages suivantes: $[3-9]$ m et $[0.3-1]$ s. Les deux signaux de parole sont convolués avec ces réponses de salle et mélangés avec un rapport signal-à-interférence dans la plage $[0-10]$ dB. Un signal de bruit multicanal réel issu du corpus CHIME-5 est ensuite ajouté avec un rapport signal-à-bruit dans la plage $[0-10]$ dB. Pour trouver les parties du corpus CHiME-5 contenant uniquement du bruit, les étiquettes de détection d'activité vocale du défi DIHARD-II (Ryant et al., 2019) sont utilisées plutôt que les étiquettes originelles du corpus CHiME-5, car elles sont plus fiables. Les signaux de bruit des ensembles d'apprentissage, de développement et de test proviennent de différentes sessions d'enregistrement de CHiME-5. Ces signaux de bruit non-stationnaires rendent la tâche de séparation de la parole très difficile.

Descripteurs pour l'estimation des masques La transformée de Fourier est calculée en utilisant des fenêtres de 50 ms avec un décalage de 25 ms, ce qui donne 801 bandes de fréquence. Les 4 microphones de l'antenne forment 6 paires de microphones. Étant donné que les caractéristiques CSIPD sont constituées des sinus et des cosinus des différences de phase entre toutes les paires de microphones, 12 vecteurs de descripteurs CSIPD, chacun de dimension 801, sont obtenus pour chaque trame. Le masque idéal d'amplitude $\mathcal{M}^{\text{IRM}}$ correspondant au locuteur c_j est utilisé comme cible pour l'apprentissage.

7.2.4 Résultats

Le Tableau 7.1 concerne l'évaluation de la reconnaissance automatique de la parole avant la séparation. Un WER de $12,5\%$ est obtenu sur un signal constitué d'un seul locuteur avec de la réverbération. Le WER passe à $25,5\%$ avec l'ajout de bruit, ce qui correspond à une dégradation relative de 104%. Une dégradation relative supplémentaire de 160% est observée lorsque le mélange contient un locuteur interférent en plus de la réverbération et du bruit. Cela montre que le bruit de fond et la parole superposée ont un impact énorme sur les performances de reconnaissance automatique de la parole et que des méthodes spécifiques sont nécessaires pour faire face à ces distorsions.

Table 7.1: WER (%) obtenu sur des signaux réverbérés avant séparation.

Signal	**1 locuteur**	**1 locuteur + bruit**	**2 locuteurs + bruit**
WER	12.5%	25.5%	66.5%

Le Tableau 7.2 concerne l'évaluation de la reconnaissance automatique de la parole après séparation de sources. La méthode SLOGD proposée est comparée à Conv-TasNet, qui constitue l'état de l'art de la séparation de parole monocanale. SLOGD obtient un WER de $44,2\%$, soit une amélioration relative de 34% par rapport à la reconnaissance automatique de la parole avant séparation et une amélioration relative de 18% par rapport

à la séparation basée sur la localisation estimée par GCC-PHAT. En comparaison, Conv-TasNet obtient un WER de $53,2\%$ sur notre corpus.

Table 7.2: WER (%) obtenu sur des mélanges réverbérés (2 locuteurs + bruit) après séparation.

Méthode	WER
Séparation basée sur la direction d'arrivée réelle	35.1%
Séparation basée sur la direction estimée par GCC-PHAT	54.2%
SLOGD	44.2%
Conv-TasNet	53.2%

7.2.5 Conclusion

Dans cette partie, nous avons abordé le problème de la séparation de sources de parole distantes dans des conditions de bruit et de réverbération difficiles. Pour ce faire, nous avons créé un nouveau corpus en ajoutant de la réverbération et du bruit CHiME-5 réel au corpus WSJ0-2mix originel. De plus, nous avons proposé une stratégie basée sur la déflation pour la séparation de la parole guidée par la localisation. Pour chaque itération, nous entraînons un réseau de neurones à localiser un locuteur et utilisons la localisation obtenue pour estimer le masque temps-fréquence correspondant à ce locuteur. La source estimée est ensuite supprimée en masquant les descripteurs d'entrée correspondants avant d'extraire la source suivante. En utilisant cette approche, nous obtenons un WER de $44,2\%$, significativement meilleur que le WER de $53,2\%$ obtenu par Conv-TasNet.

7.3 Analyse des réseaux de neurones pour le rehaussement

Les modèles de rehaussement de la parole sont souvent appris de manière supervisée à l'aide de données simulées. Ces données simulées sont générées en mélangeant des signaux de parole et de bruit à différents rapports signal-à-bruit et le modèle est entraîné à estimer les spectres de parole et de bruit ou un masque temps-fréquence. Différents types de bruits ont été utilisés pour cela. Cela soulève la question de savoir quel type de bruit convient le mieux pour entraîner le réseau. Un bruit réel correspondant aux conditions dans lesquelles le modèle sera déployé est un bon choix, cependant l'enregistrement de scènes de bruit couvrant toutes ces conditions est coûteux et souvent irréalisable. Une alternative consiste à utiliser du bruit artificiel, à condition que l'impact sur la qualité de rehaussement ne soit pas drastique.
Dans cette partie, qui résume le Chapitre 5 du manuscrit, nous montrons qu'un modèle de rehaussement entraîné avec un bruit artificiel de type *speech-shaped noise* (SSN) améliore considérablement le WER sur le corpus CHiME-4, mais légèrement moins qu'un modèle de rehaussement appris avec un bruit CHiME-4 similaire aux conditions de test. Nous nous concentrons sur l'explication de ce résultat, comme première étape vers la prévision de la capacité de généralisation des modèles de rehaussement et le choix des bruits

d'apprentissage optimaux à l'avenir. Pour ce faire, nous utilisons la méthode DeepSHAP (Lundberg and Lee, 2017) pour quantifier l'importance de chaque point temps-fréquence du spectrogramme d'entrée dans l'estimation du masque de sortie.
La Partie 7.3.1 fournit un aperçu de DeepSHAP. La Partie 7.3.2 décrit l'application de DeepSHAP à la tâche de rehaussement. La Partie 7.3.3 propose une mesure objective pour analyser les valeurs SHAP obtenues. La Partie 7.3.4 détaille le protocole expérimental et la Partie 7.3.5 examine les résultats. Nous concluons dans la Partie 7.3.7.

7.3.1 DeepSHAP

Nous désignons temporairement le vecteur de descripteurs d'entrée comme

$$\mathbf{x} = [x_1, \ldots, x_D] \tag{7.14}$$

où D est le nombre de descripteurs, et nous supposons que la sortie $\mathcal{F}(\mathbf{x})$ est un scalaire. SHAP calcule la pertinence d'un descripteur x_d en observant le changement dans la sortie par rapport à sa présence ou son absence. Pour éviter de réapprendre le réseau pour chaque combinaison de descripteurs présents et absents, l'absence d'un descripteur est approchée en le remplaçant par sa valeur moyenne. Cela peut être représenté par une projection locale des descripteurs d'entrée $\mathbf{x}$ dans un vecteur d'entrée simplifié $\mathbf{x}' = \{x'_1, \ldots, x'_D\}$ de même dimension D en utilisant une fonction $h_{\mathbf{x}}(.)$ telle que

$$[h_{\mathbf{x}}(\mathbf{x}')]_d = \begin{cases} x_d & \text{si } x'_d = 1 \\ \mathbb{E}(x_d) & \text{si } x'_d = 0 \end{cases} \tag{7.15}$$

où chaque entrée simplifiée $x'_d \in \{0, 1\}$ indique la présence ou l'absence du descripteur correspondant. SHAP approche la sortie du réseau sous forme d'une combinaison linéaire des entrées simplifiées:

$$\mathcal{F}(h_{\mathbf{x}}(\mathbf{x}')) \approx \phi_0 + \sum_{d=1}^{D} \phi_d \, x'_d \,. \tag{7.16}$$

Chaque poids ϕ_d est appelé valeur SHAP et il quantifie directement la pertinence du descripteur correspondant.
En pratique, le remplacement de chaque descripteur absent par sa valeur moyenne est une mauvaise approximation lorsqu'il est appliqué à un réseau de neurones dans son ensemble. DeepSHAP combine un calcul analytique efficace des valeurs SHAP pour les briques simples du réseau (linéaire, max, activation) avec la règle de composition de DeepLIFT (Shrikumar et al., 2017a) pour rétro-propager ces valeurs jusqu'à la couche d'entrée.

7.3.2 Calcul des valeurs SHAP pour des modèles de rehaussement

Dans cette partie, nous revenons aux notations standard utilisées dans la thèse, c'est-à-dire que le signal multicanal est défini par

$$\mathbf{x}(n, f) = [x_1(n, f), \ldots, x_I(n, f)]^T \tag{7.17}$$

où I est le nombre de microphones. Étant donné que ce travail se concentre sur le rehaussement, il existe une seule source ($J' = 1$). Le mélange peut s'écrire

$$x(n, f) = c_1(n, f) + u(n, f). \tag{7.18}$$

Un réseau de neurones est entraîné à estimer le masque d'amplitude optimal $\mathcal{M}$ en utilisant le spectrogramme d'amplitude du canal 1 comme entrée:

$$\widehat{\mathcal{M}} = \mathcal{F}(|\mathbf{X}_1|). \tag{7.19}$$

La manière la plus naturelle d'utiliser DeepSHAP est de supposer que chaque point temps-fréquence $|\,x_1(n', f')|$ de l'entrée est un descripteur et de calculer la contribution de ce descripteur à chaque point temps-fréquence du masque $\widehat{\mathcal{M}}(n, f)$ estimé. Nous obtenons donc $N \times F$ matrices de valeurs SHAP $\mathbf{\Phi}^{\mathrm{TF}}(n, f)$ de taille $N \times F$ chacune. Afin de réduire le nombre de matrices à calculer et à analyser, une alternative consiste à sommer les valeurs de pertinence par trame:

$$\mathbf{\Phi}^{\mathrm{T}}(n) = \sum_f \mathbf{\Phi}^{\mathrm{TF}}(n, f). \tag{7.20}$$

Cela ne nécessite pas le calcul de chaque $\mathbf{\Phi}^{\mathrm{TF}}(n, f)$: les valeurs SHAP sont additionnées au niveau de la couche de sortie et une seule rétro-propagation vers les entrées est effectuée. Les F matrices $\mathbf{\Phi}^{\mathrm{T}}(n)$ ainsi obtenues sont également des matrices de taille $N \times F$, indiquant la pertinence pour chaque trame du masque estimé. De même, les valeurs de pertinence peuvent également être sommées sur l'ensemble du signal:

$$\mathbf{\Phi}^{\mathrm{U}} = \sum_n \mathbf{\Phi}^{\mathrm{T}}(n). \tag{7.21}$$

Notons que

$$\sum_{n,f} \mathbf{\Phi}^{\mathrm{TF}}(n, f) = \sum_n \mathbf{\Phi}^{\mathrm{T}}(n) = \mathbf{\Phi}^{\mathrm{U}}\,. \tag{7.22}$$

La Fig. 7.5 montre les valeurs SHAP obtenues pour des valeurs spécifiques de n et de f. La matrice $\mathbf{\Phi}^{\mathrm{TF}}(n, f)$ fournit la granularité de pertinence la plus fine tandis que $\mathbf{\Phi}^{\mathrm{U}}$ fournit la granularité la plus grossière. Nous avons constaté que les cartes de pertinence $\mathbf{\Phi}^{\mathrm{TF}}(n, f)$ obtenues pour différentes bandes de fréquence f au sein d'une même trame n (non montrées ici) sont similaires les unes aux autres. Nous choisissons donc $\mathbf{\Phi}^{\mathrm{T}}(n)$ pour notre analyse.

7.3.3 Métrique de pertinence

Le but d'un modèle de rehaussement de la parole est de supprimer les points temps-fréquence associés au bruit tout en conservant ceux associés à la parole. Un bon modèle de rehaussement, avec une bonne capacité de généralisation, devrait donc principalement se baser sur les points temps-fréquence correspondant à la parole. Cela est particulièrement

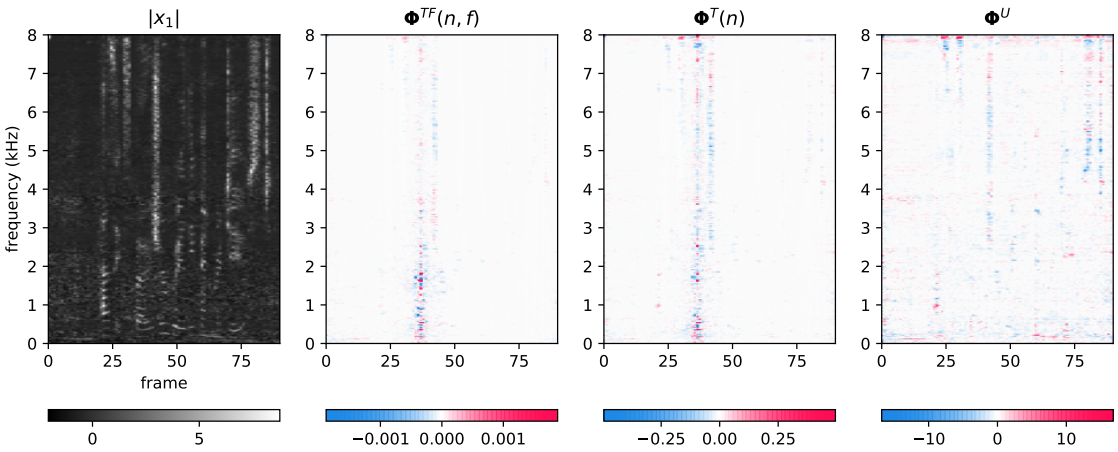


Figure 7.5: Exemple de valeurs SHAP calculées pour un mélange parole-bruit avec $n = 36$ et $f = 1635$ Hz. Le spectrogramme d'entrée a des valeurs négatives car sa moyenne et sa variance sont normalisées par phrase avant de l'utiliser comme entrée du réseau.

vrai lors de l'utilisation du modèle dans des conditions de bruit inconnues et différentes de celles de l'apprentissage. Nous proposons donc d'utiliser la métrique de pertinence globale suivante pour résumer les valeurs SHAP:

$$\eta = \frac{\sum_{n \in \text{speech}} \#\{\phi_{>T+\text{IBM}}(n)\}}{\sum_{n \in \text{speech}} \#\{\phi_{>T}(n)\}}, \tag{7.23}$$

où $\#\{\phi_{>T}(n)\}$ représente le nombre de points temps-fréquence dans $\mathbf{\Phi}^{\text{T}}(n)$ dont la valeur absolue est supérieure à un seuil T et $\#\{\phi_{>T+\text{IBM}}(n)\}$ représente le nombre de ces points associés à la parole dans un masque binaire idéal. La métrique est calculée uniquement pour les trames contenant de la parole. Le seuil T représente le T-ième centile des valeurs obtenues.

7.3.4 Protocole expérimental

Données Les expériences sont menées sur le corpus CHiME-4 (Barker et al., 2017), qui se compose de phrases de Wall Street Journal (WSJ0) prononcées par des locuteurs dans des environnements bruyants et enregistrées à l'aide d'une antenne de 6 microphones sur une tablette. Le corpus d'origine comporte quatre catégories d'environnements: bus, café, zone piétonne et croisement de rue. Il est livré avec un outil de simulation de données, qui mélange des phrases WSJ0 non réverbérés originales au bruit de fond, garantissant la même distribution de rapport signal-à-bruit que dans les enregistrements réels.
Afin d'apprendre le réseau de rehaussement, nous générons trois ensembles d'apprentissage et de validation différents correspondant aux trois conditions de bruit suivantes:

1. CHiME: enregistrements de bruit réel du corpus CHiME-4,
2. SSN: bruit SSN généré artificiellement

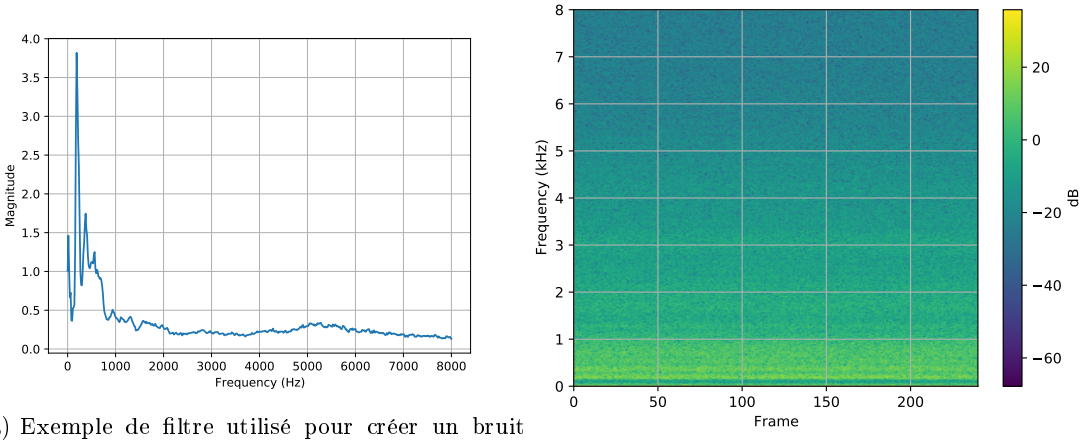
3. Network: fichiers audio de la bibliothèque Network Sound Effects[2] utilisée par Pandey and Wang (2019), contenant des sons de différentes catégories: musique, nature, ferroviaire, etc.

Chacun des trois ensembles contient 7 138 phrases d'apprentissage et 1 640 de validation. La distribution des rapports signal-à-bruit est la même. Seul le premier canal est utilisé pour l'apprentissage du réseau de rehaussement.

Génération du bruit SSN Le bruit SSN est une forme de bruit artificiel généré en filtrant un bruit blanc par un filtre dont la réponse correspond au spectre de la parole à long terme. Le filtre est obtenu en calculant la moyenne du spectre d'amplitude de plusieurs signaux de parole propre:

$$\text{filtre}(f) = \frac{1}{N_s} \sum_{k=1}^{N_s} \frac{1}{N_k} \sum_{n=1}^{N_k} |s(k, n, f)| \tag{7.24}$$

où N_s est le nombre total de signaux et N_k est la longueur du k-ième signal.



(a) Exemple de filtre utilisé pour créer un bruit SSN.

(b) Spectrogramme d'un bruit SSN.

Figure 7.6: Caractéristiques du bruit SSN.

Dans ce travail, pour chaque mélange à générer, nous prenons $N_s = 6$ autres signaux du corpus CHiME-4 pour créer le filtre. La réponse en amplitude d'un filtre et le spectrogramme du signal de bruit obtenu sont représentés sur les Fig. 7.6a et 7.6b. L'intuition derrière l'utilisation du SSN est qu'il a une haute énergie dans des bandes de fréquence généralement dominées par la parole, ce qui complique la tâche du réseau et mène à des frontières de décision plus précises.

Architectures de réseaux pour le rehaussement Les réseaux de rehaussement apprennent à estimer le masque d'amplitude en utilisant comme entrée le spectrogramme

[2] https://www.sound-ideas.com/Product/199/Network-Sound-Effects-Library

d'amplitude du mélange x_5 (c'est-à-dire le canal 5 du signal multicanal). Le spectrogramme est calculé sur des fenêtres de 50 ms avec un décalage de 25 ms, ce qui correspond à une dimension d'entrée du réseau de 401. L'architecture du réseau comporte deux couches Bi-LSTM suivies d'une normalisation par couche (Ba et al., 2016) et d'une couche linéaire de 401 neurones. La sortie est contrainte à l'intervalle $[0-1]$ par une non-linéarité sigmoïde. Les modèles appris pour les trois conditions de bruit sont désignés par $\mathcal{F}_{\text{CHIME}}$, $\mathcal{F}_{\text{SSN}}$ et $\mathcal{F}_{\text{NETWORK}}$.

Évaluation Nous évaluons la performance de reconnaissance automatique de la parole obtenue après rehaussement sur l'ensemble d'évaluation réel (`et05_real`) du corpus CHiME-4 d'origine. Étant donné que le masque oracle est nécessaire pour calculer le score de pertinence global, nous utilisons également l'ensemble de développement simulé (`dt05_simu`) dans nos expériences. Le canal 5 des enregistrements est utilisé comme entrée des réseaux pour l'estimation du masque. Le masque estimé et les signaux de tous les canaux (à l'exception du canal 2) sont utilisés pour calculer un filtre de Wiener multicanal contraint de rang 1 (Wang et al., 2018b) qui est ensuite utilisé pour le rehaussement. Le système de base de reconnaissance automatique de la parole fourni dans le cadre du défi CHiME-4 a été utilisé pour évaluer la qualité de la parole rehaussée. Ce système suit la recette `nnet1` de la boîte à outils Kaldi (Povey et al., 2011), avec un modèle acoustique MLP à 7 couches et un modèle de langage trigramme. Il a été appris sur des mélanges parole-bruit réels et simulés.

DeepSHAP DeepSHAP nécessite le calcul de l'espérance du vecteur d'entrée à chaque couche. Des échantillons pour calculer cette espérance sont tirés de l'ensemble de données associé au mélange $\mathbf{x}$ considéré. Par exemple, si $\mathbf{x}$ provient d'`et05_real`, alors une partie de `et05_real` est conservée en tant qu'échantillons pour calculer cette espérance. Par la suite, le nombre d'échantillons est fixé à 40. L'implémentation DeepSHAP de `https://github.com/slundberg/shap` est utilisée pour calculer les valeurs SHAP. La valeur seuil de $T = 99,9$ est utilisée partout, sauf indication contraire. Une valeur élevée de T entraîne un nombre inférieur de points temps-fréquence pour calculer η. Mais comme nous calculons les valeurs SHAP par trame, c'est-à-dire $\mathbf{\Phi}^T(n)$, la pertinence doit être limitée à un petit voisinage autour de n et non répartie sur toute la séquence.

7.3.5 Résultats

ASR Le tableau 7.3 montre le WER obtenu sur l'ensemble `et05_real` en utilisant les différents modèles de rehaussement. Un WER de base de $25,9\%$ a été obtenu avant rehaussement. Le WER s'améliore à $11,7\%$ après rehaussement par le modèle $\mathcal{F}_{\text{CHIME}}$. Les WERs obtenus en utilisant les modèles appris avec les bruits SSN et Network sont respectivement de $14,0\%$ et $15,1\%$. La meilleure performance du modèle appris à l'aide du bruit CHiME peut être attribuée à la similarité des conditions d'apprentissage et d'évaluation. Néanmoins, les résultats obtenus en utilisant des modèles appris avec les bruits SSN et Network sont nettement meilleurs que sans rehaussement, ce qui montre

Bruit d'apprentissage	Signal rehaussé	`et05_real` (%)	`dt05_simu`(%)
Mélange (non rehaussé)	x_5	25.9	12.7
CHiME	$\mathcal{F}_{\text{CHIME}}(x_5)$	11.7	6.7
SSN	$\mathcal{F}_{\text{SSN}}(x_5)$	14.0	7.3
Network	$\mathcal{F}_{\text{NETWORK}}(x_5)$	15.1	7.7

Table 7.3: WER (%) sur l'ensemble d'évaluation réel (`et05_real`) et l'ensemble de développement simulé (`dt05_simu`) de CHiME-4.

Modèle	$T = 99,9$	$T = 99,0$	$T = 98,0$
$\mathcal{F}_{\text{CHIME}}$	94,8	92,2	90,5
$\mathcal{F}_{\text{SSN}}$	89,6	87,2	85,4
$\mathcal{F}_{\text{NETWORK}}$	90,3	89,5	88,7

Table 7.4: Valeurs moyennes de la métrique de pertinence η (%) en fonction du seuil sur l'ensemble `dt05_simu`.

l'utilité de ces bruits pour l'apprentissage. Des gains de WER similaires peuvent être observés sur l'ensemble `dt05_simu`.

Métrique de pertinence Le tableau 7.4 montre les valeurs de la métrique de pertinence globale η obtenues sur `dt05_simu` pour tous les modèles de rehaussement avec des valeurs de seuil différentes. Les résultats sont obtenus en utilisant un total de 300 signaux. Une valeur $\eta = 94,8\%$ a été obtenue en utilisant $\mathcal{F}_{\text{CHIME}}$, ce qui signifie que pour un seuil $T = 99,9$, $94,8\%$ des points temps-fréquence du spectrogramme d'entrée qui ont été utilisés pour expliquer le masque de sortie étaient dominés par la parole. Les autres valeurs suivent les tendances de WER observées dans le tableau 7.3. De meilleures valeurs η sont observées pour le modèle $\mathcal{F}_{\text{CHIME}}$, qui a donné le meilleur WER. La différence négligeable entre les valeurs η pour $\mathcal{F}_{\text{SSN}}$ et $\mathcal{F}_{\text{NETWORK}}$ reflète la différence dans le WER des modèles correspondants sur `dt05_simu`, mais en faveur de $\mathcal{F}_{\text{NETWORK}}$. La valeur η varie selon le seuil T, indiquant que les points temps-fréquence avec des valeurs SHAP inférieures ne sont pas dominés par la parole. Nous pouvons donc conclure que $\mathcal{F}_{\text{CHIME}}$ fonctionne mieux que les autres modèles car il s'appuie sur les points temps-fréquence dominés par la parole pour estimer le masque.

7.3.6 Capacité de généralisation des modèles

Pour mieux comprendre la capacité de généralisation des modèles, nous calculons les valeurs SHAP pour plusieurs signaux d'apprentissage et de test partageant le même signal de parole sous-jacent. Nous appelons ces deux configurations apprentissage et test. Dans la configuration d'apprentissage, les valeurs SHAP sont calculées pour chaque modèle en mélangeant le signal de parole avec du bruit du même type que celui utilisé pour apprendre le modèle. Dans la configuration de test, les valeurs SHAP sont calculées

Configuration	$\mathcal{F}_{\text{NETWORK}}$ (%)	$\mathcal{F}_{\text{SSN}}$ (%)
Apprentissage	82,5	81,7
Test	58,6	74,4

Table 7.5: Valeur moyenne η obtenue sur 28 phrases sélectionnés aléatoirement pour les configurations d'apprentissage et de test. La valeur η pour $\mathcal{F}_{\text{CHIME}}$ était 81,7%.

pour tous les modèles à l'exception de $\mathcal{F}_{\text{CHIME}}$ en mélangeant le signal de parole avec du bruit CHiME. Pour un modèle qui se généralise bien, les points temps-fréquence pertinents devraient être les mêmes dans les configurations d'apprentissage et de test. Le tableau 7.5 montre les scores η calculés pour les configurations d'apprentissage et de test sur un plus grand nombre de signaux. On peut observer qu'η est plus élevé dans la configuration d'apprentissage pour $\mathcal{F}_{\text{NETWORK}}$ que pour $\mathcal{F}_{\text{SSN}}$ mais il est plus faible dans la configuration de test, ce qui indique que $\mathcal{F}_{\text{SSN}}$ a une meilleure capacité de généralisation que $\mathcal{F}_{\text{NETWORK}}$.

7.3.7 Conclusion

Dans cette partie, nous avons abordé le problème de l'explication d'un modèle de rehaussement. DeepSHAP, une méthode d'attribution aux caractéristiques, est utilisée pour déterminer les points temps-fréquence du spectrogramme d'entrée qui sont utilisés par le réseau pour estimer le masque. En partant de l'idée qu'un bon modèle devrait se baser sur les points temps-fréquence dominés par la parole plutôt que ceux dominés par le bruit, nous avons proposé une métrique de pertinence globale afin d'analyser les valeurs SHAP obtenues. Nous avons montré que les modèles de rehaussement ayant une valeur de pertinence globale supérieure donnent de meilleures performances de reconnaissance automatique de la parole. Nous avons également montré que la capacité de généralisation d'un modèle de rehaussement entraîné à l'aide de bruit artificiel SSN est meilleure que celle d'un modèle de rehaussement appris avec les bruits Network.

Bibliography

Abdulaziz, A. and Kepuska, V. (2017). Noisy TIMIT speech. *Linguistic Data Consortium V1 , http://hdl.handle.net/11272/UFA9N*.

Adams, 1952-2001, D. (1980). *The Hitchhiker's Guide to the Galaxy*. Harmony Books.

Adavanne, S., Politis, A., Nikunen, J., and Virtanen, T. (2019). Sound event localization and detection of overlapping sources using convolutional recurrent neural networks. *IEEE Journal of Selected Topics in Signal Processing*, 13(1):34–48.

Adavanne, S., Politis, A., and Virtanen, T. (2018). Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network. In *European Signal Processing Conference*, pages 1462–1466.

Alameda-Pineda, X. and Horaud, R. (2015). Vision-guided robot hearing. *The International Journal of Robotics Research*, 34(4-5):437–456.

Alber, M., Lapuschkin, S., Seegerer, P., Hägele, M., Schütt, K. T., Montavon, G., Samek, W., Müller, K.-R., Dähne, S., and Kindermans, P.-J. (2019). iNNvestigate neural networks. *Journal of Machine Learning Research*, 20(93):1–8.

Allen, J. B. and Berkley, D. A. (1979). Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America*, 65(4):943–950.

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., Nakatani, T., Sawada, H., and Makino, S. (2009). Stereo source separation and source counting with MAP estimation with Dirichlet prior considering spatial aliasing problem. In *5th International Conference on Independent Component Analysis and Signal Separation*, pages 742–750.

Asaei, A., Bourlard, H., Taghizadeh, M. J., and Cevher, V. (2016). Computational methods for underdetermined convolutive speech localization and separation via model-based sparse component analysis. *Speech Communication*, 76:201–217.

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. In *Advances in Neural Information Processing Systems*.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):e0130140.

Bahl, L., Brown, P., de Souza, P., and Mercer, R. (1986). Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 11, pages 49–52.

Barfuss, H. and Kellermann, W. (2016). On the impact of localization errors on HRTF-based robust least-squares beamforming. In *German Annual Conference on Acoustics*, pages 1072–1075.

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., 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.

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., and Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115.

Becker, S., Ackermann, M., Lapuschkin, S., Müller, K.-R., and Samek, W. (2018). Interpreting and explaining deep neural networks for classification of audio signals. *arXiv:1807.03418 [cs, eess]*.

Bharadhwaj, H. (2018). Layer-wise relevance propagation for explainable deep learning based speech recognition. In *IEEE International Symposium on Signal Processing and Information Technology*, pages 168–174.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Boeddeker, C., Hanebrink, P., Drude, L., Heymann, J., and Haeb-Umbach, R. (2017). Optimizing neural-network supported acoustic beamforming by algorithmic differentiation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 171–175.

Bregman, A. S. (1994). *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press.

Bronkhorst, A. W. (2000). The cocktail party phenomenon: A review of research on speech intelligibility in multiple-talker conditions. *Acta Acustica united with Acustica*, 86(1):117–128.

Brutti, A. and Matassoni, M. (2016). On the relationship between early-to-late ratio of room impulse responses and ASR performance in reverberant environments. *Speech Communication*, 76:170–185.

Carter, G., Nuttall, A., and Cable, P. (1973). The smoothed coherence transform. *Proceedings of the IEEE*, 61(10):1497–1498.

Chakrabarty, S. and Habets, E. A. P. (2017a). Broadband DOA estimation using convolutional neural networks trained with noise signals. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 136–140.

Chakrabarty, S. and Habets, E. A. P. (2017b). Multi-speaker localization using convolutional neural network trained with noise. In *NIPS 2017 Workshop on Machine Learning for Audio Processing*.

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*, pages 246–250.

Chen, Z., Watanabe, S., Erdogan, H., and Hershey, J. R. (2015). Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks. In *Interspeech*, pages 3274–3278.

Chen, Z., Xiao, X., Yoshioka, T., Erdogan, H., Li, J., and Gong, Y. (2018a). Multi-channel overlapped speech recognition with location guided speech extraction network. In *IEEE Spoken Language Technology Workshop*, pages 558–565.

Chen, Z., Yoshioka, T., Xiao, X., Li, J., Seltzer, M. L., and Gong, Y. (2018b). Efficient integration of fixed beamformers and speech separation networks for multi-channel far-field speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5384–5388.

Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y. (2014a). On the properties of neural machine translation: Encoder–decoder approaches. In *8th Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014b). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734.

Cox, H., Zeskind, R., and Owen, M. (1987). Robust adaptive beamforming. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(10):1365–1376.

Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366.

Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., and Ouellet, P. (2011). Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798.

Deleforge, A., Di Carlo, D., Strauss, M., Serizel, R., and Marcenaro, L. (2019a). Audio-based search and rescue with a drone: Highlights from the IEEE Signal Processing Cup 2019 student competition. *IEEE Signal Processing Magazine*, 36(5):138–144.

Deleforge, A., Forbes, F., and Horaud, R. (2014). Acoustic space learning for sound-source separation and localization on binaural manifolds. *International Journal of Neural Systems*, 25(01):1440003.

Deleforge, A., Horaud, R., Schechner, Y. Y., and Girin, L. (2015). Co-localization of audio sources in images using binaural features and locally-linear regression. *IEEE Transactions on Audio, Speech, and Language Processing*, 23(4):718–731.

Deleforge, A., Schmidt, A., and Kellermann, W. (2019b). Audio-motor integration for robot audition. In *Multimodal Behavior Analysis in the Wild*, pages 27–51.

Delfosse, N. and Loubaton, P. (1995). Adaptive blind separation of independent sources: A deflation approach. *Signal Processing*, 45(1):59–83.

Deng, L. and Yu, D. (2014). *Deep Learning: Methods and Applications*. NOW Publishers.

Dibiase, J. H. (2000). *A High-Accuracy, Low-Latency Technique for Talker Localization in Reverberant Environments Using Microphone Arrays*. PhD thesis, Brown University.

Doclo, S. and Moonen, M. (2002). GSVD-based optimal filtering for single and multimicrophone speech enhancement. *IEEE Transactions on Signal Processing*, 50(9):2230–2244.

Doclo, S., Spriet, A., Wouters, J., and Moonen, M. (2007). Frequency-domain criterion for the speech distortion weighted multichannel Wiener filter for robust noise reduction. *Speech Communication*, 49(7):636–656.

Drude, L., Heitkaemper, J., Boeddeker, C., and Haeb-Umbach, R. (2019). SMS-WSJ: Database, performance measures, and baseline recipe for multi-channel source separation and recognition. *arXiv:1910.13934 [cs, eess]*.

Du, J., Gao, T., Sun, L., Ma, F., Wang, H.-K., Pan, J., Liu, C., Chen, J.-D., and Lee, C.-H. (2016). The USTC-iFlytek system for CHiME-4 challenge. In *4th International Workshop on Speech Processing in Everyday Environments*, pages 36–38.

Duong, N. Q. K., Vincent, E., and Gribonval, R. (2010). Under-determined reverberant audio source separation using a full-rank spatial covariance model. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(7):1830–1840.

Duong, N. Q. K., Vincent, E., and Gribonval, R. (2013). Spatial location priors for Gaussian model based reverberant audio source separation. *EURASIP Journal on Advances in Signal Processing*, 2013(1):149.

El Badawy, D., Dokmanić, I., and Vetterli, M. (2017). Acoustic DoA estimation by one unsophisticated sensor. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 89–98.

Ephrat, A., Mosseri, I., Lang, O., Dekel, T., Wilson, K., Hassidim, A., Freeman, W. T., and Rubinstein, M. (2018). Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *ACM Transactions on Graphics*, 37(4):112:1–112:11.

Erdogan, H., Hershey, J. R., Watanabe, S., and Le Roux, J. (2015). Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 708–712.

Evers, C., Loellmann, H., Mellmann, H., Schmidt, A., Barfuss, H., Naylor, P., and Kellermann, W. (2020). The LOCATA challenge: Acoustic source localization and tracking. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1620–1643.

Eyring, C. F. (1930). Reverberation time in "dead" rooms. *The Journal of the Acoustical Society of America*, 1(2A):168–168.

Gales, M. J. F. (1998). Maximum likelihood linear transformations for HMM based speech recognition. *Computer Speech & Language*, 12(2):75–98.

Galliano, S., Geoffrois, E., Gravier, G., Bonastre, J., Mostefa, D., and Choukri, K. (2006). Corpus description of the ESTER evaluation campaign for the rich transcription of French broadcast news. In *5th International Conference on Language Resources and Evaluation*, pages 139–142.

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.

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

Gao, Ruohan, Oh, T.-H., Grauman, K., and Torresani, L. (2020). Listen to look: Action recognition by previewing audio. In *Conference on Computer Vision and Pattern Recognition*.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

Gravier, G., Adda, G., Paulson, N., Carré, M., Giraudel, A., and Galibert, O. (2012). The ETAPE corpus for the evaluation of speech-based TV content processing in the French language. In *8th International Conference on Language Resources and Evaluation*, pages 114–118.

Greenberg, C. S., Mason, L. P., Sadjadi, S. O., and Reynolds, D. A. (2020). Two decades of speaker recognition evaluation at the national institute of standards and technology. *Computer Speech & Language*, 60:101032.

Gretsistas, A. and Plumbley, M. D. (2010). A multichannel spatial compressed sensing approach for direction of arrival estimation. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 458–465.

Griffin, A., Pavlidi, D., Puigt, M., and Mouchtaris, A. (2012). Real-time multiple speaker DOA estimation in a circular microphone array based on matching pursuit. In *European Signal Processing Conference*, pages 2303–2307.

Griffin, D. and Lim, J. (1984). Signal estimation from modified short-time Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243.

Griffiths, L. and Jim, C. (1982). An alternative approach to linearly constrained adaptive beamforming. *IEEE Transactions on Antennas and Propagation*, 30(1):27–34.

Gustafsson, T., Rao, B., and Trivedi, M. (2003). Source localization in reverberant environments: Modeling and statistical analysis. *IEEE Transactions on Speech and Audio Processing*, 11(6):791–803.

Habets, E. A. P. (2018). RIR-Generator: Room impulse response generator. https://github.com/ehabets/RIR-Generator.

Habets, E. A. P. and Naylor, P. A. (2018). Dereverberation. In *Audio Source Separation and Speech Enhancement*, pages 317–343. Wiley.

Hansen, J. H. and Hasan, T. (2015). Speaker recognition by machines and humans: A tutorial review. *IEEE Signal Processing Magazine*, 32(6):74–99.

Hänsler, E. and Schmidt, G. (2005). *Acoustic Echo and Noise Control: A Practical Approach*. Wiley.

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

Hazrati, O. and Loizou, P. C. (2012). The combined effects of reverberation and noise on speech intelligibility by cochlear implant listeners. *International Journal of Audiology*, 51(6):437–443.

He, W., Motlicek, P., and Odobez, J.-M. (2018a). Deep neural networks for multiple speaker detection and localization. In *IEEE International Conference on Robotics and Automation*, pages 74–79.

He, W., Motlicek, P., and Odobez, J.-M. (2018b). Joint localization and classification of multiple sound sources using a multi-task neural network. In *Interspeech*, pages 312–316.

Heitkaemper, J., Jakobeit, D., Boeddeker, C., Drude, L., and Haeb-Umbach, R. (2020). Demystifying TasNet: A dissecting approach. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6359–6363.

Herbordt, W., Buchner, H., Nakamura, S., and Kellermann, W. (2005). Outlier-robust DFT-domain adaptive filtering for bin-wise stepsize controls, and its application to a generalized sidelobe canceller. In *International Workshop on Acoustic Echo and Noise Control*, pages 113–116.

Herbordt, W. and Kellermann, W. (2003). Adaptive beamforming for audio signal acquisition. In *Adaptive Signal Processing: Applications to Real-World Problems*, pages 155–194.

Herbordt, W., Trini, T., and Kellermann, W. (2003). Robust spatial estimation of the signal-to-interference ratio for non-stationary mixtures. In *International Workshop on Acoustic Echo and Noise Control*, pages 247–250.

Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. *The Journal of the Acoustical Society of America*, 87(4):1738–1752.

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*, pages 31–35.

Heymann, J., Drude, L., Boeddeker, C., Hanebrink, P., and Haeb-Umbach, R. (2017). Beamnet: End-to-end training of a beamformer-supported multi-channel ASR system. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5325–5329.

Heymann, J., Drude, L., Chinaev, A., and Haeb-Umbach, R. (2015). BLSTM supported GEV beamformer front-end for the 3rd CHiME challenge. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 444–451.

Higuchi, T., Ito, N., Yoshioka, T., and Nakatani, T. (2016). Robust MVDR beamforming using time-frequency masks for online/offline ASR in noise. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5210–5214.

Higuchi, T., Kinoshita, K., Delcroix, M., Žmolíková, K., and Nakatani, T. (2017). Deep clustering-based beamforming for separation with unknown number of sources. In *Interspeech*, pages 1183–1187.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., and Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Hoshen, Y., Weiss, R. J., and Wilson, K. W. (2015). Speech acoustic modeling from raw multichannel waveforms. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4624–4628.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *32nd International Conference on Machine Learning*, volume 37, pages 448–456.

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.

Kayser, H. and Anemüller, J. (2014). A discriminative learning approach to probabilistic acoustic source localization. In *International Workshop on Acoustic Signal Enhancement*, pages 99–103.

Kim, C. and Stern, R. M. (2016). Power-normalized cepstral coefficients (PNCC) for robust speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(7):1315–1329.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd International Conference for Learning Representations*.

Kinoshita, K., Delcroix, M., Gannot, S., Habets, E. A. P., Haeb-Umbach, R., Kellermann, W., Leutnant, V., Maas, R., Nakatani, T., Raj, B., and et al. (2016). A summary of the REVERB challenge: State-of-the-Art and remaining challenges in reverberant speech processing research. *EURASIP Journal on Advances in Signal Processing*, 2016(1):1–19.

Kinoshita, K., Drude, L., Delcroix, M., and Nakatani, T. (2018). Listening to each speaker one by one with recurrent selective hearing networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5064–5068.

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.

Koehn, P. (2020). *Neural Machine Translation*. Cambridge University Press.

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.

Kuttruff, H. (2016). *Room Acoustics*. CRC Press.

Lam, C. J. and Singer, A. C. (2006). Bayesian beamforming for DOA uncertainty: Theory and implementation. *IEEE Transactions on Signal Processing*, 54(11):4435–4445.

Laufer, B., Talmon, R., and Gannot, S. (2013). Relative transfer function modeling for supervised source localization. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 1–4.

Laufer, B., Talmon, R., and Gannot, S. (2016). Semi-supervised sound source localization based on manifold regularization. *IEEE/ACM Transaction on Audio, Speech and Language*, 24(8):1393–1407.

Le Roux, J., Hershey, J. R., and Weninger, F. (2015). Deep NMF for speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 66–70.

Le Roux, J., Wichern, G., Watanabe, S., Sarroff, A., and Hershey, J. R. (2019a). Phasebook and friends: Leveraging discrete representations for source separation. *IEEE Journal of Selected Topics in Signal Processing*, 13:370–382.

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

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

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

Li, F., Nidadavolu, P. S., and Hermansky, H. (2014). A long, deep and wide artificial neural net for robust speech recognition in unknown noise. In *Interspeech*, pages 358–362.

Li, J., Deng, L., Haeb-Umbach, R., and Gong, Y. (2015). *Robust Automatic Speech Recognition: A Bridge to Practical Applications*. Academic Press.

Li, X., Ban, Y., Girin, L., Alameda-Pineda, X., and Horaud, R. (2019). Online localization and tracking of multiple moving speakers in reverberant environments. *IEEE Journal of Selected Topics in Signal Processing*, 13(1):88–103.

Liu, H., Yin, Q., and Wang, W. Y. (2019). Towards explainable NLP: A generative explanation framework for text classification. In *57th Annual Meeting of the ACL*, pages 5570–5581.

Loizou, P. C. (2013). *Speech Enhancement: Theory and Practice*. CRC Press.

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., and Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1):56–67.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774.

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

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

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.

Ma, N., Brown, G. J., and May, T. (2015). Exploiting deep neural networks and head movements for binaural localisation of multiple speakers in reverberant conditions. In *Interspeech*, pages 3302–3306.

Maiwald, D. and Kraus, D. (2000). Calculation of moments of complex Wishart and complex inverse Wishart distributed matrices. *IEE Proceedings on Radar, Sonar and Navigation*, 147(4):162–168.

Malioutov, D., Cetin, M., and Willsky, A. (2005). A sparse signal reconstruction perspective for source localization with sensor arrays. *IEEE Transactions on Signal Processing*, 53(8):3010–3022.

Mandel, M. I. and Ellis, D. P. W. (2007). EM localization and separation using interaural level and phase cues. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 275–278.

Marchand, S. and Vialard, A. (2009). The Hough transform for binaural source localization. In *Digital Audio Effects Conference*, pages 252–259.

Martin, R. and Cohen, I. (2018). Single-channel speech presence probability estimation and noise tracking. In *Audio Source Separation and Speech Enhancement*, pages 87–106.

May, T., van de Par, S., and Kohlrausch, A. (2011). A probabilistic model for robust localization based on a binaural auditory front-end. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(1):1–13.

McDermott, J. H. (2009). The cocktail party problem. *Current Biology*, 19(22):R1024–R1027.

Menne, T., Heymann, J., Alexandridis, A., Irie, K., Zeyer, A., Kitza, M., Golik, P., Kulikov, I., Drude, L., and Schlüter, R. (2016). The RWTH/UPB/FORTH system combination for the 4th CHiME challenge evaluation. In *4th International Workshop on Speech Processing in Everyday Environments*, pages 39–44.

Menne, T., Schlüter, R., and Ney, H. (2019a). 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*, pages 6660–6664.

Menne, T., Sklyar, I., Schlüter, R., and Ney, H. (2019b). Analysis of deep clustering as preprocessing for automatic speech recognition of sparsely overlapping speech. In *Interspeech*, pages 2638–2642.

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

Molnar, C. (2019). *Interpretable Machine Learning - A Guide for Making Black Box Models Explainable*.

Mouba, J. and Marchand, S. (2006). A source localization/separation/respatialization system based on unsupervised classification of interaural cues. In *Digital Audio Effects Conference*, pages 233–238.

Muckenhirn, H., Abrol, V., Magimai.-Doss, M., and Marcel, S. (2019). Understanding and visualizing raw waveform-based CNNs. In *Interspeech*, pages 2345–2349.

Mysore, G. J. and Smaragdis, P. (2011). A non-negative approach to semi-supervised separation of speech from noise with the use of temporal dynamics. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 17–20.

Nakamura, K., Nakadai, K., and Okuno, H. G. (2013). A real-time super-resolution robot audition system that improves the robustness of simultaneous speech recognition. *Advanced Robotics*, 27(12):933–945.

Nguyen, Q., Girin, L., Bailly, G., Elisei, F., and Nguyen, D.-C. (2018). Autonomous sensorimotor learning for sound source localization by a humanoid robot. In *Workshop on Crossmodal Learning for Intelligent Robotics in Conjunction with IEEE/RSJ IROS*.

Nishino, T. and Takeda, K. (2008). Binaural sound localization for untrained directions based on a Gaussian mixture model. In *European Signal Processing Conference*, pages 1–5.

Nugraha, A. A. (2017). *Deep Neural Networks for Source Separation and Noise-Robust Speech Recognition*. PhD thesis, Université de Lorraine.

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.

Oppenheim, A. V. and Schafer, R. W. (2009). *Discrete-Time Signal Processing*. Prentice Hall, 3rd edition.

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*, pages 5206–5210.

Pandey, A. and Wang, D. (2019). A new framework for CNN-based speech enhancement in the time domain. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(7):1179–1188.

Parhizkar, R., Dokmanić, I., and Vetterli, M. (2014). Single-channel indoor microphone localization. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1434–1438.

Pariente, M., Cornell, S., Deleforge, A., and Vincent, E. (2020). Filterbank design for end-to-end speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6364–6368.

Pariente, M. and Pressnitzer, D. (2017). Predictive denoising of speech in noise using deep neural networks. *The Journal of the Acoustical Society of America*, 142(4):2611–2611.

Pavlidi, D., Griffin, A., Puigt, M., and Mouchtaris, A. (2013). Real-time multiple sound source localization and counting using a circular microphone array. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(10):2193–2206.

Peddinti, V., Povey, D., and Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts. In *Interspeech*, pages 2440–2444.

Perez, S. (2019). Report: Voice assistants in use to triple to 8 billion by 2023. http://social.techcrunch.com/2019/02/12/report-voice-assistants-in-use-to-triple-to-8-billion-by-2023/.

Perotin, L. (2019). *Localisation et Rehaussement de Sources de Parole au Format Ambisonique*. PhD thesis, Université de Lorraine.

Perotin, L., Défossez, A., Vincent, E., Serizel, R., and Guérin, A. (2019a). Regression versus classification for neural network based audio source localization. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 343–347.

Perotin, L., Serizel, R., Vincent, E., and Guérin, A. (2018). Multichannel speech separation with recurrent neural networks from high-order ambisonics recordings. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 36–40.

Perotin, L., Serizel, R., Vincent, E., and Guérin, A. (2019b). CRNN-based multiple DoA estimation using acoustic intensity features for ambisonics recordings. *IEEE Journal of Selected Topics in Signal Processing*, 13(1):22–33.

Povey, D. (2003). *Discriminative Training for Large Vocabulary Speech Recognition*. PhD thesis, University of Cambridge.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., and Schwarz, P. (2011). The Kaldi speech recognition toolkit. In *IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society.

Povey, D., Peddinti, V., Galvez, D., Ghahremani, P., Manohar, V., Na, X., Wang, Y., and Khudanpur, S. (2016). Purely sequence-trained neural networks for ASR based on lattice-free MMI. In *Interspeech*, pages 2751–2755.

Qian, K., Zhang, Y., Chang, S., Yang, X., Florencio, D., and Hasegawa-Johnson, M. (2018). Deep learning based speech beamforming. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5389–5393.

Rabiner, L. and Juang, B.-H. (1993). *Fundamentals of Speech Recognition*. Prentice Hall.

Rascon, C. and Meza, I. (2017). Localization of sound sources in robotics: A review. *Robotics and Autonomous Systems*, 96:184–210.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144.

Rickard, S. and Yılmaz, Ö. (2002). On the approximate W-disjoint orthogonality of speech. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 529–532.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408.

Roth, P. R. (1971). Effective measurements using digital signal analysis. *IEEE Spectrum*, 8(4):62–70.

Ryant, N., Church, K., Cieri, C., Cristia, A., Du, J., Ganapathy, S., and Liberman, M. (2019). The second DIHARD diarization challenge: Dataset, task, and baselines. In *Interspeech*, pages 978–982.

Sadeghi, M. and Alameda-Pineda, X. (2020). Robust unsupervised audio-visual speech enhancement using a mixture of variational autoencoders. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7534–7538.

Sadeghi, M., Leglaive, S., Alameda-Pineda, X., Girin, L., and Horaud, R. (to appear). Audio-visual speech enhancement using conditional variational auto-encoder. *IEEE/ACM Transactions on Audio, Speech and Language Processing*.

Sainath, T. N., Weiss, R. J., Wilson, K. W., Narayanan, A., Bacchiani, M., Li, B., Variani, E., Shafran, I., Senior, A., Chin, K., Misra, A., and Kim, C. (2017). Raw multichannel processing using deep neural networks. In *New Era for Robust Speech Recognition: Exploiting Deep Learning*, pages 105–133.

Salvati, D., Drioli, C., and Foresti, G. L. (2018). Exploiting CNNs for improving acoustic source localization in noisy and reverberant conditions. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(2):103–116.

Sawada, H., Araki, S., Mukai, R., and Makino, S. (2007). Grouping separated frequency components by estimating propagation model parameters in frequency-domain blind source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(5):1592–1604.

Schmidt, R. (1986). Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, 34(3):276–280.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626.

Serizel, R., Moonen, M., Van Dijk, B., and Wouters, J. (2014). Low-rank approximation based multichannel Wiener filter algorithms for noise reduction with application in cochlear implants. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4):785–799.

Shapley, L. S. (1953). A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317.

Shi, Z., Lin, H., Liu, L., Liu, R., Hayakawa, S., Harada, S., and Han, J. (2019). FurcaNet: An end-to-end deep gated convolutional, long short-term memory, deep neural networks for single channel speech separation. *arXiv:1902.00651 [cs, eess]*.

Shrikumar, A., Greenside, P., and Kundaje, A. (2017a). Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153.

Shrikumar, A., Greenside, P., Shcherbina, A., and Kundaje, A. (2017b). Not just a black box: Learning important features through propagating activation differences. *arXiv:1605.01713 [cs]*.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR Workshop*.

Sivasankaran, S., Nugraha, A. A., Vincent, E., Morales Cordovilla, J. A., Dalmia, S., Illina, I., and Liutkus, A. (2015). Robust ASR using neural network based speech enhancement and feature simulation. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 482–489.

Smaragdis, P. (2007). Convolutive speech bases and their application to supervised speech separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(1):1–12.

Smaragdis, P. and Brown, J. (2003). Non-negative matrix factorization for polyphonic music transcription. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 177–180.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. (2017). SmoothGrad: Removing noise by adding noise. In *ICML Workshop on Visualization for Deep Learning*.

Souden, M., Araki, S., Kinoshita, K., Nakatani, T., and Sawada, H. (2013). A multichannel MMSE-based framework for speech source separation and noise reduction. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(9):1913–1928.

Souden, M., Benesty, J., and Affes, S. (2010). On optimal frequency-domain multichannel linear filtering for noise reduction. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(2):260–276.

Spriet, A., Moonen, M., and Wouters, J. (2004). Spatially pre-processed speech distortion weighted multi-channel Wiener filtering for noise reduction. *Signal Processing*, 84(12):2367–2387.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. (2015). Striving for simplicity: The all convolutional net. In *ICLR Workshop*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328.

Swietojanski, P., Li, J., and Renals, S. (2016). Learning hidden unit contributions for unsupervised acoustic model adaptation. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 24(8):1450–1463.

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

Takeda, R. and Komatani, K. (2016). Discriminative multiple sound source localization based on deep neural networks using independent location model. In *IEEE Spoken Language Technology Workshop*, pages 603–609.

Talmon, R., Cohen, I., and Gannot, S. (2011). Supervised source localization using diffusion kernels. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 245–248.

Talmon, R., Kushnir, D., Coifman, R. R., Cohen, I., and Gannot, S. (2012). Parametrization of linear systems using diffusion kernels. *IEEE Transactions on Signal Processing*, 60(3):1159–1173.

Taseska, M. and Habets, E. A. P. (2013). MMSE-based source extraction using position-based posterior probabilities. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 664–668.

Taseska, M. and Habets, E. A. P. (2016). Spotforming: Spatial filtering with distributed arrays for position-selective sound acquisition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(7):1291–1304.

Taseska, M. and Habets, E. A. P. (2017). DOA-informed source extraction in the presence of competing talkers and background noise. *EURASIP Journal on Advances in Signal Processing*, 2017(1):60.

Tashev, I. J., Le, L., Gopalakrishna, V., and Lovitt, A. (2017). Cost function for sound source localization with arbitrary microphone arrays. In *Joint Workshop on Hands-Free Speech Communications and Microphone Arrays*, pages 76–80.

Taylor, P. (2009). *Text-to-Speech Synthesis*. Cambridge University Press, 1st edition.

Tur, G. (2011). *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. Wiley.

Vacher, M., Vincent, E., Bobillier Chaumon, M.-E., Joubert, T., Portet, F., Fohr, D., Caffiau, S., and Desot, T. (2018). The VocADom project: Speech interaction for well-being and reliance improvement. In *20th International Conference on Human-Computer Interaction with Mobile Devices and Services*.

Valentini-Botinhao, C., Wang, X., Takaki, S., and Yamagishi, J. (2016). Speech enhancement for a noise-robust text-to-speech synthesis system using deep recurrent neural networks. In *Interspeech*, pages 352–356.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. In *9th ISCA Speech Synthesis Workshop*, pages 125–125.

Van Trees, H. L. (2002). *Optimum Array Processing*. Wiley.

Vecchiotti, P., Ma, N., Squartini, S., and Brown, G. J. (2019). End-to-end binaural sound localisation from the raw waveform. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 451–455.

Veselý, K., Ghoshal, A., Burget, L., and Povey, D. (2013). Sequence-discriminative training of deep neural networks. In *Interspeech*, pages 2345–2349.

Vesperini, F., Vecchiotti, P., Principi, E., Squartini, S., and Piazza, F. (2016). A neural network based algorithm for speaker localization in a multi-room environment. In *IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6.

Vijayasenan, D. and Valente, F. (2012). Speaker diarization of meetings based on large TDOA feature vectors. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4173–4176.

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., Jafari, M. G., Abdallah, S. A., Plumbley, M. D., and Davies, M. E. (2011). Probabilistic modeling paradigms for audio source separation. In *Machine Audition: Principles, Algorithms and Systems*, pages 162–185.

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

Virtanen, T., Singh, R., and Raj, B. (2012). *Techniques for Noise Robustness in Automatic Speech Recognition*. Wiley.

Wang, Z., Li, J., Yan, Y., and Vincent, E. (2018a). Semi-supervised learning with deep neural networks for relative transfer function inverse regression. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 191–195.

Wang, Z., Vincent, E., Serizel, R., and Yan, Y. (2018b). Rank-1 constrained multichannel Wiener filter for speech recognition in noisy environments. *Computer Speech & Language*, 49:37–51.

Wang, Z. and Wang, D. (2019). Combining spectral and spatial features for deep learning based blind speaker separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(2):457–468.

Wang, Z.-Q., Le Roux, J., and Hershey, J. R. (2018c). Alternative objective functions for deep clustering. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 686–690.

Wang, Z.-Q., Le Roux, J., and Hershey, J. R. (2018d). Multi-channel deep clustering: discriminative spectral and spatial embeddings for speaker-independent speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1–5.

Wang, Z.-Q., Le Roux, J., Wang, D., and Hershey, J. R. (2018e). End-to-end speech separation with unfolded iterative phase reconstruction. In *Interspeech*, pages 2708–2712.

Wang, Z.-Q., Tan, K., and Wang, D. (2019). Deep learning based phase reconstruction for speaker separation: A trigonometric perspective. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 71–75.

Warsitz, E. and Haeb-Umbach, R. (2007). Blind acoustic beamforming based on generalized eigenvalue decomposition. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(5):1529–1539.

Watanabe, S., Delcroix, M., Metze, F., and Hershey, J. R. (2017). *New Era for Robust Speech Recognition: Exploiting Deep Learning*. Springer.

Wichern, G., Antognini, J., Flynn, M., Zhu, L. R., McQuinn, E., Crow, D., Manilow, E., and Le Roux, J. (2019). WHAM!: Extending speech separation to noisy environments. In *Interspeech*, pages 1368–1372.

Wölfel, M. and McDonough, J. (2009). *Distant Speech Recognition*. Wiley.

Xiao, X., Watanabe, S., Erdogan, H., Lu, L., Hershey, J. R., Seltzer, M. L., Chen, G., Zhang, Y., Mandel, M., and Yu, D. (2016). Deep beamforming networks for multi-channel speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5745–5749.

Xiao, X., Zhao, S., Zhong, X., Jones, D. L., Chng, E. S., and Li, H. (2015). A learning-based approach to direction of arrival estimation in noisy and reverberant environments. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2814–2818.

Xiong, W., Droppo, J., Huang, X., Seide, F., Seltzer, M. L., Stolcke, A., Yu, D., and Zweig, G. (2017). Toward human parity in conversational speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(12):2410–2423.

Yılmaz, O. and Rickard, S. (2004). Blind separation of speech mixtures via time-frequency masking. *IEEE Transactions on Signal Processing*, 52(7):1830–1847.

Yin, J. and Chen, T. (2011). Direction-of-arrival estimation using a sparse representation of array covariance vectors. *IEEE Transactions on Signal Processing*, 59(9):4489–4493.

Yoshioka, T., Ito, N., Delcroix, M., Ogawa, A., Kinoshita, K., Fujimoto, M., Yu, C., Fabian, W. J., Espi, M., Higuchi, T., Araki, S., and Nakatani, T. (2015). The NTT CHiME-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 436–443.

Yoshioka, T., Sehr, A., Delcroix, M., Kinoshita, K., Maas, R., Nakatani, T., and Kellermann, W. (2012). Making machines understand us in reverberant rooms: Robustness against reverberation for automatic speech recognition. *IEEE Signal Processing Magazine*, 29(6):114–126.

Young, S. J., Odell, J. J., and Woodland, P. C. (1994). Tree-based state tying for high accuracy acoustic modelling. In *Workshop on Human Language Technology*, pages 307–312.

Yu, D., Kolbæk, M., Tan, Z., and Jensen, J. (2017). Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 241–245.

Yu, J., Wu, B., Gu, R., Zang, S.-X., Chen, L., Xu, Y., Yu, M., Su, D., Yu, D., Liu, X., and Meng, H. (2020). Audio-visual multi-channel recognition of overlapped speech. *arXiv:2005.08571 [eess.AS]*.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833.

Zhang, L., Shi, Z., Han, J., Shi, A., and Ma, D. (2020). FurcaNeXt: End-to-end monaural speech separation with dynamic gated dilated temporal convolutional networks. In *International Conference on Multimedia Modeling*, pages 653–665.

Zhang, W., Zhou, Y., and Qian, Y. (2019). Robust DOA estimation based on convolutional neural network and time-frequency masking. In *Interspeech*, pages 2703–2707.

Žmolíková, K., Delcroix, M., Kinoshita, K., Ochiai, T., Nakatani, T., Burget, L., and Černocký, J. (2019). SpeakerBeam: Speaker aware neural network for target speaker extraction in speech mixtures. *IEEE Journal of Selected Topics in Signal Processing*, 13(4):800–814.