BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

NEURAL TARGET SPEECH EXTRACTION

NEURÁLNÍ EXTRAKCE ŘEČI CÍLOVÉHO ŘEČNÍKA

PHD THESIS
DISERTAČNÍ PRÁCE

AUTHOR — **Ing. KATEŘINA ŽMOLÍKOVÁ**
AUTOR PRÁCE

SUPERVISOR — **prof. Dr. Ing. JAN ČERNOCKÝ**
ŠKOLITEL

BRNO 2021

Abstract

As speech processing technologies are getting increasingly more applied in the real world, their robustness has become a very important issue. Particularly, the processing of speech corrupted by interfering overlapping speakers is one of the challenging problems today. Speech separation approaches tackle this problem by separating the mixed speech into signals of individual speakers. These methods have made a big headway recently by leveraging the progress in deep learning.

In many applications, such as smartphones or digital home assistants, the goal is to enhance the speech signal of one speaker of interest, while suppressing other speakers and noise. In our work, we formulate this problem as target speech extraction and propose to solve it directly, i.e. to use a neural network with the enrollment speech and the mixture as inputs and the extracted speech of the target speaker as the output. We discuss and experimentally show the benefits of this approach compared to conventional speech separation: needlessness of counting speakers in the mixture, or better consistency of the output for longer recordings. We explore different aspects of the neural target speech extraction pipeline, namely the speaker embeddings, methods to inform the neural network about the target speaker, input and output domain, or loss function.

Furthermore, we demonstrate how to combine target speech extraction with multi-channel methods, such as neural mask-based beamforming and spatial clustering. Such combinations make use of both conventional statistical methods (for processing the spatial information) and strong modeling power of neural networks.

Finally, we apply target speech extraction as a pre-processing for two downstream tasks: automatic speech recognition, and clustering-based diarization. We investigate how to efficiently combine the front-end processing with the downstream systems, including joint optimization, or training with weakly supervised loss function based on speaker labels.

Keywords

target speech extraction, neural networks, multi-channel processing, multi-speaker automatic speech recognition, multi-speaker diarization

Reference

ŽMOLÍKOVÁ, Kateřina. *Neural target speech extraction.* Brno, 2021. PhD thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Jan Černocký.

Abstrakt

S rostoucím nasazením řečových technologií v praxi roste důležitost jejich robustnosti. Zejména zpracování řeči poškozené rušícími překrývajícími se řečníky je stále výzva. Přístupy separace řeči tento problém řeší rozkladem smíchané řeči na signály jednotlivých řečníků. Tyto metody v nedávné době výrazně pokročily s využitím vývoje v hlubokém učení.

Ve spoustě aplikací, jako jsou chytré telefony nebo digitální domácí asistenti, je cílem zvýraznit řečový signál jednoho cílového řečníka, a potlačit ostatní řečníky a šum. V této práci formulujeme tento problém jako extrakci řeči cílového řečníka a navrhujeme přímé řešení — použití neuronové sítě, která na vstupu přijímá předregistrovanou nahrávku cílového řečníka a pozorovanou směs, a na výstupu vrací extrahovanou řeč cílového řečníka. Diskutujeme a experimentálně ukazujeme výhody tohoto přístupu ve srovnání s konvenční separací řeči. Výhody zahrnují nepotřebnost počítání řečníku ve směsi nebo lepší konzistenci výstupu pro delší nahrávky. Zkoumáme různé aspekty neurální extrakce řeči cílového řečníka, jako jsou embeddingy reprezentující řečníka, metody jak informovat neuronovou síť, vstupní a výstupní doména a ztrátová funkce.

Dále demonstrujeme, jak kombinovat extrakci cílového řečníka s multi-kanálovými metodami, jako je beamforming založený na neurálních maskách nebo prostorové shlukování. Tyto kombinace využívají jak konvenčních statistických metod pro zpracování prostorové informace, tak silné modelovací schopnosti neuronových sítí.

Na závěr aplikujeme extrakci řeči cílového řečníka na dva finální úkoly: automatické rozpoznávání řeči a diarizaci založenou na shlukování. Zkoumáme jak nejlépe zkombinovat předzpracování signálu s cílovými systémy včetně společné optimalizace, nebo trénování se slabou supervizí založenou na informaci o řečnících.

Klíčová slova

extrakce řeči cílového řečníka, neuronové sítě, multi-kanálové zpracování, rozpoznávání řeči více řečníků, diarizace řeči více řečníků

Citace

ŽMOLÍKOVÁ, Kateřina. *Neural target speech extraction.* Brno, 2021. Disertační práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Školitel Jan Černocký.

Neural target speech extraction

Declaration

I hereby declare that this Ph.D. thesis was prepared as an original work by the author under the supervision of prof. Dr. Ing. Jan Černocký. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

.....
Kateřina Žmolíková
March 21, 2022

Acknowledgements

During my Ph.D. studies, I had the luck and the honor to work with many great people. I would like to take this opportunity to express my gratitude.

I thank my supervisor and the leader of our BUTSpeech@FIT team, Honza Černocký, for pulling me into the world of speech research, letting me grow, and supporting all my decisions on the way. Huge thanks also belongs to Lukáš Burget for leading the research in our group, pushing us to learn, to be curious, and to question everything. This work also would not exist without the mentoring and help of Marc Delcroix, whom I thank for the countless hours of discussions, guidance and for always staying positive and encouraging.

I am very grateful to the whole BUTSpeech@FIT research group for the friendly working environment, many coffee breaks, beer days, and long debates about both the research-related and the random topics. I would also like to thank the entire team in NTT Kyoto, where I did my internship. I always felt very welcome during my stays, I learned a lot, and the whole experience kick-started my research.

My thanks also go to the organizers and participants of both Frederick Jelinek Summer Workshops that I took part in. I am thankful for the opportunity to meet and collaborate with such amazing researchers. I also thank the co-authors of all of my publications, who pushed the quality of my research up.

Finally, I thank my family for enabling me to pursue my goals freely. Thanks to my friends for inspiring and shaping me. And a special important thank you to Fede for always pulling my mind away from the work.

Contents

Acronyms

ASA Auditory scene analysis.

ASR automatic speech recognition.

ATF acoustic transfer function.

BLSTM bidirectional long short-term memory.

CACG complex angular central Gaussian.

CACGMM complex angual central Gaussian mixture model.

CASA Computational auditory scene analysis.

CE cross-entropy.

DC Deep clustering.

DER diarization.

EER equal error rate.

FHMM factorial hidden Markov model.

GMM Gaussian mixture model.

HMM Hidden Markov model.

IAM ideal amplitude mask.

IBM ideal binary mask.

IPSM ideal phase-sensitive mask.

LF-MMI lattice-free maximum mutual information.

LSTM long short-term memory.

MFCC Mel-frequency ceptral coefficients.

MSE mean-square error.

MVDR Minimum Variance Distortionless Response.

MWF Multi-channel Wiener filter.

NMF Non-negative matrix factorization.

NN neural network.

PESQ perceptual evaluation of speech quality.

PIT permutation invariant training.

PS-MSE phase-sensitive mean-square error.

RIR room impulse response.

RNN recurrent neural network.

SCM spatial correlation matrix.

SDR signal-to-distortion ratio.

SI-SDR scale-invariant signal-to-distortion ratio.

SNR signal-to-noise ratio.

SPL sound pressure level.

SPP speech presence probability.

STFT short-time Fourier transform.

STOI short-time objective intelligibility.

T-F time-frequency.

TDNN time-delay neural network.

TS-VAD Target-speaker voice activity detection.

TSE target speech extraction.

UBM Universal background model.

WER word error rate.

Chapter 1

Introduction

The applications of speech processing technologies are rising fast in recent years. Digital home assistants with spoken interface have become a common consumer device [HUWN+19]. Applications transcribing speech in real-time can help hard of hearing people communicate more easily [LBK+20]. Annotated meeting transcripts are getting very accurate [YAA+19]. The increase in the practical use of these technologies highlights the need for them to be more robust against environmental distortions. These include background noise, reverberation, and interfering speakers. Especially when all of these factors are present, the performance of speech processing technologies substantially degrades.

The research in speech separation tackles the problem of obtaining speech signals of individual speakers given the observed mixture of all speakers. The separated signals can then be used as an input for further processing, such as speech recognition. Over the past, the research moved from rule-based approaches [BC94] to data-driven techniques based on neural networks [HCLRW16]. These advances broadened the domain where the techniques can be applied, from very limited scenarios, such as known speakers and constrained vocabulary, to the uncontrolled speech of unseen speakers. Nowadays, the field is gradually moving to realistic mixtures that also contain background noise and reverberation.

In many applications, the goal is to enhance the signal of one, or several, pre-defined speakers of interest, while suppressing all remaining interfering speech and noise. For instance, a smartphone can be set to react to speech commands of its owner only, even in environments with background interference. The digital home assistant might focus only on the speech of the user who uttered a particular wake-up keyword. Meeting participants could be pre-enrolled for the system to transcribe each of their speech signals in turn. Such a problem can be tackled by applying speech separation with a subsequent selection of the target speaker.

In this work, we formulate this problem as target speech extraction (TSE) and propose to tackle it more directly, i.e. use a neural network with the enrollment speech and the mixture as the input and the extracted speech of the target speaker as the output. This direct approach has its benefits, such as no need of counting speakers in the mixture, avoiding permutation problems, or better consistency of the output for longer recordings.

There are several aspects of the problem that we describe and analyze. These include the representation of the target speaker, the method of how to inform the neural network using this representation, the domain in which the input and output are represented, or the loss function to train the neural network with. We combine the advances in speech separation, speaker verification, and speaker adaptation in acoustic modeling to guide these choices.

In some practical scenarios, multiple microphones are used to record the scene. When such multi-channel recording is available, it is possible to make use of spatial information to better distinguish different sources of sound. We show how it is possible to take this advantage and combine multi-channel methods with neural-network-based target speech extraction. This includes using target speech extraction in a mask-based beamforming scheme or integrating it with spatial clustering.

Finally, it is important not to study the target speech extraction techniques only in isolation, but also as a pre-processing for subsequent tasks. We inspect the problems of automatic speech recognition and speaker diarization of overlapped speech. We show the challenges of applying target speech extraction for these tasks, such as processing artifacts or inaccurate speaker identification, and possible ways to address those, such as joint training of TSE with the task-specific systems.

1.1 Organization of the thesis

The thesis is organized as follows:

- In Chapter 2, we summarize the general speech separation problem and the popular approaches.
- In Chapter 3, we formulate the target speech extraction problem, emphasize differences to other related tasks, and characterize the challenges.
- In Chapter 4, we further look into different aspects of target speech extraction in a single-channel setting. These aspects are experimentally analyzed and compared with speech separation methods.
- In Chapter 5, we show how to combine target speech extraction with multi-channel approaches, both theoretically and experimentally.
- In Chapter 6, we describe how to combine target speech extraction with automatic speech recognition (ASR) and speaker diarization, including joint training scheme of training TSE and ASR modules.

1.2 Contributions

The formulation of target speech extraction and the use of the speaker-informed neural network to address this problem was first proposed in our work in [ŽDK+17b]. We subsequently developed this idea in [ŽDK+17a, ŽDK+18, ŽDK+19, ŽDR+21, DŽO+19, DOŽ+20]. These publications gave rise to a lot of interest in this problem and many works followed the basic ideas (according to Google Scholar, these works were cited 255 times in total).

This thesis is based on the publications [ŽDK+17b, ŽDK+17a, ŽDK+18, ŽDK+19, ŽDR+21, DŽO+19, DOŽ+20] and updates the experiments with up-to-date architectures and datasets, more closely inspects different aspects of the method and extends it in several directions. Here, we clearly lay out the contributions of our work and link the content of this thesis to our published papers.

The contributions of our work are the following:

- We formulate the target speech extraction problem as opposed to general speech separation. We discuss the advantages of both of these approaches and experimentally compare them. The formulation of target speech extraction was first published in [ŽDK+17b] and made more precise and compared with speech separation in [ŽDK+19]. In this thesis, the comparison is updated to use a more recent neural network model and a variety of datasets.

- We analyze several aspects of the single-channel target speech extraction, namely speaker embeddings, methods to inform the neural network, input/output domain, and the loss function. A similar analysis was partly done in our previous publications (comparison of i-vectors and jointly learned embeddings in [ŽDK+17a], comparison of concatenation, multiplication, and factorized layer in [ŽDK+19]). We include also techniques proposed by other works in the analysis (such as attention-based method, x-vector embedding, time-domain inputs/outputs, and linked loss function). This thesis however presents the first systematic analysis of all of these aspects in one consistent setting over several datasets.

- We adapt and apply techniques that combine neural networks with multi-channel approaches to the problem of target speech extraction. This includes mask-based beamforming, which we first applied to target speech extraction in [ŽDK+17b]. In this thesis, we further extend this with a combination of target speech extraction with spatial clustering.

- We combine target speech extraction with automatic speech recognition (ASR) and investigate several options of training the joint system. We first published our ASR experiments on extracted speech in [ŽDK+17a] and joint training of TSE and ASR in [ŽDK+18].

- We propose a way to combine TSE with clustering-based speaker diarization. This combination is first proposed in this thesis and was previously not published.

- We propose an auxiliary weakly supervised loss function based on speaker characteristics to fine-tune TSE with the application of ASR. This was proposed in our Interspeech 2021 paper [ŽDR+21].

Chapter 2

Speech separation

This thesis focuses on the problem of target speech extraction, i.e. extracting the speech of a target speaker from a mixture of multiple speakers. Most relevant previous work for this problem is in the field of speech separation, i.e. blind estimation of signals of all speakers in the mixture. In this chapter, we thus overview the speech separation task and approaches. In the next chapter, we will relate this problem to target speech extraction.

2.1 Problem of overlapping speakers

In many applications of speech technologies, we encounter the problem of interfering speakers disrupting the speech of interest. For instance, automatic speech recognition has the potential to be applied for automatic transcription of meetings, where the ratio of speaking time interrupted by interfering speaker is estimated to be more than 10% [ÇS06]. In other application areas, such as in voice assistants used in home environments, or when creating automatic subtitles for YouTube videos, the overlapped speech also naturally occurs.

The presence of the interfering speakers deteriorates the performance of speech technologies significantly. In a recent CHiME-5 challenge [BWVT18], considering the problem of distant speech recognition in a challenging scenario, the error rate of the winning system drops from 40% on low-overlap recordings to 60% on high-overlap ones [Bar]. The overlapped speech also presents a big challenge in today's state-of-the-art diarization systems [SSM+18]. Furthermore, it has been shown that overlapped speech is a big obstacle in understanding speech for users of hearing aids [BP92].

In all of these scenarios, a pre-processing step that removes the interfering speech and enhances the speech of interest could lead to a major improvement of the technology. Such a task, formulated as speech separation or target speech extraction, has been long considered extremely challenging. Traditional methods aiming at enhancement of speech in noise often leveraged the different statistics of speech and noise signals to distinguish between the two [Loi07]. In the case of interfering speech, this is not possible, as the interference is a signal of the same category. However, with the rise of neural networks, there have been huge advances in the task [HCLRW16, YKTJ17, LM19] and today, we are realistically close to applying such pre-processing in practice.

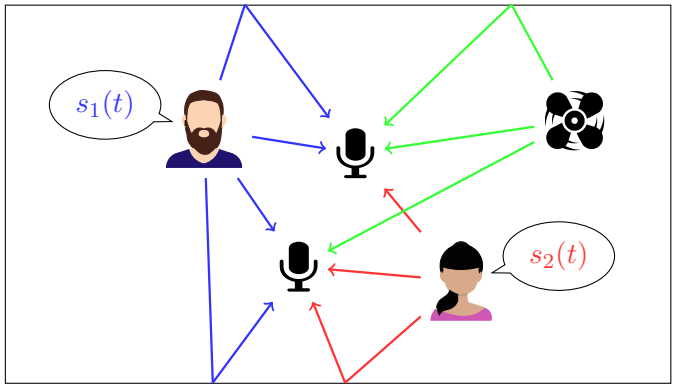


Figure 2.1: Considered scenario. Multiple speech and noise sources are present in the room and captured by several microphones. The microphones capture both the direct paths and the reflections of the walls, floor, ceiling, or objects.

2.2 Task definition

In this section, we formally define the task of speech separation. We describe the overall setting — the observed signals, the goal of the method, its inputs, and outputs. The depiction of the considered scenario is in Figure 2.1.

We expect a scenario where multiple (J) speakers are speaking in a room with possible other sources of noise and possibly multiple (K) microphones recording the scene. The signals received at the microphones can then be modeled as:

$$y^{(m)}(t) = \sum_{j=1}^{J} a_j^{(m)}(t) \star s_j(t) + v^{(m)}(t), \tag{2.1}$$

where t is the index of the sample, $y^{(m)}(t)$ is the observed mixture at the microphone m, $s_j(t)$ is the speech signal of the speaker j, $a_j^{(m)}(t)$ is the room impulse response (RIR) between the speaker j and microphone m, and $v^{(m)}(t)$ is the noise signal including the RIR from the sources of the noise to the microphone m. When dealing with a single-channel use case, we will omit the microphone index $^{(m)}$.

Often, the signals are processed in the frequency domain, in which case the assumed model transforms into

$$Y^{(m)}(n,f) = \sum_{j=1}^{J} A_j^{(m)}(n,f) S_j(n,f) + V^{(m)}(n,f), \tag{2.2}$$

where $Y^{(m)}(n,f), S_j(n,f), V^{(m)}(n,f)$ are the short-time Fourier transform (STFT) counterparts of $y^{(m)}(t), s_j(t), v^{(m)}(t)$, respectively, $A_j^{(m)}(n,f)$ models the effect of the room impulse response in the frequency domain, n is the index of STFT frame, and f is the index of frequency bin. With all signals in STFT domain $\cdot(n,f)$, we denote the magnitude part as $|\cdot(n,f)|$ and the phase part as $\angle\cdot(n,f)$.

The task of speech separation is to retrieve the speech signal of the speakers $s_j(t)$ given the observed signal $y^{(m)}(t)$ at all microphones $m \in [1,K]$. We define the processing as

$$\{\hat{s}_j(t)\}_{j=1..J} = \mathcal{F}\left(\left[y^{(m)}(t)\right]_{m=1..K}\right), \tag{2.3}$$

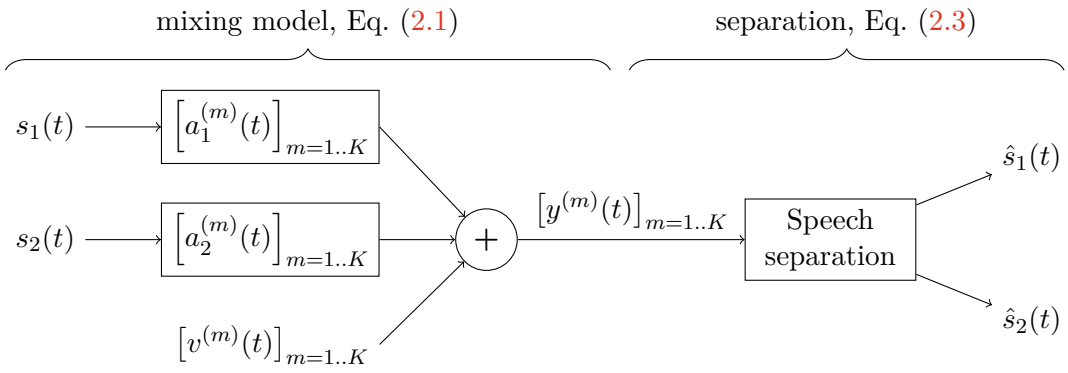


Figure 2.2: The assumed mixing model of the observed signal and the processing by speech separation method. This is an example for $J = 2$ speakers.

where $\hat{s}_j(t)$ is the speech of speaker j estimated by the method. The schematic overview of both assumed generative and separation process is shown in Figure 2.2.

2.3 Evaluation

The goal of speech separation is to get the estimate of $\hat{s}_j(t)$ as close as possible to the source signal $s_j(t)$. There are different ways to evaluate the performance of a particular method, which we organize into three categories:

1. objective metrics
2. subjective metrics
3. down-stream task evaluation metrics

Objective metrics

Objective metrics are the most commonly reported ones in the literature, mainly for the ease of their evaluation. They can be divided into two categories, depending on whether they make use of the ground truth signal. In *intrusive* metrics, we evaluate the discrepancy between the ground truth $s_j(t)$ and the estimated signal $\hat{s}_j(t)$. When the ground truth signal is not available, it is possible to use *non-intrusive* metrics computed only from the estimated signal. Non-intrusive evaluation metrics are more difficult to design and generally have a lower correlation with the actual intelligibility [And17]. For this reason, we use intrusive evaluation metrics in this work. Here, we review three different metrics, namely signal-to-distortion ratio (SDR) and its variants, short-time objective intelligibility (STOI), and perceptual evaluation of speech quality (PESQ).

Signal-to-distortion ratio is probably the most popular metric in recent speech separation and target speech extraction literature. Here, we will discuss its two variants, i.e. signal-to-distortion as defined in [VGF06] and implemented in `bss_eval` toolbox[1] (`bss_eval` SDR or just SDR) and scale-invariant signal-to-distortion ratio (SI-SDR, also

[1] bss_eval toolbox `http://bass-db.gforge.inria.fr/bss_eval/`

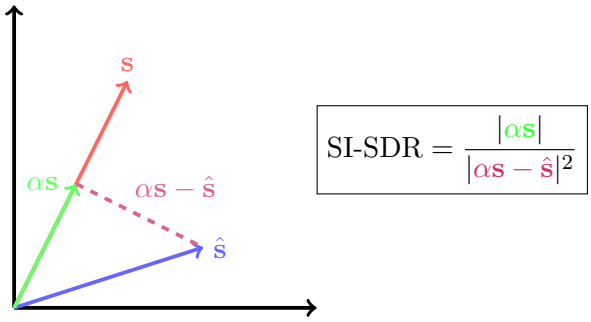


Figure 2.3: Geometrical interpretation of the SI-SDR loss. For illustration $s(t)$ and $\hat{s}(t)$ are reduced here to two-dimensional vectors, $\mathbf{s}$ and $\hat{\mathbf{s}}$. Scaling of the reference $\mathbf{s}$ corresponds to orthogonal projection of the estimate $\hat{\mathbf{s}}$ onto $\mathbf{s}$.

known as SI-SNR) [LRWEH19]. First, `bss_eval` SDR is defined as

$$\begin{aligned} \mathrm{SDR}(s(t), \hat{s}(t)) &= 10 \log_{10} \frac{\sum_t |\alpha(t) \star s(t)|^2}{\sum_t |\alpha(t) \star s(t) - \hat{s}(t)|^2} \\ \text{with} \quad \boldsymbol{\alpha} &= \operatorname*{argmin}_{\boldsymbol{\alpha}} \sum_t |\alpha(t) \star s(t) - \hat{s}(t)|^2, \end{aligned} \tag{2.4}$$

where $\boldsymbol{\alpha}$ is a filter of maximum length $\tau_{\max} = 512$ (corresponding to 64 ms for sampling frequency 8000 Hz) [VGF06, DHBHU19]. This measures the distortion between reference $s(t)$ and estimate $\hat{s}(t)$, but allows for a short linear filter to be applied on the target to match the estimate. This can tolerate short delays of the estimated signal, which can happen for example when we use recording of different microphone as the reference [DHBHU19]. However, SDR can be overly permissive and lead to over-optimistic results even when some frequency bands are completely omitted, as discussed in [LRWEH19].

SI-SDR is less permissive in this regard and allows only for a scale α to be applied to the target signal [LRWEH19]:

$$\begin{aligned} \text{SI-SDR}(s(t), \hat{s}(t)) &= 10 \log_{10} \frac{\sum_t |\alpha s(t)|^2}{\sum_t |\alpha s(t) - \hat{s}(t)|^2} \\ \text{with} \quad \alpha &= \operatorname*{argmin}_{\alpha} \sum_t |\alpha s(t) - \hat{s}(t)|^2. \end{aligned} \tag{2.5}$$

The optimal scale α can be obtained as $\alpha = \sum_t s(t)\hat{s}(t) / \sum_t s^2(t)$. The scaled reference $\alpha s(t)$ thus becomes an orthogonal projection of the estimated signal $\hat{s}(t)$. This is depicted in Figure 2.3. The metric is highly sensitive to small delays between the estimated and target signal. On the other hand, in contrast with SDR, it does not score well the results of degenerate solutions such as band-stop filters.

Short-time objective intelligibility [THHJ10] is a metric designed to well correlate with subjective intelligibility, and thus is able to replace costly listening tests. It first converts both reference and estimated signals into one-third octave band representations $S^{(oct)}(n, o)$, $\hat{S}^{(oct)}(n, o)$, respectively, where $o \in [0, 14]$ is the index of third-octave band. The measure is first computed locally, i.e. for each time frame n and band index o, a score is computed from a window of 30 consecutive frames of the reference $S^{(oct)}(n-30+1 \ldots n, o)$ and the estimate $\hat{S}^{(oct)}(n-30+1 \ldots n, o)$. The estimate window is first normalized to have

equal energy to the reference window and then clipped to prevent extreme values. STOI is then the average linear correlation coefficient between the reference and estimate window

$$\text{STOI}(s(t), \hat{s}(t)) = \frac{1}{NO} \sum_{n,o} \text{corr}(S^{(oct)}(n-30+1 \dots n, o), \hat{S}'^{(oct)}(n-30+1 \dots n, o)), \quad (2.6)$$

where $\hat{S}'^{(oct)}(n-30+1 \dots n, o))$ is the normalized and clipped version of the estimate window, n is the frame index, o is the octave band index and corr is the linear correlation coefficient. For more details on the computation steps, we refer to the original proposal [THHJ10].

Perceptual evaluation of speech quality (PESQ) [RBHH01] is a measure designed to approximate the subjective mean opinion score (MOS) of speech quality. It was originally proposed for the assessment of telephone networks and codecs and today is also often used for evaluating speech enhancement algorithms. Both the reference and the estimated signals are transformed into a representation of perceived loudness in time and frequency, using a psycho-acoustic model. The difference between the representations is further passed through processing inspired by human cognition. We refer to the original work [RBHH01] for more details about the computation.

Subjective metrics

Subjective metrics are obtained by performing listening tests with a group of listeners. For applications where the enhanced signals are intended to be listened to, such as hearing aids, performing listening tests can best reflect the final performance. There are standard methodologies, that can be followed, such as the *MUltiple Stimuli with Hidden Reference and Anchor* framework (MUSHRA) [Ser14] or the *Mean opinion score* framework (MOS) [SWH16], which can be used for measuring speech quality. For measuring speech intelligibility, the participants of the listening test are asked to identify the spoken words. The percentage of correctly recognized words can then be used as the intelligibility measure [MSCM12].

The disadvantage of subjective metrics is the difficulty of the evaluation. Especially when developing new methods, it would be very costly to perform listening tests for each modification of the techniques. Recently, there has been research on approximating the subjective metrics using neural networks [RGC21]. Furthermore, when the speech extraction acts as a pre-processing for another system, such as an automatic speech recognizer, the subjective metrics may not well reflect the effect on such system.

Down-stream task metrics

In the case, when the outputs of the target speech extraction are used as input to another system, the best evaluation is the actual performance of the final system. This can be for example word error rate (WER) of an automatic speech recognizer, equal error rate (EER) of a speaker verification system, or diarization (DER) of a diarization system. In Chapter 6, we cover some of these use-cases in more detail.

The disadvantage of such kind of evaluation is that it is usually more time-consuming than the more easy-to-evaluate objective metrics. In addition, the evaluation is dependent on a particular system and if the final system changes during the development, the metric may need to be re-evaluated.

2.4 Pre-neural approaches

The problem of speech separation has a long history of research. Although today, the accuracy of neural networks seems to surpass all the previously used approaches, some ideas from the past get re-used and combined with more modern methods. Here, we summarize three important approaches, i.e. Computational auditory scene analysis (CASA), Non-negative matrix factorization (NMF), and factorial hidden Markov model (FHMM).

2.4.1 Computational auditory scene analysis

Computational auditory scene analysis [BC94, Ell96] is inspired by findings about human auditory system. It is known that humans have quite a remarkable ability to understand speech in very challenging conditions, even in presence of interfering speakers. The auditory and cognitive processes, enabling us to do so, have been long studied and described by a theory known as Auditory scene analysis (ASA) [Bre94]. The basic theory of ASA states that the auditory processes first transform the signal into a spectro-temporal representation. Elements in this representation then get clustered based on different grouping cues, such as harmonicity, common onsets, or amplitude modulation of different harmonic components. CASA exactly follows these steps, with variations among works in the type of representation used or differently designed grouping rules. Although CASA methods are well-grounded in psychoacoustic research, they do not achieve the performance of later data-driven methods.

2.4.2 Non-negative matrix factorization

Non-negative matrix factorization [SFM+14, LS01, Vir07] has been for a long time a very popular method for speech separation and was often used as a baseline for later approaches. NMF was originally designed for dimensionality reduction [WZ13, PT94] and is part of a more general family of approaches for decomposition of a data matrix into two factor matrices (including Principal component analysis, Linear discriminant analysis, and others). The uniqueness of NMF lies in the non-negativity constraint imposed on the factor matrices. Due to this constraint, the learned components can be only combined in an additive way and cannot cancel each other. This brings better interpretability of the components as physically meaningful parts of the input.

When using NMF for speech separation, we decompose the observed spectrogram $\mathbf{V}$ into two matrices $\mathbf{V} \approx \mathbf{WH}$, where $\mathbf{W}$ are the components and $\mathbf{H}$ their activations. The components $\mathbf{W}$ are usually learned on clean examples of the target sources, leading to a source-specific dictionary. By inferring the activations $\mathbf{H}$ for a mixture with the fixed pre-learned dictionary $\mathbf{W}$, we can reconstruct the individual sources. In the simplest form, NMF does not leverage any temporal dependencies in the data. To overcome this, several dynamic versions of NMF have been proposed, where the temporal continuity can be either enforced on the level of the components (e.g. convolutive NMF [Sma07]) or on the level of the activations (e.g. smooth NMF [FBD09], non-negative dynamical systems [FLRH13], non-negative hidden Markov models [MS12]). Today, most of the research on NMF for speech separation looks into its combinations or extensions with neural networks [LRHW15, SV17].

2.4.3 Factorial hidden Markov models

Factorial hidden Markov models (FHMM) [HROK10, Vir06] are another notable approach that used to be considered state-of-the-art in speech separation [KHO+06]. FHMM is

used as a generative model of mixed speech, usually at the level of log-spectrogram. It consists of separate models for generating (hidden) features of each of the speakers and an interaction model, combining the features generated by the speaker models to explain the mixed features. The single-speaker speech is modeled by GMM-HMM. There are different alternatives for the interaction model which approximate the exact interaction function in the log-spectral domain. One common example is the max-model, where the mixed feature is modeled as the maximum of the features of individual speakers. The process of separation is then an inference of the hidden features for each speaker given the mixed features. The inference in the model is the main shortcoming of the method, as it is highly complex. The model also cannot effectively handle unknown speakers or environments.

2.5 Neural approaches

Over the last decade, neural networks have become a predominant model in many machine learning fields such as computer vision, language modeling, speech recognition, or speaker identification. They have been shown to significantly outperform the previous approaches. The same trend can be seen for the speech separation task, where the deployment of neural networks leads to substantial advances in performance. In this section, we summarize several neural approaches for speech separation.

2.5.1 Early neural network methods

In early neural network approaches, the problem of speech separation was often constrained to a limited scenario. Numerous works focus on the case of two-speaker male-female mixtures [WDDL16, CK17, HKHJS14]. In this case, the neural network's output is split into two parts, one for the female signal and one for the male signal. A similar example of a simplifying assumption is the work in [WYSD15a], where the neural network is trained to separate two-speaker mixtures, where one of the speakers is always strongly dominant. This can be a reasonable assumption for some use-cases, however, still not applicable in general. Another line of research focused on the case of a closed-set of speakers, where the data used in test-time contain only the speakers present in the training dataset [DTX+14, DTDL16, ZW16]. In this case, the neural network is trained separately for each speaker pair. This assumes the availability of a sufficient amount of data from each speaker and is not easily extendable to unseen speakers.

Although the listed works have quite strong limitations, they are valuable for showing the capabilities of neural networks to separate speech and exploring various aspects, such as suitable input and target representations. This set the ground for more elaborate methods extending the applicability to more general cases.

2.5.2 Deep clustering

In contrast with most of the other approaches using the neural network to directly perform the regression from the mixed to the clean speech, Deep clustering (DC) [HCLRW16, IRC+16] splits the procedure into two distinct steps. First, the neural network predicts an embedding for each time-frequency (T-F) point of the input mixture. Second, these embeddings get clustered using conventional clustering methods, such as k-means. For training of the neural network, the objective function is constructed in such a way, that embeddings for T-F points corresponding to the same speaker are pulled closer to each other, while em-

beddings for T-F points of different speakers are pulled away. In particular, the objective function takes the form

$$\mathcal{L}_{\text{dc}} = |\mathbf{V}\mathbf{V}^{\mathsf{T}} - \mathbf{E}\mathbf{E}^{\mathsf{T}}|_{\text{F}}^2, \tag{2.7}$$

where $\mathbf{E}$ is the $N \times J$ matrix denoting the target speaker for each T-F point (N denotes the number of T-F points, J denotes the number of speakers), $\mathbf{V}$ is the $N \times D$ matrix of estimated embeddings (D denotes dimension of the embedding) and $|\cdot|_F^2$ is the squared Frobenius norm. The target speaker for each T-F bin is determined from the clean signals as the most dominant of the speakers. The multiplication $\mathbf{E}\mathbf{E}^{\mathsf{T}}$ results in $N \times N$ affinity matrix with 1 for a pair of T-F bins from the same speaker and 0 for a pair of T-F bins from different speakers. The $\mathbf{V}\mathbf{V}^{\mathsf{T}}$ is the $N \times N$ estimated affinity matrix, where each element contains the dot product of the embeddings for the pair of T-F bins.

Note that the objective function of DC is independent of the order of the speakers in the labels, i.e. permuting the columns of $\mathbf{E}$ does not change the value of the objective. The architecture of the neural network and the objective function are also independent of the number of speakers in the mixture, although in the testing time, the number of speakers needs to be known or estimated in the clustering step.

A common critique of Deep clustering is the mismatch between the objective function and the clustering algorithm actually used in testing time to separate the sources. The indirect nature of the objective function also makes it difficult to use DC in end-to-end frameworks, where the separation is trained to optimize a final task, such as speech recognition. A variant of DC, Deep attractor network (DaNet) [CLM17] addresses these issues. In DaNet, the objective function is modified by creating attractor points in the embedding space as centroids of embeddings of each of the sources. Based on the attractor points, T-F masks are computed, and the objective function is the mean squared error between the masked mixture signal and the clean signal. In the test time, the masks are computed with the same procedure, eliminating the training-testing mismatch.

2.5.3 Permutation invariant training

In permutation invariant training (PIT) [YKTJ17, KYT+17, QCY18] the neural network maps an input mixture speech signal to an output consisting of clean speech signals from all the sources. The key idea of PIT lies in how the reference signals of all speakers are assigned to the outputs of the neural network during training. All possible permutations of the outputs are considered for computing the objective function, and the one with the lowest error is selected. The neural network is thus free to choose any order of the speakers on the output. The objective function is

$$\mathcal{L}_{\text{pit}} = \min_{\phi \in P} \sum_{j=1}^{J} \ell(\hat{s}_{\phi(j)}, s_j), \tag{2.8}$$

where ℓ is the loss function comparing two signals, $\hat{s}_i$ is the ith estimated signal, s_j is the jth reference signal, P is the set of all possible permutations of the outputs and ϕ is one of the permutations. Although the original PIT works used the loss in frequency-domain, we choose a more general formulation, as in later publications it was applied to different representations as well.

The architecture of the neural network in PIT depends on the number of speakers. However, in [KYT+17] authors claim that this can be addressed by fixing the number of

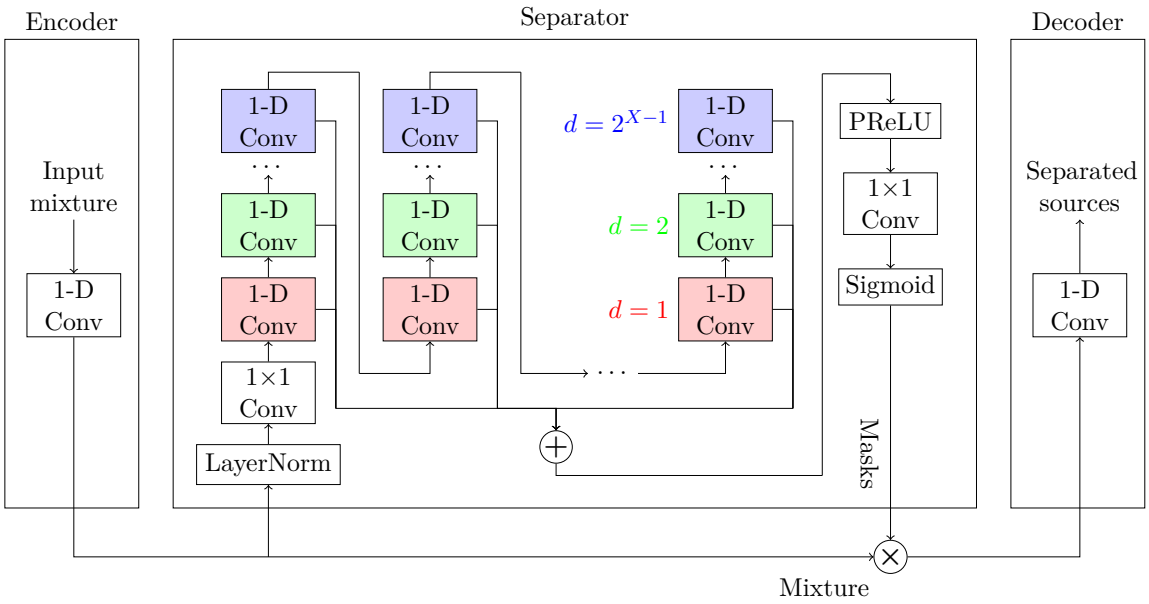


Figure 2.4: ConvTasnet model for speech separation. Colors denote the dilations factors d in the individual 1-D convolutional blocks. Image adapted from [LM19].

outputs as the maximum number of speakers in the mixture and training the network to output silence in case there are fewer.

The permutation invariant objective became quite popular for its simplicity and is commonly used. The combination of PIT and DC was also explored in some works [LCH+17, SLRH+18]. PIT can be also easily extended with end-to-end training with a speech recognition system [YCQ17].

2.5.4 ConvTasnet

Both DC and PIT were introduced as frequency-domain approaches, where both inputs and outputs of the network are time-frequency matrices. The neural networks processing this input were then usually recurrent networks, such as Long-short term memory networks [HS97] or their bidirectional variant. This trend changed with the publication of the ConvTasnet neural network for speech separation [LM19]. ConvTasnet is a convolutional network that works directly with time-domain signals and uses the PIT principle to compute the loss function.

ConvTasnet architecture is composed of three main parts, namely encoder, separator, and decoder as depicted in Figure 2.4. The encoder is a convolutional layer, transforming the time-domain signal into a higher-level representation. The decoder on the other hand reverses this representation back into the time domain with a transposed convolutional layer. The most important part is the separator, estimating a mask to apply to the encoded representation in order to separate the individual speakers. The separator consists of several repetitions, where each repetition is a sequence of convolutional blocks with increasing dilation. This architecture is inspired by the previous success of temporal convolutional networks [LVRH16]. The increasing dilation factor in the convolutional filters results in an

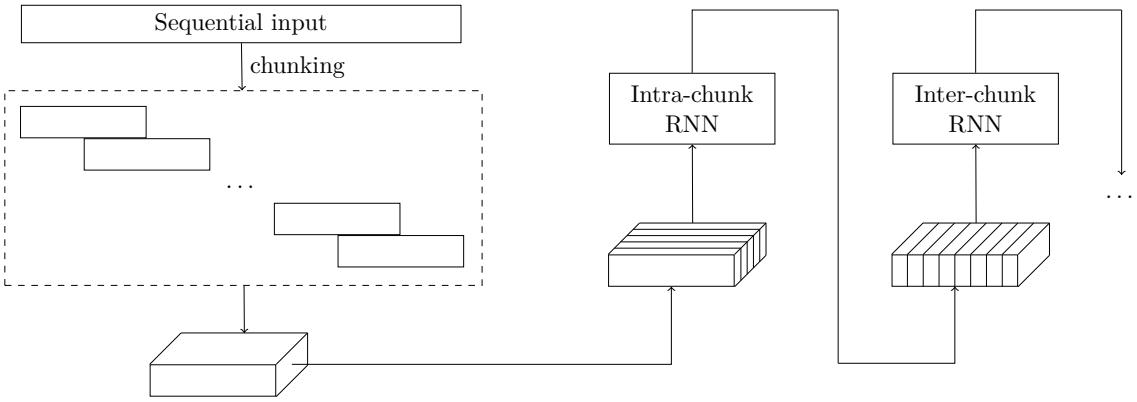


Figure 2.5: Dual path RNN model for speech separation. Image adapted from [LCY20].

increased reception field of the model. The exact size of the reception field depends on the hyper-parameters of the model, but is typically in the order of seconds.

The ConvTasnet work caused a small revolution in the speech separation field. Most subsequent works follow the trend of time-domain approaches with PIT-based loss functions and only differ in the architecture used in the separator.

2.5.5 Dual path RNN

The success of time-domain approaches has shown that it is beneficial to process the mixed-signal with a high time resolution. For example, with ConvTasnet, the encoder uses much smaller windows than would be typically used for frequency-domain representations. This leads to several times longer sequences of features that the neural network works with, increasing the need for a model to be able to work with long context. While recurrent neural networks (RNN) can in theory learn long-term temporal dependency, in practice, they are not effective due to optimization issues, such as vanishing gradients.

Dual-path RNN networks [LCY20] use RNN layers, but organize them in a way that allows them to model long sequences. The input sequence is split into chunks and two types of RNN are interleaved – intra-chunk RNN, processing each chunk independently, and inter-chunk RNN aggregating the information over all the chunks. The two types of RNN layers are then repeated several times. In this way, the individual RNNs perform on much shorter sequences, but together, they can learn even the long-term dependency. The dual-path principle is depicted in Figure 2.5.

The dual-path network has been then followed in other works. For example, in [NAW20] the RNNs were extended with multiplication-concatenation blocks, or in [SRC+21] the RNN layers were replaced by attention.

Chapter 3

Target speech extraction problem

In this chapter, we introduce the problem of target speech extraction (TSE) and its relation to other tasks, including speech separation. We also discuss the main factors influencing TSE performance and briefly overview how TSE relates to the processing of multi-talker speech by humans.

3.1 Task definition

In our work, we aim to tackle the problem of overlapping speakers by applying target speech extraction, i.e. extracting the speech signal of the speaker of interest while suppressing the speech of all interfering speakers and potential noise. In this section, we point out the differences between this formulation of the task and the task definition of speech separation presented in Section 2.2.

The scenario we tackle is the same as in speech separation, with the difference of considering one speaker as the target speaker. We can rewrite the signal model in both time- and frequency-domain as

$$y^{(m)}(t) = a_i^{(m)}(t) \star s_i(t) + \sum_{j \neq i} a_j^{(m)}(t) \star s_j(t) + v^{(m)}(t) \tag{3.1}$$

$$Y^{(m)}(n,f) = A_i^{(m)}(n,f) S_i(n,f) + \sum_{j \neq i} A_j^{(m)}(n,f) S_j(n,f) + V^{(m)}(n,f), \tag{3.2}$$

where i is the index of the target speaker. We follow the notation introduced in Section 2.2, i.e. t is the index of the sample, $y^{(m)}(t)$ is the observed mixture at microphone m, $s_j(t)$ is the speech signal of the speaker j, $a_j^{(m)}(t)$ is the RIR between the speaker j and microphone m, $v^{(m)}(t)$ is the noise signal including the RIR from the sources of the noise to the microphone m; $Y^{(m)}(n,f), S_j(n,f), V^{(m)}(n,f)$ are the short-time Fourier transform (STFT) counterparts of $y^{(m)}(t), s_j(t), v^{(m)}(t)$, respectively, $A_j^{(m)}(n,f)$ models the effect of the room impulse response in the frequency domain, n is the index of STFT frame, and f is the index of frequency bin.

To characterize the target speaker, we assume having an enrollment signal spoken by the target speaker i, denoted as $e_i(t)$. In our work, we consider the enrollment to be single-channel. The task of target speech extraction is to retrieve the speech signal of the target speaker $s_i(t)$ given the observed signal $y^{(m)}(t)$ at all microphones $m \in [1, K]$ and the enrollment utterance $e_i(t)$. We define the processing as

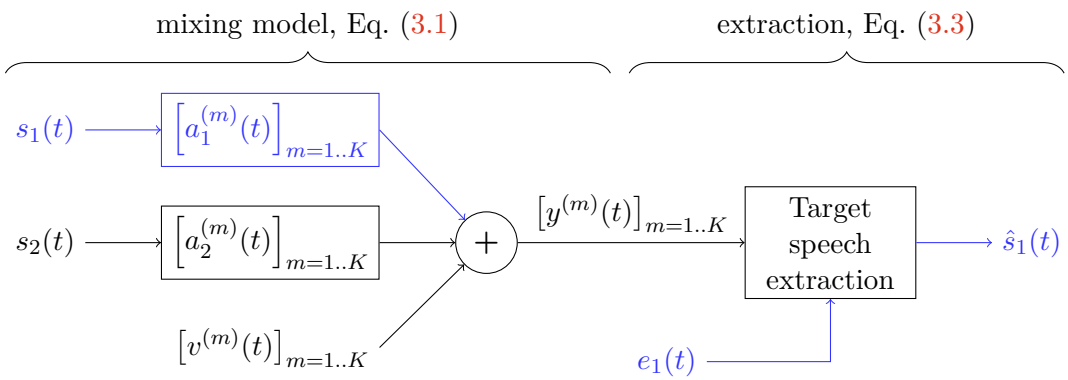


Figure 3.1: Assumed generative model and processing by target speech extraction method. This is an example of two speakers and target speaker $i = 1$ (in blue).

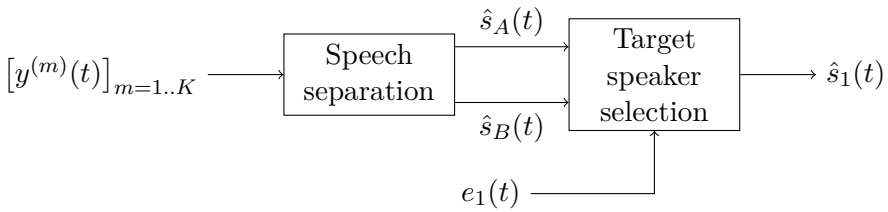


Figure 3.2: Cascade of speech separation and target speaker selection.

$$\hat{s}_i(t) = \mathcal{F}\left(\left[y^{(m)}(t)\right]_{m=1..K}, e_i(t)\right), \tag{3.3}$$

where $\hat{s}_i(t)$ is the target speech estimated by the method. The schematic overview of both the assumed generative process and our method is shown in Figure 3.1.

3.2 Relation to speech separation

In speech separation, the task is to extract the speech signals of all speakers present in a mixture. Nowadays, this task is mostly tackled by neural networks, and target speech extraction methods often build upon these models. The task of target speech extraction could be also solved by a cascade of speech separation and selection of the target speaker from the outputs using, for example, a speaker recognition system. This option is schematically depicted in Figure 3.2.

In the following, we discuss the benefits and drawbacks of both approaches, i.e. target speech extraction and speech separation. We mainly focus on neural network methods here, as they are the current state-of-the-art.

3.2.1 Benefits of target speech extraction

1. *No speaker counting necessary.* A typical neural network for the speech separation task has as many outputs as there are speakers [YKTJ17]. This makes the architecture dependent on the number of speakers in the mixture, and there is a need to either

have prior knowledge about the number of speakers, or first perform speaker counting. Some speech separation works showed that it is possible to train a neural network that has a larger number of outputs to separate a variable number of speakers and keep the remaining outputs as zero or small noise [KYT+17]. This partially solves the problem, but the neural network still needs to perform the counting of the speakers internally. In some cases, such as a restaurant environment with a lot of babble noise, it might be also difficult to define, which part of the signal should be still considered as a speech to separate and which is already part of the noise. The direct target speech extraction completely avoids these problems, as the neural network has only one output for the target speaker. The architecture is thus completely independent of the number of speakers present in the mixture.

2. *No permutation solving.* The fact that the neural network for speech separation has one output per speaker, leads to issues related to the permutation of the speakers. During the training of the network, it is not clear which speaker appears at which output. For example, if an input mixture consists of three speakers A-B-C, there are six possible orderings in which the speakers can appear at the outputs of the network (A-B-C, A-C-B, B-A-C, B-C-A, C-A-B, C-B-A) and all of these should be considered correct. The training thus needs to use specialized loss functions, such as the most popular permutation invariant training [YKTJ17], which computes the loss for all possible permutations and chooses the best one. The permutation problems might also appear during the inference when we split recordings into blocks to be separated using the network. In each block, the permutation on the output might be different, and we need to use "stitching" techniques to correctly align the speakers [YEC+18]. In some works, this is called "the global permutation problem". Target speech extraction completely avoids the permutation problems by using the additional target speaker information.

3. *Consistent output for longer recordings.* Speech separation models sometimes make errors by switching the speakers in the output streams in the middle of a sequence [ŽDK+19]. Although such behavior is penalized in the loss function during the training, keeping the speakers on the output in a consistent order requires the network to see enough context and "remember" the ordering in previous frames correctly. Empirically, this is a common source of errors. In target speech extraction, the output is kept consistent due to the usage of the enrollment speaker information.

4. *One compact model trained for the task.* In general, having one model solving the task directly can be preferable to having separate modules. This is because all parameters are optimized directly for the final task and can thus get closer to the optimum than when two modules are optimized with different objective functions.

3.2.2 Benefits of speech separation

1. *No enrollment necessary.* When a prior enrollment recording of the speaker is not available and not possible to obtain, the target speech extraction models cannot be applied. The task of target speech extraction is itself ill-defined in this case, and it is necessary to use the speech separation model if we want to extract the speech of individual speakers. Some works explored models that can act as both separation and target speech extraction, depending on the availability of the enrollment utterance [ODK+19b].

2. *Possibility to tune the selection module separately.* In some cases, it might be advantageous to be able to explicitly modify only the speaker selection process. For instance, it might be possible to make use of speaker verification models pre-trained on a large amount of data. Furthermore, in some situations, we want to allow the enrollment speaker not to be present in the mixture. The goal in such a case is to extract a silent signal. For such cases, it might be useful to have an explicit threshold on the similarity between the enrollment and mixture speakers, which decides whether the enrollment speaker is present in the mixture or not. In the case of the target speech extraction model, the threshold is implicitly learned by the neural network, and it is not possible to tune it easily.

3. *Less computation when extracting multiple speakers.* If our goal is to extract all or multiple speakers from the mixture, applying speech separation models requires less computation than the target speech extraction. In the target speech extraction case, it is necessary to forward the data once for each desired speaker. In speech separation, only one forward pass extracts all the speakers.

In general, both approaches have their pros and cons and can be beneficial in different applications. We further elaborate and experimentally verify some mentioned properties in Section 4.8. It is noteworthy that some recent models [ZG21] perform speech separation by first estimating speaker embeddings and then extracting all speakers given the embeddings. Such a scheme can combine some benefits of both approaches.

3.3 Relation to other tasks

Target speech extraction is closely related to several other tasks in the speech processing field. Here, we discuss the similarity and differences with several tasks, namely speech enhancement, speaker diarization, and speaker adaptation.

Speech enhancement

In speech enhancement [Loi07], the task is to remove noise from a noisy speech signal. This is similar to target speech extraction, as in both cases, we have speech corrupted by interference as the input and the clean speech signal as the desired output. Similar methods can be applied in both tasks, and neural architectures for speech enhancement and target speech extraction are indeed very similar.

The main difference between the tasks comes from the nature of the interference. In speech enhancement, the models can use the specific properties of speech and noise to differentiate between the two, remove the noise and extract the speech. Generally, in target speech extraction, this is not possible because the interference is also a speech signal. To break this symmetry, we need to provide additional information about the target speaker. Some early works in target speech extraction also avoided the symmetry by solving more constrained tasks, such as extracting the dominant speaker [WYSD15b] or training a specialized neural network for a particular speaker [ŽDK+17b, DTX+14, ZW16]. This brings the task closer to speech enhancement, but has limited applicability.

Speaker diarization

The task of speaker diarization is to identify "who speaks when", i.e. identify how many speakers there are in a recording and when each of them is speaking [ABE+12]. Typically, this is done on long recordings of several minutes. If enrollment utterances for each of the speakers were available, we could imagine solving the task by running the target speech extraction system for each of the speakers and then using voice activity detection on the outputs. Even when no enrollment is available (as is usually the case for diarization tasks), it could be obtained from some single-speaker segments of the recording identified by preliminary diarization.

Although this is possible in theory, such a scheme is not widely used in diarization. A reason for that might be that target speech extraction is essentially a more difficult task than diarization. While diarization needs to only decide about the speaker activity pattern, target speech extraction needs to estimate the speech signals themselves. For this reason, it is likely that approaches directly designed for diarization will always perform better, than the application of target speech extraction.

Despite these reasons, both approaches can be combined and benefit from each other, as done in some recent works [DŽO+21]. We further explore the combination of target speech extraction and diarization in Chapter 6. Furthermore, a recent approach for diarization, Target-speaker voice activity detection (TS-VAD) [MKP+20], has been inspired by target speech extraction techniques. TS-VAD estimated target speaker activity given an enrollment utterance of the speaker. This approach has been very successful in recent diarization challenges [WMB+20, RSK+21].

Speaker adaptation in speech recognition

In automatic speech recognition (ASR) systems, the task is to obtain a sequence of words spoken in the input recording. Today, this is mostly tackled by neural networks. A lot of research has been done on how to adapt the neural network to a particular speaker [SSNP13, Lia13, KVv+21], i.e. given a short speech of the speaker, alter the forward pass through the network so that it better recognizes the target speaker's speech.

The basic principle — altering the forward pass through the neural network using an enrollment utterance of the speaker — is analogous to the task of target speech extraction. Indeed, many methods for target speech extraction have been inspired by speaker adaptation in ASR. However, there are some differences between the two tasks. In speaker adaptation, we aim to only slightly alter the output of the network by changing the speaker information, and the model can work reasonably well even without adapting to a particular speaker. In contrast, in target speech extraction, the speaker information is essential, and changing the enrollment speaker should totally change the output. Consequently, the optimal methods for both tasks may be different.

3.4 Factors influencing target speech extraction performance

The target speech extraction is influenced by different factors, which make the task more difficult. Here, we describe several of those, namely noise, reverberation, voice characteristics, and domain mismatch.

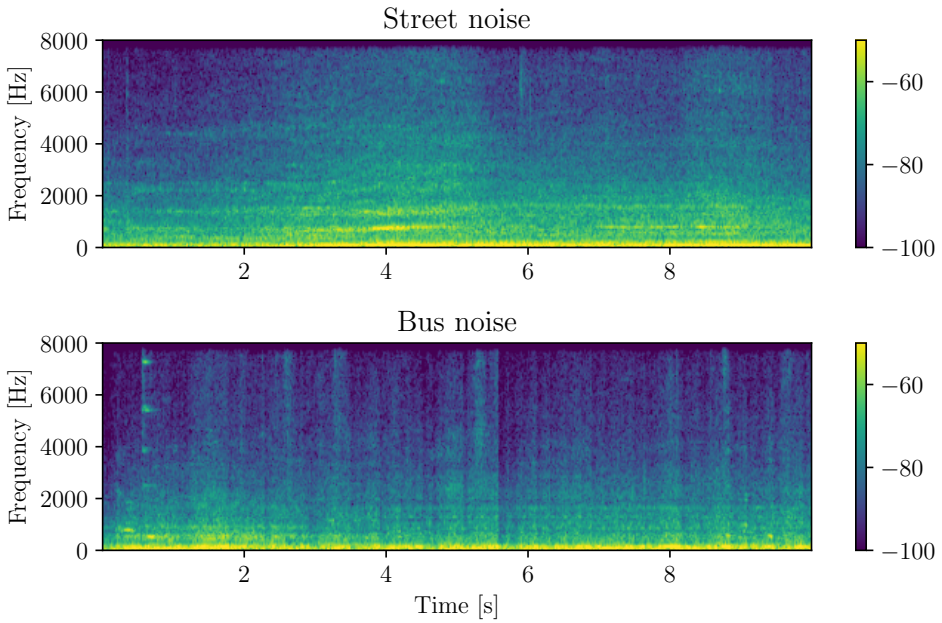


Figure 3.3: Examples of noise from CHiME-3 database [BMVW15]. Street noise includes passing cars, bus noise includes sound of the engine.

3.4.1 Noise

Noise is present in many real-world situations. Examples of noise include cars passing by, engine noise in a car, PC fan noise, air ducts, telephone ringing, or water running in the kitchen. Figure 3.3 shows two example spectrograms of noises. We can classify noise as either stationary (e.g. PC fan) or non-stationary (e.g. passing cars) [Loi07]. In non-stationary noise, the spectral characteristics of the noise change over time, while for stationary, they stay constant. The non-stationary noise is much more difficult to remove. According to [Loi07], we can also categorize noises based on the shape of their spectrum. Some noises are concentrated in a narrow range of frequencies (e.g. wind noise), while others are spread across the entire frequency range.

The level of noise can differ in different environments. It is usually measured in terms of dB of sound pressure level (SPL) — the relative pressure in reference to barely audible sound pressure. Low levels of noise are present, for instance, in classrooms or hospitals (around 50 dB to 55 dB SPL). On the other hand, train and airplane noise ranges around 70 dB to 75 dB SPL [Loi07]. The level of speech signal depends on the distance of the listener or recording device to the speaker. It usually ranges around 60 dB to 70 dB SPL in one-meter distance and decreases by 6 dB for every doubling of the distance. In very high levels of noise, the speech level increases (a phenomenon known as the Lombard effect [ZB11]).

Noise has been shown to significantly influence speech separation performance. For instance, on the 8 kHz WHAM database, the SI-SDR performance of the chimera++ network degrades from 11.0 dB to 5.4 dB [WAF+19]. The same trend stands also for target speech extraction, as is shown in this work (Section 4.8.4).

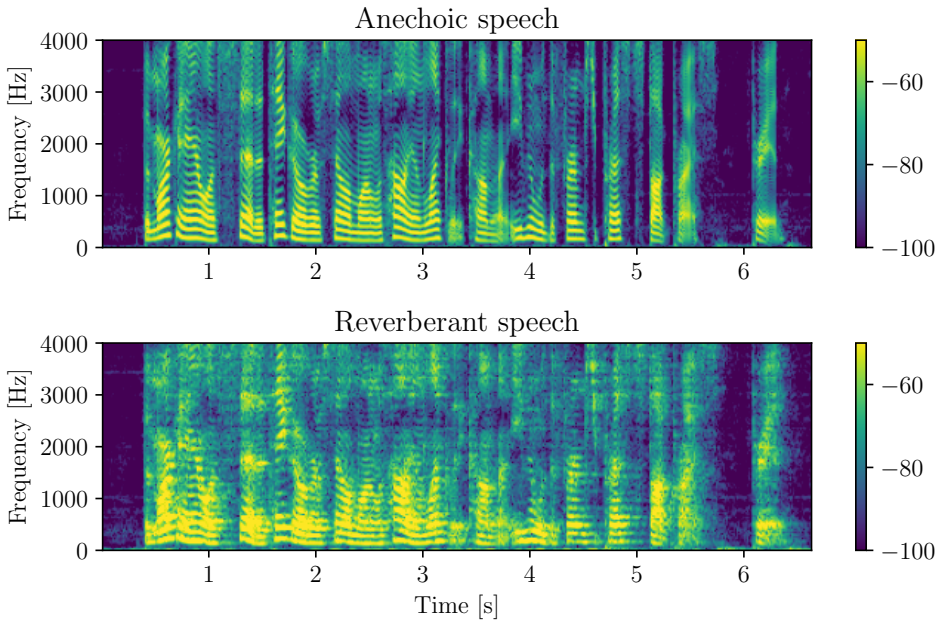


Figure 3.4: Example of clean and reverberant speech from WHAMr [MWMLR20] database. The room impulse response here is artificially generated with $T_{60} = 0.56\,\mathrm{s}$.

3.4.2 Reverberation

In most environments, the sound captured by a distant microphone contains not only noise but also reflections of the signal from walls and other objects. This effect is known as reverberation [YSD+12]. Reverberation causes the microphone to receive multiple copies of the original signal with different delays and attenuation. Due to this effect, the resulting speech signal is less intelligible and hurts the accuracy of many speech technologies, including speech separation and target speech extraction.

Figure 3.4 shows an example of clean and reverberant speech. We can see that reverberation corrupts the transitions between the phonemes and makes them less distinct. Perceptually, these effects lead to speech sounding "distant" and "echoic".

According to [YSD+12], the room impulse response (RIR) describing the reverberation, can be divided into three parts corresponding to three components of the reverberation

- *direct sound:* This part of reverberant speech is the signal received through the shortest path from the source to the microphone.
- *early reflections:* Early reflections describe the first few copies of the signal received by the microphone within the first 50 milliseconds after the direct sound. This part of reverberation has been shown to improve the intelligibility of speech.
- *late reverberation:* The late reverberation describes the mixture of many small reflections received by the microphone after the first 50 milliseconds after the direct signal. This part of reverberation is causing the degradation in intelligibility.

Figure 3.5 shows an example of a real room impulse response.

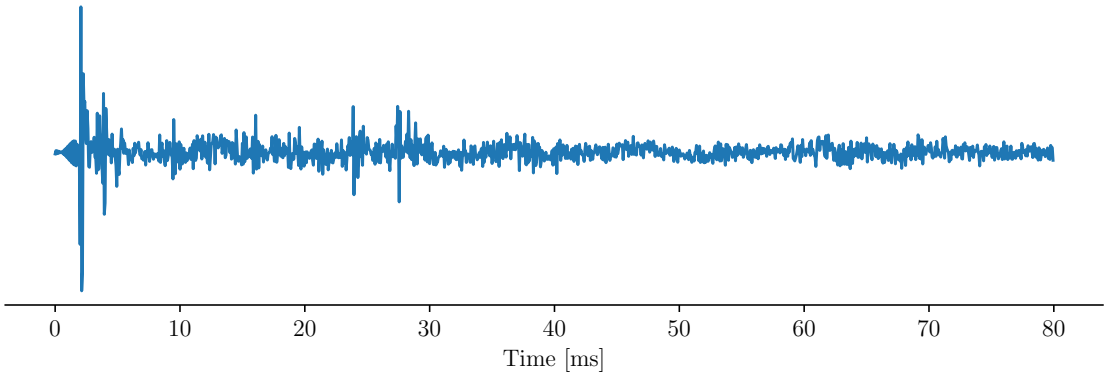


Figure 3.5: Room impulse response from BUTReverbDB [SSM+19] recorded in a lecture room.

The reverberation intensity is often described by so-called reverberation time T_{60}. This is the time required for the late reverberation to decay by 60 dB relative to the level of direct sound. In office environments, this usually ranges from 0.2 s to 1 s [YSD+12].

Reverberation significantly influences speech separation performance. In [MWMLR20], it is shown that SI-SDR performance of the ConvTasnet network reduced from 14.2 dB on anechoic data to 5.6 dB on reverberant data. In this work, we will also show how reverberation influences target speech extraction performance (Section 4.8.4).

3.4.3 Voice characteristics

Some works on single-channel speech separation report a significant gap in performance between same- and different-gender mixtures [IRC+16]. This effect was further studied in [DG19], where the authors analyze which speaker-specific characteristics influence the performance. In particular, two characteristics are considered:

- *Vocal tract length* is the length of the tube starting at the vocal cords and ending at the mouth entrance. For a particular speaker, this length is fixed and does not change. It can be estimated from a speech signal.
- *Fundamental frequency* of speech is time-varying and can be measured during voiced frames. It varies based on intonation, Lombard effect, speaker state, and can be voluntarily changed.

The study shows that difference in the fundamental frequency of two speakers in the mixture has an important effect on speech separation performance. In contrast, the difference in vocal tract length is not predictive of the performance.

Although the same study has not been done for target speech extraction, the difference in performance on mixtures of same- and different-gender speakers is also present [DOŽ+20] and is shown also in this work (Section 4.8.7). In target speech extraction, there is an additional challenge of identifying correctly the target speaker in the mixture. When the speaking style (and consequently fundamental frequency) of the target speaker in the enrollment utterance and the mixture differs, the identification is more challenging.

3.4.4 Domain mismatch

Most of the work done today in both speech separation and target speech extraction is based on data-driven methods. In these, the training dataset is first used to learn parameters of a system, which is then applied during test-time. When there is a mismatch between the distribution of training and test data, the performance sharply degrades. For instance, in the context of speech enhancement, [KTJ16] reports much better performance of neural network-based methods when they are trained on matched noise types. In [MSF+19], authors report that speech separation systems do not generalize well to recordings from realistic environments.

Another source of mismatch is caused by the fact that today's speech separation and target speech extraction systems are mostly trained on artificially mixed data. The reason for that is that it is difficult to record realistic overlapped data together with parallel single-speaker references, needed to train the systems. The artificial simulation of the data might not capture all the characteristics of real conditions, such as the Lombard effect, natural overlap ratios, or realistic room impulse responses. This can cause problems when the systems are applied in real applications.

3.5 Alternative target speaker clues

In this work, we focus on the case when the target speaker is determined based on enrollment utterance, i.e. a short segment of their speech. In some applications, it might be however possible to also use other sources of information. In this section, we give a short overview of works utilizing alternative speaker clues, namely visual, speaker activity, location, or brain signals.

Visual clue

Visual information has been utilized in the past for many problems in speech processing such as speech recognition [NYN+15], voice activity detection [AM08] or source localization [ALMD19]. Notably, some works showed that visual information can be also useful for speech separation [HC02] and speech enhancement [MTZ+21]. Some studies have also shown that visual clues help humans with focusing on a particular sound source [GCSP13]. It is thus natural to consider this clue in the context of target speech extraction.

Most of the works in this direction use video to provide the speaker clue [GSP18]. Such systems can use very similar architectures as when using the audio clue. The video of the scene is usually pre-processed with a face detector, then the face of the target speaker is cropped. In some works, only the lip region is used as the clue. The usage of video of the speaker's face is beneficial, especially in cases when the speakers have very similar voice characteristics, where audio-based clues may not be sufficient. On the other hand, in realistic videos, the face may be obstructed in some parts. For these reasons, it is particularly helpful to use multi-modal clues, combining audio and video [ACZ19, ODK+19a].

Apart from the video, there are works using only still image as the speaker clue [CCCK20]. Such works assume that it is possible to infer the speaker characteristics only from the visual characteristics of the person's face. The authors in [CCCK20] report modest SDR improvements with the image-clue, more notable in different-gender cases.

Speaker activity

As mentioned above, many works using video clues utilize only the lip region of the video. This suggests that one important piece of information to identify the target speaker is their activity, as this is something that can be easily predicted from the movement of the lips. This hypothesis was explored in our work [DŽO+21], where we used the speaker activity as the speaker clue. The activity can be estimated, for example, by using the diarization system or the visual information. The results of the study suggest that using the activity as the clue can indeed achieve similar performance as audio-based approaches.

Location clue

When the signals are recorded with multiple microphones, it is possible to infer some spatial information about the speakers in the mixture. In this work, we explore multi-channel methods in Chapter 5, but we assume not having any prior knowledge about the target speaker's position. In other works, however, location-based clues have been explored. Some works use the location to construct a fixed beamformer in the direction of the target speaker. The output of the fixed beamformer can then be post-processed into an additional feature used at the input of a neural network [CXY+18, GCZ+19]. Other works use the beamformer estimated from the location as a pre-processing [HFFHU19].

Brain signals

Some recent works [AD20] suggest using EEG signals of the listener together with the auditory attention decoding method to identify which speaker is the target the listener is attending to. This information is then used to extract the target speaker from a mixture. Using brain signals as the target speaker clue has huge potential in hearing aid applications. Although this research is only at the beginning, it is attracting much interest.

3.6 Human processing of multi-talker speech

Although the speech with competing speakers is very difficult to process for automatic algorithms, humans seem to have a remarkable ability to attend to a single source in multi-talker situations [MC12]. There is a long history of research into how the processing of multi-talker speech is performed by the human auditory and cognitive system. According to [FEDS07], there are several stages of the processing, including primitive grouping, using auditory memory, and attention. The stage of primitive grouping uses simple cues, such as harmonicity or common onset time, to group fragments of the speech signal together. Auditory memory can further improve the grouping by using previously learned patterns of speech and other signals. The mechanism of attention allows the listener to select and focus on a source of interest.

The conceptual model of speech processing in [Bro15] shows that the attention mechanism influences also the very early stages of hearing, which can thus focus on the target source. This is in line with the concept of target speech extraction as opposed to speech separation, where the selection of the target source is done only at the very end of the processing. Authors in [SCB08] argue that if the listener has prior knowledge of what distinguishes the desired source from the competing sources, the auditory system can perform a "parallel search" of the target, which is more efficient and less error-prone than "serial search", where the listener selectively samples each stream in the mixture to test whether it is the desired source.

Chapter 4

Single-channel approaches to target speech extraction

In this chapter, we focus on approaches for single-channel target speech extraction, i.e. using a signal recorded with a single microphone. Most of the principles introduced in this chapter can be however also re-used in the multi-channel case, further discussed in Chapter 5. Our focus is on neural target speech extraction and its different aspects, such as speaker embeddings, ways to inform the neural network, input and output domain, loss function, and neural network architecture. Different approaches are compared in the experiments.

4.1 Neural approaches and their aspects

Since the introduction of Deep clustering [HCLRW16] and Permutation invariant training [YKTJ17] the field of speech separation has been dominated by approaches using neural networks. This later also transferred into the emerging field of target speech extraction, which builds upon the neural methods [ŽDK+19, WMW+19]. Figure 4.1 depicts the overall scheme of the neural target speech extraction. The input of the neural network for this task is the mixed signal of multiple speakers $y(t)$, and the network is additionally informed by a speaker embedding $\boldsymbol{\lambda}_i$ extracted from the enrollment signal $e_i(t)$. The network outputs the estimated target speech signal $\hat{s}_i(t)$, which is (during the training stage) compared with the true target speech signal $s_i(t)$ using a loss function $\mathcal{L}(s_i(t), \hat{s}_i(t))$.

There are several aspects to consider when designing the target speech extraction method, namely:

1. *Speaker embedding.* How to compactly represent the speaker information from the enrollment signal?
2. *Informing the neural network.* How to use the enrollment speaker embedding to alter the behavior of the neural network to extract the target speech?
3. *Input and output domain.* In which domain to represent the input and output of the neural network?
4. *Loss function.* How to compare the estimated and the true target speech signal?
5. *Neural network architecture.* Which neural network building blocks to use to form the architecture?

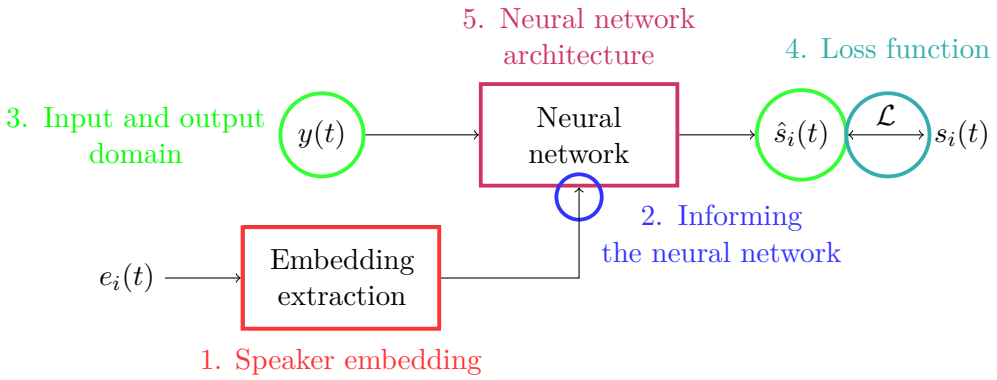


Figure 4.1: The overall scheme of the neural target speech extraction and different aspects to consider when designing the method.

We explain the different aspects and possible choices in the following sections.

4.2 Speaker embeddings

The role of speaker embedding is to accurately and compactly represent the information about the speaker in a speech signal. Speaker embeddings are heavily researched for the task of speaker verification. In neural target speech extraction, speaker embeddings are used to represent the target speaker using the information from the enrollment signal $e_i(t)$. The speaker embedding is then used to inform the neural network, guiding it towards the extraction of the target speaker. In this section, we describe several most popular choices of speaker embedding, namely i-vector, neural network-based speaker embedding, and jointly learned embedding.

I-vectors

I-vectors were for a long time considered state-of-the-art in speaker verification [MPG+20]. I-vector [DKD+10] is a fixed-length vector representing a speech utterance. The idea is to model the features extracted from the utterance using a Gaussian mixture model (GMM) with parameters constrained to a subspace. The subspace is defined by the Universal background model (UBM), i.e. GMM trained on a large amount of data from many speakers, and a total variability subspace matrix.

Formally, we assume the following model [Plc14]:

$$\mathbf{s} = \mathbf{u} + \mathbf{T}\mathbf{w}, \tag{4.1}$$

where $\mathbf{s}$ is the mean super-vector of the utterance GMM, $\mathbf{u}$ is the mean super-vector of the UBM, and $\mathbf{T}$ is a low-rank rectangular matrix representing the bases spanning the subspace. Both the matrix $\mathbf{T}$ and the UBM are pre-trained on a large amount of data. Note that no speaker labels are used in the training, each utterance is considered as a different speaker. Vector $\mathbf{w}$ is a random variable with standard normal prior distribution. The posterior distribution of $\mathbf{w}$ given the sequence of input features $\mathcal{X}$ is a Gaussian distribution $p(\mathbf{w}|\mathcal{X})$. The MAP point estimate of $\mathbf{w}$ (the mean of $p(\mathbf{w}|\mathcal{X})$) is what is called i-vector.

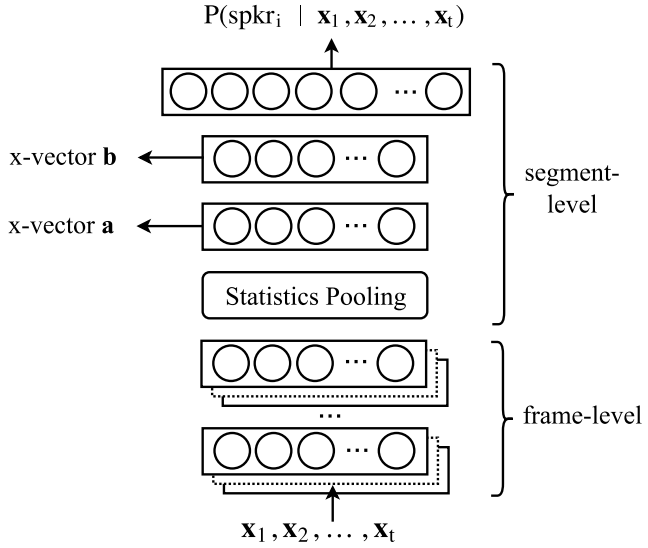


Figure 4.2: Scheme of x-vector extractor. $\mathbf{x}_1, \ldots, \mathbf{x}_t$ denote the sequence of input features, commonly MFCCs. X-vector embeddings are obtained from any layer after statistics pooling. Image from [Sny20].

An important characteristic of i-vectors is that they capture not only the speaker variability, but also the channel variability. This may be desired in some applications of target speech extraction, where we obtain the enrollment utterance in the same conditions as the mixed speech. In such a situation, the information about the channel may also help to distinguish the speakers. I-vectors have been used in several works of target speech extraction [ZLD21, XRCL19b, ŽDK+19] and are also very commonly used for speaker adaptation in acoustic modeling [KBM+11, SSNP13].

Neural network based embeddings: x-vectors, d-vectors

The current state-of-the-art systems in speaker verification are dominantly using neural network-based speaker embeddings [MPG+20]. The most commonly used NN-based speaker embeddings are x-vectors [SGRS+18], which have been also used for target speech extraction [LZY19]. In many target speech extraction works, d-vectors [WWPM18] are also used [WMW+19, MCHC20, ZHZ20].

A common idea is to train a neural network for the task of speaker classification. Such a neural network usually contains a "pooling layer" which converts a sequence of features into one vector. The pooling layer can be done, for example, by simple computation of mean and standard deviation [SGRS+18] or by applying long short-term memory networks (LSTM) [WWPM18] or attention [OKS18] layer. The pooled vector is then either classified into speaker classes [SGRS+18] or trained by other loss functions to be speaker discriminative [WWPM18]. For extraction of the embedding, the final layers are discarded, and speaker embedding is obtained as a vector of activations in one of the last layers in the network. The basic scheme of x-vector extractor is shown in Figure 4.2.

Since the neural network is trained for speaker classification or a related task, the embeddings are usually highly discriminative and most other variability (like channel or content) is discarded. Another advantage of this class of embeddings is that the models are usually trained on large corpora with many speakers, noises, and other variations, which makes the

embedding extractors very robust. Such trained models are often publicly available, and the embeddings can be thus readily used for the task of target speech extraction.

Jointly learned embeddings

The neural network-based embeddings, such as x-vectors, are designed and trained for the task of speaker classification. Although this causes them to contain speaker information, it is questionable whether the same representation is optimal for the task of target speech extraction. A way to obtain embeddings that are closer to optimal is to train the neural embeddings extractor together with the neural network performing the target speech extraction. This has been proposed in [VWŽ+16] for speaker adaptation, in our paper [ŽDK+17a] for target speech extraction and used in numerous other publications (e.g. [XCY+19a, XRCL19a, HLZ20]).

The neural network performing the speaker embedding extraction takes the enrollment utterance $e_i(t)$ as the input and, in most cases, contains a pooling layer converting the frame-level features into one vector, similar to the embedding extractors discussed above. The neural network is trained together with the main neural network using a common objective function. Possibly, a second objective function can be used on the embeddings to further improve their speaker discriminability.

As mentioned above, the advantage of such embeddings is that they are trained directly for the task of target speech extraction and thus contain the information important for this task. On the other hand, the pre-trained embedding extractors are often trained on larger corpora and may be more robust. A possible middle ground could be to take a pre-trained embeddings extractor and fine-tune it jointly with the target speech extraction task. This has, however, to our knowledge, not been done yet.

4.3 Informing the neural network

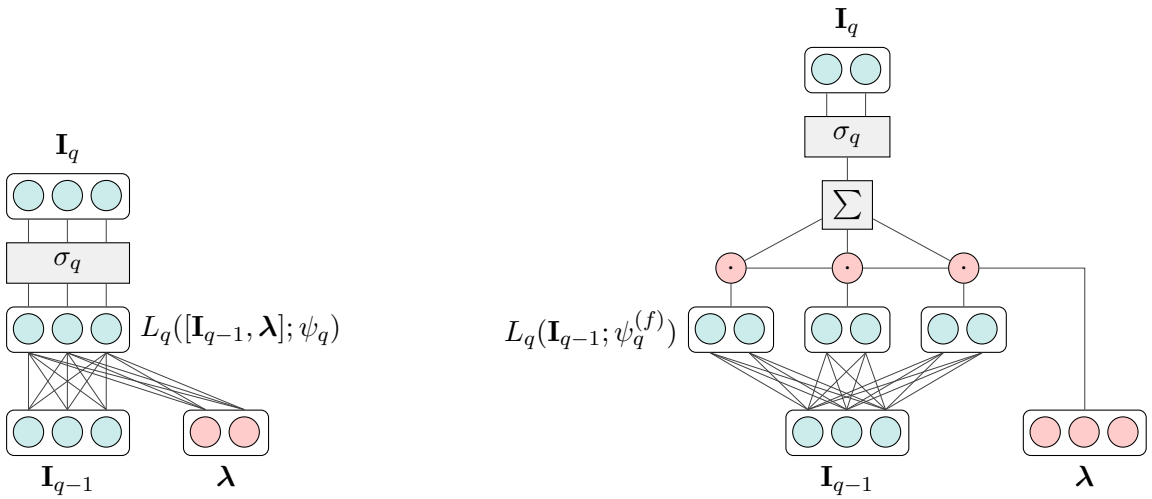
Given an embedding extracted from the enrollment utterance, the neural network for target speech extraction should extract the target speaker. There are different ways in which the embedding can be passed to and utilized by the network. Here, we describe several most popular ways, namely concatenation-based, multiplication-based, factorized layer, and attention-based schemes. The schemes are depicted in Figures 4.3, 4.4 and 4.5.

Concatenation-based

Possibly the simplest way to pass the embedding to the neural network is to concatenate it to the input of one of the layers (the adaptation layer). We can express the neural network processing as

$$\mathbf{I}_k = \begin{cases} \sigma_k(L_k([\mathbf{I}_{k-1}, \boldsymbol{\lambda}]; \psi_k)) & \text{for } k = q, \\ \sigma_k(L_k(\mathbf{I}_{k-1}; \psi_k)) & \text{for } k \neq q, \end{cases} \tag{4.2}$$

where $\mathbf{I}_k$ is the input to the kth layer, q is the index of the adaptation layer, $L_k(\mathbf{I}_{k-1}, \psi_k)$ is the transformation computed by the kth layer parameterized by ψ_k, and σ_k is an activation function. For example, with fully connected layers, $\psi = \{\mathbf{W}, \mathbf{b}\}$ and $L(\mathbf{I}, \psi) = \mathbf{W}\mathbf{I} + \mathbf{b}$, where $\mathbf{W}$ is a weight matrix and $\mathbf{b}$ is a bias vector. In the case of an affine layer, the concatenation is equivalent to adaptation of the biases of the layer q (since $L_q([\mathbf{I}_{q-1}, \boldsymbol{\lambda}]; \psi_q) =$



(a) Concatenation.

(b) Factorized layer.

Figure 4.3: Two options for informing the neural network about the target speaker: concatenation and factorized layer. Blue color denotes hidden values in the main network, red values denote the speaker embedding.

$\mathbf{W}_q[\mathbf{I}_{q-1}, \boldsymbol{\lambda}] + \mathbf{b} = \mathbf{W}_q^{(I)}\mathbf{I}_{q-1} + \mathbf{W}_q^{(\lambda)}\boldsymbol{\lambda} + \mathbf{b}$ and $\mathbf{W}_q^{(\lambda)}\boldsymbol{\lambda}$ is not dependent on the input). The processing is depicted in Figure 4.3a.

The concatenation is very simple and has been heavily used for the task of speaker adaptation in acoustic modeling. However, it may be a too weak scheme for target speech extraction, as it modifies only the bias parameters. This has been shown in our earlier work [ŽDK+19], where concatenation performed much worse than other methods. Different works however applied this scheme successfully (e.g. [WMW+19, LZY19, XRCL19a]). The suitability of the scheme may be thus dependent on the particular data and architecture.

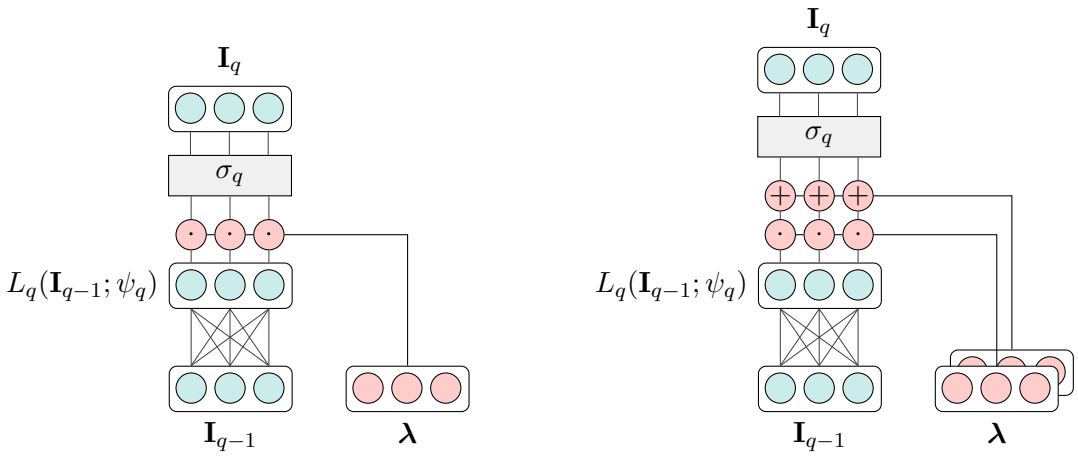
Factorized layer

One option to adapt larger number of parameters in the network and so to influence its behavior stronger, is to use factorized layer. This method was first proposed for speaker adaptation in acoustic modeling [DKHN15, WG15] and later applied to the task of target speech extraction [ŽDK+17b, HFFHU19]. The idea is to replace one layer in the network with a set of sub-layers. The output of such factorized layer is then a weighted combination of outputs of all sub-layers. Using different weights in the combination causes the network to extract different speakers. Following the previous notation and denoting the index of the factorized layer q and the number of sub-layers as F, the network processing is defined as

$$\mathbf{I}_k = \begin{cases} \sigma_k\left(\sum_{f=0}^{F-1} \lambda_f L_k(\mathbf{I}_{k-1}; \psi_k^{(f)})\right) & \text{for } k = q, \\ \sigma_k(L_k(\mathbf{I}_{k-1}; \psi_k)) & \text{for } k \neq q, \end{cases} \tag{4.3}$$

where $\psi_k^{(f)}$ are the parameters of the fth sublayer. The processing is depicted in Figure 4.3b.

If the layer L_q is an affine transform, the weighted combination can equivalently be done directly on the parameters of the layer. We can then see the factorized layer as defining



(a) Multiplication.

(b) Multiplication-addition.

Figure 4.4: Two options for informing the neural network about the target speaker: multiplication and multiplication-addition. Blue color denotes hidden values in the main network, red values denote the speaker embedding.

a subspace in the parameter space, where different coordinates in the subspace correspond to different target speakers.

The factorized layer method enables a strong influence of the speaker embedding on the behavior of the network. Previous works have reported it to perform well in target speech extraction [ŽDK+19]. However, this method is quite computationally and memory expensive due to the large number of sub-layers.

Multiplication-based

An alternative approach was introduced for speaker adaptation [SR14] and later adopted for target speech extraction [ŽDK+19, DŽO+19, HXS+20] where the outputs of a layer are element-wise multiplied with a speaker embedding. Similar to factorized layer method, this has a strong effect on the behavior of the neural network. However, it is computationally simpler. In this case, the processing performed by the neural network is:

$$\mathbf{I}_k = \begin{cases} \sigma_k(\boldsymbol{\lambda} \odot L_k(\mathbf{I}_{k-1};\psi_k)) & \text{for } k = q, \\ \sigma_k(L_k(\mathbf{I}_{k-1};\psi_k)) & \text{for } k \neq q, \end{cases} \tag{4.4}$$

where $\odot$ is element-wise multiplication. A similar conditioning scheme called Feature-wise linear modulation (FiLM) was also proposed for visual reasoning [PSDV+18]. FiLM uses a bias vector on top of the multiplication.

$$\mathbf{I}_k = \begin{cases} \sigma_k\left(\boldsymbol{\lambda}^{(mul)} \odot L_k(\mathbf{I}_{k-1};\psi_k) + \boldsymbol{\lambda}^{(add)}\right) & \text{for } k = q, \\ \sigma_k(L_k(\mathbf{I}_{k-1};\psi_k)) & \text{for } k \neq q. \end{cases} \tag{4.5}$$

Both options are depicted in Figure 4.4.

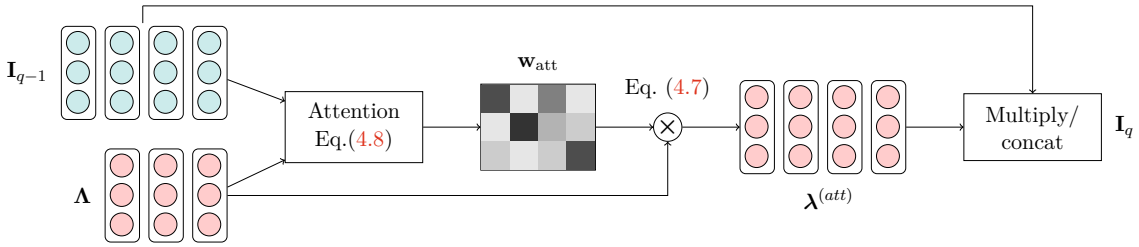


Figure 4.5: Attention-based scheme for informing the neural network. Blue color denotes hidden values in the main network, red values denote the speaker embedding. Multiply/concat block denotes multiplication or concatenation, as shown in Figures 4.4 and 4.3a.

Multiplication-based method combine strength of the adaptation with simplicity and were shown by some works [ŽDK+19] to be a good choice for target speech extraction.

Attention-based

All the previous methods for informing the neural network apply the same speaker embedding at each time frame of the mixed speech. However, depending on the content of the mixed speech in the current time frame, different information from the enrollment utterance might be more important. For example, if the speakers in the mixed speech pronounce vowels in the current time, we could focus more on parts of the enrollment utterance where the target speaker pronounces vowels too. This might be achieved by using an attention mechanism, as was proposed in [XCY+19b].

The processing with attention-mechanism is performed as follows:

$$\mathbf{I}_k = \begin{cases} \sigma_k(\boldsymbol{\lambda}^{(att)} \odot L_k(\mathbf{I}_{k-1}; \psi_k)) & \text{for } k = q, \\ \sigma_k(L_k(\mathbf{I}_{k-1}; \psi_k)) & \text{for } k \neq q. \end{cases} \tag{4.6}$$

with

$$\boldsymbol{\lambda}^{(att)} = \mathbf{w}_{\text{att}}^{\mathsf{T}} \mathbf{\Lambda} \tag{4.7}$$

$$\mathbf{w}_{\text{att}} = \text{softmax}(\mathbf{\Lambda} L_k(\mathbf{I}_{k-1}, \psi_k)^{\mathsf{T}}). \tag{4.8}$$

In this case, the embedding $\mathbf{\Lambda}$ is dynamic embedding, $\mathbf{\Lambda} \in \mathbb{R}^{N_e \times D}$, where N_e is the number of time frames in the enrollment utterance and D is the dimension of the embedding. In Equation 4.8, the similarity between the dynamic speaker embeddings and the representations of the mixed speech is computed. This leads to attention weight $\mathbf{w}_{\text{att}} \in \mathbb{R}^{N_e \times N}$, where N is the number of frames in the mixed speech. These weights are then applied to the dynamic speaker embedding $\mathbf{\Lambda}$ to create the final weighted dynamic speaker embedding $\boldsymbol{\lambda}^{(att)} \in \mathbb{R}^{N \times D}$. This is depicted in Figure 4.5. Due to the attention weights, the final speaker embedding contains more information from the parts of the enrollment utterance which are similar to the current mixed representations. In Equation 4.6, we use the multiplication scheme to apply the embedding. This could however be also replaced with concatenation or factorized layer method.

Note that the attention-based mechanism blends the stages of embedding extraction and informing the neural network. As such, it slightly violates the overall scheme that

we introduced in Section 4.1, where the embedding extraction and application are strictly separated. Due to this, the attention scheme is impossible to combine with i-vector or x-vector embeddings, which do not provide the dynamic information necessary for attention.

4.4 Input and output domain

There are different representations of the speech signal that can be used on the inputs and at the output of the neural network performing target speech extraction. Here, we describe the two most commonly used, namely time-domain and frequency-domain representations. In the field of target speech extraction, higher-level features (such as Mel frequency cepstral coefficients) are rarely seen, because of their low frequency resolution. We thus omit such types of features in this section.

Frequency-domain

A natural representation when analyzing speech signals is the frequency domain as speech is both produced by a periodic process in the vocal tract and frequency-analyzed by the human ear. The early works in both speech separation and target speech extraction often used short-time Fourier transform (STFT) to represent the mixed speech signal. The input of the network is then usually the magnitude of the mixed-signal $|Y(n, f)|$ or the log-magnitude $\log |Y(n, f)|$. The output can then be either in the same domain as the input, or very commonly the neural network predicts a mask $M(n, f)$. The mask contains values between 0 and 1 and the magnitude STFT of the final extracted speech can be then obtained by element-wise multiplying the mask with the magnitude mixture $M(n, f) \cdot |Y(n, f)|$.

In all of these cases, the neural network does not predict the phase of the output. Instead, the phase of the mixture is usually used to reconstruct the extracted signal. This is not optimal and some works have shown that using complex representation instead of working with magnitude only can lead to better performance [WWW16].

Time-domain

In 2018, the publication of ConvTasnet [LM19] (Section 2.5.4) for speech separation has shown that using the time-domain representation of speech directly can significantly improve the performance. This advance was later transferred also into target speech extraction works [XRCL19b, DOŽ+20, ZGS20]. In these, the first layer of the neural network is the so-called encoder, which contains a single convolutional layer. The convolutional layer acts on a time-domain signal and transforms it into an encoded representation $\mathbf{S}^{(enc)} \in \mathbb{R}^{N \times K}$, where K is the number of filters in the convolution and N is the number of resulting frames, determined by the window shift in the convolution. Such layer can in theory implement Fourier transform and thus the representations can be equivalent to STFT. In this case, the parameters are however learned during the training of the network and can thus be optimized for the task. The decoder on the other hand reconstructs the output extracted signal from the estimated encoded representations and is usually implemented by a transposed convolutional layer with parameters also learned during the training.

As analyzed in [HJB+20], the performance improvement when transitioning from frequency-domain approaches to time-domain ConvTasnet comes from several sources. First, the network can use and predict phase information, which has been discarded in approaches working with magnitude-STFT. Second, the time-domain loss function used in ConvTasnet

is a closer match with common evaluation metrics and as such can lead to better results in these metrics[1]. Finally, it has been observed that the fact, that the window size used in the ConvTasnet encoder is much smaller than typical STFT window sizes, has a big influence on the results.

4.5 Loss function

Training of the neural network for target speech extraction requires computing a loss function that measures the discrepancy between the neural network prediction and ground truth target speaker signal. The choice of the loss function is very connected with the domain that the neural network works with. In this section, we summarize common loss functions for both frequency and time domains.

Loss function in frequency domain

When the neural network works with frequency-domain signals, there are two options of what it can predict: either the time-frequency mask $M(n, f)$ or directly the frequency-domain target speech, most commonly in magnitude STFT $|\hat{S}(n, f)|$. In the case of predicting time-frequency masks, the loss function can be computed either directly from the estimated mask, or masked signal $M(n, f) \cdot Y(n, f)$.

First, we describe the loss functions acting directly on the predicted mask. The loss function quantifies the discrepancy between the estimated mask and an "ideal mask". There are several options of how the ideal mask can be defined [EHWLR15]:

- **Ideal binary mask** (IBM) is defined as

$$M_i^{(ideal)}(n, f) = \delta\left(\left|\sum_{j \neq i} S_j(n, f) + N(n, f)\right| < |S_i(n, f)|\right), \tag{4.9}$$

  i.e. it is 1 in T-F bins with dominant target source and 0 in T-F bins with dominant interference. Such mask maximizes the signal-to-noise ratio (SNR) of the masked mixture $M(n, f) \cdot Y(n, f)$ under the constraint of binary mask $M(n, f) \in \{0, 1\}$.

- **Ideal amplitude mask** (IAM) is defined as

$$M_i^{(ideal)}(n, f) = \frac{|S_i(n, f)|}{|Y(n, f)|}. \tag{4.10}$$

  This mask leads to maximum SNR under the assumption of equal phases of the target source and the mixture $\angle S_i(n, f) = \angle Y(n, f)$. Applying the ideal amplitude mask also leads to perfect reconstruction of the magnitude of the source.

- **Ideal phase-sensitive mask** (IPSM) is defined as

$$M_i^{(ideal)}(n, f) = \frac{|S_i(n, f)|}{|Y(n, f)|} \cos(\angle Y(n, f) - \angle S_i(n, f)). \tag{4.11}$$

  This is an optimal mask in terms of SNR under the constraint that the mask is real-valued (as opposed to a complex-valued mask).

[1] Such improvement would not be very meaningful by itself if it does not transfer to other evaluation metrics or down-stream performance. It however has been shown that the match of the loss function and the metric is not the only source of improvement in ConvTasnet.

The above list shows the most popular choices of ideal masks, however, it is not exhaustive. In the literature, other definitions of ideal masks can be found, such as ideal ratio mask, Wiener-like mask, and ideal complex mask [EHWLR15].

Using one of the definitions of the ideal mask, the loss function can be computed as the mean-square error between the estimated and the ideal mask

$$\mathcal{L}(\hat{\mathbf{M}}_i, \mathbf{M}_i^{(ideal)}) = \sum_{n,f} |\hat{M}_i(n,f) - M_i^{(ideal)}(n,f)|^2. \tag{4.12}$$

In the case of ideal binary mask, the task can be also posed as a binary classification problem and binary cross-entropy (CE) can be used

$$\mathcal{L}(\hat{\mathbf{M}}_i, \mathbf{M}_i^{(ideal)}) = \sum_{n,f} M_i^{(ideal)}(n,f) \log \hat{M}_i(n,f) + (1 - M_i^{(ideal)}(n,f)) \log(1 - \hat{M}_i(n,f)). \tag{4.13}$$

The second category of loss functions in the frequency domain are loss functions computed directly from the predicted signal in the frequency domain, most commonly the magnitude STFT $|\hat{S}_i(n,f)|$. Note that these loss functions can be applied not only when the network predicts the magnitude directly, but also when the predicted value is a mask, by setting $\hat{S}_i(n,f) = \hat{M}_i(n,f) \cdot |Y(n,f)|$. The most popular loss function, in this case, is mean-square error (MSE) on the magnitude

$$\mathcal{L}(|\hat{\mathbf{S}}_i, \mathbf{S}_i|) = \sum_{n,f} |\hat{S}_i(n,f) - S_i(n,f)|^2 \tag{4.14}$$

or its phase-sensitive counterpart (PS-MSE)

$$\mathcal{L}(|\hat{\mathbf{S}}_i, \mathbf{S}_i|) = \sum_{n,f} |\hat{S}_i(n,f) - S_i(n,f) \cos(\angle Y(n,f) - \angle S_i(n,f)|^2. \tag{4.15}$$

Loss function in time domain

The rise of neural networks predicting signals directly in the time-domain [LM19] got reflected also in the loss functions, that are now often computed on the time-domain signals. It is noteworthy that time-domain loss functions can be applied also when the neural network predicts frequency-domain signal (or mask) by using differentiable inverse STFT on the output during the training. Vice-versa, time-domain networks can also be optimized with frequency-domain loss functions by the usage of differentiable STFT. In the literature, it is however much more common to match the domain of the output and the loss function.

The most popular loss function in the time-domain is scale-invariant signal to distortion ratio (SI-SDR), which exactly matches the evaluation metric defined in Section 2.3, Equation (2.5):

$$\begin{aligned} \text{SI-SDR}(s(t), \hat{s}(t)) &= 10 \log_{10} \frac{\sum_t |\alpha s(t)|^2}{\sum_t |\alpha s(t) - \hat{s}(t)|^2} \\ \text{with} \quad \alpha &= \operatorname*{argmin}_{\alpha} \sum_t |\alpha s(t) - \hat{s}(t)|^2. \end{aligned} \tag{2.5 revisited}$$

Some works also propose using simple signal-to-noise ratio (SNR)

$$\text{SNR}(s(t), \hat{s}(t)) = \frac{\sum_t |s(t)|^2}{\sum_t |s(t) - \hat{s}(t)|^2}, \tag{4.16}$$

suitable when the scale of the estimated signal is of interest.

Table 4.1: Selected approaches for target speech extraction and their classification into categories.

Approach	Speaker embedding	Informing method	I/O domain	Loss function	NN architecture
SpeakerBeam [ŽDK+17b]	posterior	factorized	frequency	mask CE	BLSTM
[ŽDK+17a]	joint	factorized	frequency	mask CE	BLSTM
[ŽDK+19]	joint	multiplication	frequency	PS-MSE	BLSTM
[DOŽ+20]	joint	multiplication	time	SI-SDR	ConvTasnet
VoiceFilter [WMW+19]	d-vector	concatenation	frequency	MSE	CNN+LSTM
Speaker Inventory [XCY+19a]	joint	attention	frequency	MSE	BLSTM
Deep extractor [WCS+18]	joint	concatenation	frequency	MSE	LSTM

4.6 Neural network architecture

Target speech extraction models can easily re-use the advances in neural network architecture for speech separation introduced in Section 2.5. As such, in earlier works, bidirectional LSTM (BLSTM) or LSTM [ŽDK+19, XCY+19a, WCS+18] architectures were widely used. More recently, ConvTasnet architecture became a popular choice [XRCL19b, ZHZ20, GXW+20] (Section 2.5.4) and some works also employ dual-path RNN [HXS+20, DMS+21] (Section 2.5.5).

The additional concern in target speech extraction is the architecture of the auxiliary network in the case of jointly learned embeddings and the position where the network is informed about the target speaker. For the auxiliary network, mostly very simple architectures have been used. In earlier works, the auxiliary network used either a few fully-connected layers with a non-linearity or one recurrent layer. With the arrival of ConvTasnet, time-dilated convolutional layers have also been commonly applied. For the position where information about the target speaker is brought in, earlier layers in the network are usually preferred. In BLSTM models, the position was usually chosen to be the second layer of the model. In the case of ConvTasnet, the additional information is usually applied after the first repetition of the convolutional blocks. Some works apply the information at multiple positions, for instance at every layer of the network.

4.7 Existing approaches

After our publication [ŽDK+17b], many works devised approaches tackling the problem of target speech extraction. In this section, we introduce several representative ones. First, we cover our approach "SpeakerBeam" presented in [ŽDK+17b, ŽDK+17a, ŽDK+19, DOŽ+20]. Next, we introduce VoiceFilter, Speaker inventory, and Deep extractor. In Table 4.1, we describe these approaches in terms of the categories laid out earlier in this chapter. Note that each of the approaches has a slightly different focus (e.g. ASR performance, or TSE performance with very short enrollment). For this reason, the published results were obtained on different datasets and are thus very difficult to compare.

4.7.1 SpeakerBeam

SpeakerBeam was first introduced in [ŽDK+17b]. Although it was applied to multi-channel signals, the core of the method is applicable also to the single-channel scenario. The work introduced the idea of using speaker embedding to inform the neural network and extract the target speaker's speech. This was compared to a strategy where a separate neural network or a separate layer is trained for each target speaker [DTX+14, ZW16]. Such schemes can work only for a closed set of speakers. The speaker embedding used in [ŽDK+17b] was a vector of posteriors obtained by neural network classification of the enrollment utterance into classes formed by speakers from the training set. This is resembling methods using x-vectors or d-vectors, which are also obtained using external speaker classifiers.

In [ŽDK+17a], SpeakerBeam was extended with jointly learned speaker embeddings, which were also compared to i-vectors. The work also presented automatic speech recognition of the extracted signals. This was later extended in [ŽDK+18] to joint training with an ASR system. In [ŽDK+19], the previous work was summarized and a detailed comparison with speech separation methods was provided. This work also applied a multiplication scheme for informing the neural network in addition to factorized layer. Finally, in [DOŽ+20], SpeakerBeam was extended to time-domain using ConvTasnet architecture.

As stated in Section 1.2, most experiments in this thesis follow the work done in the SpeakerBeam papers. Moreover, we test on more recent datasets and provide a more detailed analysis of different aspects of the method.

4.7.2 VoiceFilter

VoiceFilter [WMW+19] is an approach similar to SpeakerBeam with differences in the used architecture, speaker embedding, and way of informing the neural network. It is one of the first TSE works successfully applying concatenation to inform the neural network. The evaluation in [WMW+19] considers both TSE performance and ASR performance. The experiments show that VoiceFilter is able not only to improve the ASR performance in multi-speaker cases, but also to preserve the good performance when the input is single-speaker. This is an important property for using the model in practice. One interesting aspect of the experiment design in [WMW+19] is that the mean SDR of the original mixtures is 10.1 dB, i.e. the target speaker is often dominant.

The VoiceFilter was later extended in several subsequent works [WMS+20, RWL+21b, RWL+21a]. In [WMS+20], the authors focus on improving streaming speech recognition, which requires designing the model to have minimal impact on CPU, memory, battery, and latency. In [RWL+21b], VoiceFilter was used as a front-end for speaker verification, as a part of streaming key phrase detection. In [RWL+21a], VoiceFilter was extended to handle multi-speaker enrollment, similarly to the speaker inventory approach introduced in the next section.

4.7.3 Speaker inventory

The Speaker inventory approach was applied to speech separation [WCX+19] but also to target speech extraction [XCY+19a] problem. In contrast with conventional TSE, the speaker inventory approach does not inform the network only about the target speaker, but also about the potential interfering ones. This could be used for example in the scenario of meeting transcription, where the list of possible speakers is available. The approach uses

attention to identify the interfering speakers actually present in the mixture among the provided speaker inventory.

The results presented in [XCY+19a] clearly show that having the enrollment of the interfering speaker helps the network to better extract the target speaker's speech. In case the information about the interfering speaker is not precise, i.e. inventory of possible interfering speakers is provided, the gain is smaller but still significant.

4.7.4 Deep extractor

The Deep extractor approach [WCS+18] takes a very different approach for TSE which builds upon Deep Attractor networks (Section 2.5.2). The neural network in the Deep extractor outputs a mask corresponding to the target speaker in three steps:

1. The Neural network computes one embedding for each time-frequency bin. The embedding should encode the information about the speaker which is dominant in the T-F bin. This step is the same as in Deep clustering and Deep attractor approaches.

2. The embedding space is transformed by the second part of the neural network, which is informed by the speaker embedding extracted from the enrollment utterance. This should transform the embeddings in such a way, that the bins dominated by the target speaker are moved to a particular part of the embedding space.

3. The mask is computed based on the similarity of the embeddings to a "canonical extractor" embedding. The position of the canonical extractor is determined based on the positions of the target speaker embeddings in the training data.

The evaluation of Deep Extraction focused on the use-case of spoken commands. Experiments in [WCS+18] showed that the approach is very effective with very short enrollment utterances (0.9 seconds on average).

4.8 Experiments

4.8.1 Datasets

For experiments in this chapter, we make use of three datasets: WSJ0mix [HCLRW16], WHAM [WAF+19] and WHAMr [MWMLR20]. All three datasets were created for experiments with speech separation and have been widely used for experiments in the literature. We chose these datasets to cover all clean, noisy, and noisy-reverberant conditions. Further, these three datasets are based on the same set of mixed utterances and are therefore useful to isolate the effects of noise and reverberation on the performance. For our experiments with target speech extraction, we extend the datasets by assigning each test mixture an enrollment utterance from the same dataset. The enrollment utterance is always chosen to be a different from the utterance in the mixture. In all three datasets, the mixtures are artificially created from single-speaker data. This is a common choice in the literature because it enables supervised training and direct evaluation with clean references.

WSJ0mix

The WSJ0-2mix [HCLRW16] contains mixtures of two speakers at signal-to-noise ratios between 0 dB and 5 dB. It consists of a training set, a validation set, and an evaluation set

of 30, 10, and 5 hours, respectively. For training and validation sets, the mixed utterances were randomly selected from the `si_tr_s`, while for evaluation set, the utterances were taken from `si_dt_05` and `si_et_05` parts of Wall Street Journal (WSJ0) [PB92]. In total, the training set contains 20000 mixtures from 101 speakers, the cross-validation set contains 5000 mixtures from the same 101 speakers and the evaluation set contains 3000 utterances from 18 speakers (unseen in the training). The WSJ0-3mix [IRC+16] contains three-speaker mixtures analogous to WSJ0-2mix in terms of the amounts of data, numbers of speakers, and WSJ0 sets from which the utterances are selected. All data are at 8 kHz sampling rate for consistency with previous studies. We use "min" versions of the datasets, where the mixture is cut to the length of the shortest utterance (for consistency with previous work). For generating the WSJ0-4mix dataset with 4 overlapping speakers, we used lists provided by [NAW20][2]. In addition, we use our own dataset WSJ0-2mix-long. With this dataset, we aim to test the performance of the methods on longer mixtures. We created the dataset by concatenating several utterances for both speakers in the mixture. We created several versions of the dataset with different lengths, by concatenating 2 to 5 utterances for each speaker. The utterances were first used in their full length, but the final mixture was cut to the length of the shorter one of the concatenated signals.

WHAM

WSJ0 Hipster Ambient Mixtures (WHAM) [WAF+19] is a dataset based on WSJ0-2mix, extended by adding background noises. It was created to help move the field of speech separation towards more realistic scenarios. The added background noises were recorded in urban environments such as coffee shops, restaurants, bars, office buildings, or parks in the San Francisco Bay Area. In total, there were 44 different locations, which were then assigned to training, validation, or test split. The noises were recorded with two microphones at 15 cm to 17 cm distance. During post-processing, all noise recordings with intelligible speech were removed. The original noise recordings have 48 kHz sampling rate, but were down-sampled to 8 kHz and 16 kHz. In this work, we use the 8 kHz version, as in most of the previous literature.

WHAMr

WHAMr [MWMLR20] dataset augments WHAM by adding reverberation. It was created to advance the research on speech separation on reverberant speech, as it is more common in real-world scenarios than anechoic speech. The room impulse responses convolved with the original data were created by pyroomacoustics[3] [SBD18]. The reverberation times and room sizes were randomly sampled in ranges approximating domestic and classroom environments. This was chosen to be similar to the reverberation condition in WHAM noises, which were recorded spatialized. The dataset contains several versions with different levels of distortion, i.e. anechoic clean, anechoic noisy, reverberant clean, reverberant noisy. In this work, we use the most difficult reverberant noisy condition.

4.8.2 Speech separation benchmarks

We first overview the published results of several representative speech separation approaches to put our baseline into context. Table 4.2 shows the results on the three mentioned

[2]WSJ0-4mix lists https://enk100.github.io/speaker_separation
[3]pyroomacoustics https://github.com/LCAV/pyroomacoustics

datasets. Note that many of the published papers report results on WSJ0-2mix only. Also, the literature is often inconsistent in terms of the evaluation metric. Some works report SI-SDR, while others SDR (Section 2.3). However, results computed using these two metrics are usually very similar and roughly comparable.

In this work, we use ConvTasnet trained for speech separation as the baseline for our target speech extraction experiments, that are done using the ConvTasnet architecture, too. We use the implementation of ConvTasnet provided in Asteroid[4] [PCC+20] toolkit, with our own hyperparameter tuning. This implementation provides slightly better results than the ones originally published in ConvTasnet paper [LM19], as reported in Table 4.2 as *ConvTasnet (here)*. We have chosen ConvTasnet for our experiments because of its available implementation and faster run-time than other methods. Our methods are fully compatible with architectures such as DP-RNN [LCY20] and Mulcat [NAW20] and could be applied with these for further improvement of the overall results.

4.8.3 Configuration

The configuration described in this section applies to all experiments in this chapter unless specified otherwise (e.g. when we explicitly explore the effect of certain setting).

ConvTasnet

For both speech separation and target speech extraction, ConvTasnet model was used (Sections 2.5.4 and 4.6). The choice of hyper-parameters follows the setting used in Asteroid toolkit, specified here in Table 4.3. We train the network for 200 epochs, with batch size 6 and SI-SDR loss. As the input of the network, a randomly selected 3-second chunk of each mixture is used. We use Adam optimizer [KB14] with learning rate 0.001. The learning rate is halved if the validation loss does not improve for 10 consecutive epochs.

Target speech extraction

The default setting of the target speech extraction experiments is multiplication-based method to inform the network (Section 4.3) and jointly learned speaker embeddings (Section 4.2). The speaker embedding is used to inform the main network after the first repeat of ConvTasnet (i.e. after 8 convolutional blocks). The auxiliary network learning the speaker embeddings consists of 1 repeat of ConvTasnet with other hyper-parameters identical to the main network (Table 4.3). The encoder parameters are not shared between main and auxiliary network (from our experience, this does not have a big effect). In case of using external speaker embeddings (i-vectors or x-vectors), we transform them by two fully-connected layers with Leaky ReLU activation after the first one.

During training, we use 0.5 s randomly chosen segments of the enrollment utterance as input of the auxiliary network. Analogously, the x-vectors are extracted from 0.5 s segments during training. For i-vectors, we chose to extract them from the full utterances, as this led to better performance.

For experiments with factorized layer, we chose 30 sub-layers. The dimension of the speaker embedding (output of auxiliary network) depends on the used method to inform the neural network. This is 128 in case of multiplication and attention-multiplication (corresponding to the number of channels in bottleneck), 30 in case of factorized layer (corre-

[4] Asteroid toolkit https://github.com/asteroid-team/asteroid

Table 4.2: Selected speech separation results in the literature. *ConvTasnet (here)* denotes our version of ConvTasnet used as a baseline in our work. Δ denotes improvement in the respective metric over the original mixture.

	WSJ0-2mix		WHAM		WHAMr	
	ΔSDR	ΔSI-SDR	ΔSDR	ΔSI-SDR	ΔSDR	ΔSI-SDR
DC	10.8 [IRC+16]	-	-	-	-	-
PIT+DC	11.5 [WRH18]	-	9.9 [WAF+19]	-	-	
ConvTasnet	15.6 [LM19]	15.3 [LM19]	-	-	-	8.3 [MWMLR20]
DP-RNN	19.0 [LCY20]	18.8 [LCY20]	-	-	-	-
Mulcat	-	20.1 [NAW20]	-	15.2 [NAW20]	-	12.2 [NAW20]
ConvTasnet (here)	16.8	16.6	14.0	13.7	9.7	10.6

Table 4.3: Hyper-parameters of the used ConvTasnet architecture.

Encoder/decoder	
Number of filters	512
Length of filters	16 samples
Stride of filters	8 samples
Separator	
Number of repeats	3
Number of convolutional blocks in each repeat	8
Number of channels in convolutional blocks	512
Number of channels in bottleneck	128
Number of channels in skip connection	128
Kernel size on convolutional blocks	3
Mask activation	ReLU

sponding to the number of sub-layers) and 128 in case of concatenation (this is chosen as hyper-parameter).

I-vectors

The i-vectors used in our experiments are obtained using i-vector extraction scripts in Kaldi toolkit[5] [PGB+11]. Both the Universal Background Model (UBM) and i-vector extractor are trained on the single-speaker utterances in the training set of WSJ0-2mix, WHAM, and WHAMr, respectively, for experiments on these datasets. The input features are high-resolution Mel-frequency ceptral coefficients (MFCC) of dimension 40. The UBM is composed of 256 Gaussians and the dimension of the i-vectors is 100.

X-vectors

For x-vectors, we used x-vectors extractor provided with VBx recipe[6] [LPDB22]. The x-vector extractor has ResNet101 [HZRS16, ZWS+19] architecture consisting of one 2-D convolutional layer, followed by 4 standard ResNet blocks [HZRS16]. The architecture is further described in Table 4.4. The inputs of the x-vector extractor are 64-dimensional log Mel filterbank features extracted from 25 ms windows with 10 ms shift. Mean and standard-deviation pooling are used to obtain 256-dimensional x-vectors. The model was trained using additive angular margin loss on the following datasets: VoxCeleb1 [NCZ17], VoxCeleb2 [CNZ18a], CN-CELEB [FKL+20], Mixer collection (NIST SRE 2004-2010), Switchboard, DeepMine [ZSS18]. This makes around 8540 hours and 16881 speakers in total. All wide-band data in the datasets are down-sampled to 8 kHz.

4.8.4 Comparison of target extraction and separation

In the first part of the experiments, we applied target speech extraction to the three datasets and compared the results to using cascade speech separation and target speaker selection, as discussed in Section 3.2. For speech separation, we report results using two different methods for target speaker selection, i.e. *oracle* and *x-vector*. In the oracle setting, we

[5]Kaldi toolkit https://github.com/kaldi-asr/kaldi
[6]VBx recipe https://github.com/BUTSpeechFIT/VBx

Table 4.4: The structure of the ResNet101 architecture used for x-vector extractor, as described in [LPDB22].

Layer	Structure	Stride	Output
Input	-	-	64×T×1
Conv2D-1	$3 \times 3, 32$	1	64×T×32
ResNetBlock-1	$\begin{bmatrix} 1\times1,32 \\ 3\times3,32 \\ 1\times1,128 \end{bmatrix} \times 3$	1	64×T×128
ResNetBlock-2	$\begin{bmatrix} 1\times1,64 \\ 3\times3,64 \\ 1\times1,256 \end{bmatrix} \times 4$	2	32×T/2×256
ResNetBlock-3	$\begin{bmatrix} 1\times1,128 \\ 3\times3,128 \\ 1\times1,512 \end{bmatrix} \times 23$	2	16×T/4×512
ResNetBlock-4	$\begin{bmatrix} 1\times1,256 \\ 3\times3,256 \\ 1\times1,1024 \end{bmatrix} \times 3$	2	8×T/8×1024
Statistics Pooling	-	-	16×1024
Flatten	-	-	16384
Linear	-	-	256

Table 4.5: Comparison of target speech extraction and speech separation on three datasets. The results show improvements in the metrics over the value of the metric for the original mixture, which is shown in the first row.

	WSJ0-2mix		WHAM		WHAMr	
	SI-SDR	STOI	SI-SDR	STOI	SI-SDR	STOI
Mixture	0.00	73.81	-4.49	62.78	-6.13	59.95
Improvements over mixture						
Separation (oracle)	16.56	22.37	13.69	24.65	10.63	21.47
Separation (x-vector)	16.54	22.38	13.48	24.34	9.93	20.33
Extraction	**17.11**	**22.50**	**14.05**	**25.49**	**11.43**	**23.23**

compare the reference target signal with all outputs of the network and choose the output with the lowest SI-SDR. This experiment thus shows the performance of speech separation with perfect speaker selection. In the x-vector setting, we extract x-vectors from both outputs of the neural network and the enrollment utterance. We then choose the output whose x-vector has the lowest cosine distance to the x-vector of the enrollment utterance. The x-vector extractor was trained on a variety of datasets with high speaker and environment variability and should be fairly robust. Note that this cascade setup has much more parameters than the direct target speech extraction due to the additional x-vector extractor.

The results are shown in Table 4.5. The performance is reported in terms of improvements over mixture in SI-SDR and STOI metrics. We can see that the target speech extraction achieves better performance than separation with oracle speaker selection. This is likely caused by the utilization of additional speaker information in TSE. For the separation experiments, speaker selection using x-vectors introduces additional errors compared to the oracle selection. The errors are more common with the more challenging data, especially for the WHAMr dataset, where TSE leads to significantly better performance.

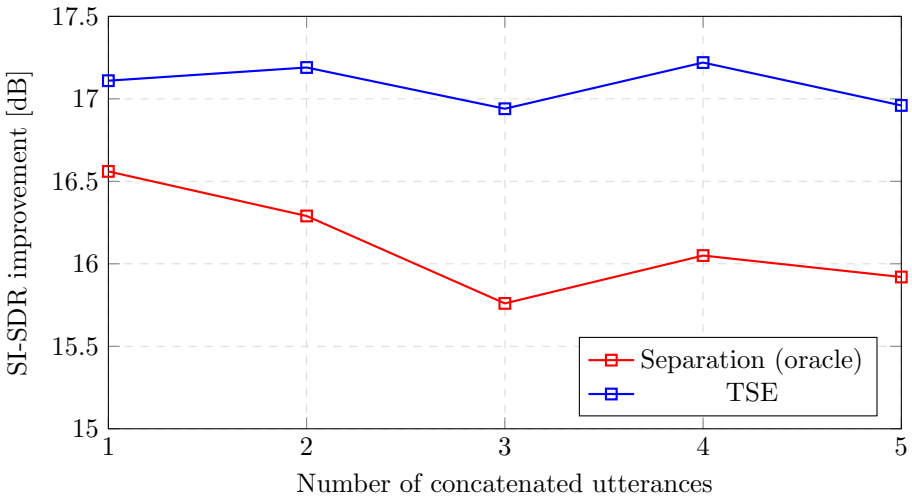


Figure 4.6: Performance of target speech extraction and speech separation for different lengths of the input recordings.

From the results, we can also see a significant drop in performance in presence of noise and reverberation for both TSE and separation methods.

4.8.5 Performance for long recordings

In Section 3.3, we pointed out that speech separation methods often make mistakes by switching the speakers on the output in the middle of the recording. This behavior is pronounced when the recordings are longer. Target speech extraction can keep the speaker at the output consistent due to the additional speaker information. We show this behavior by testing both methods on WSJ0-2mix-long containing multiple concatenated utterances from both speakers, as described in 4.8.1. The results in Figure 4.6 show the SI-SDR improvements as a function of the number of concatenated utterances for each speaker in the mixtures (*1* corresponds to results reported in Table 4.5). The results show that the performance of TSE stays approximately constant when making the mixtures longer, in contrast to the separation, where the performance decreases. Figure 4.7 shows one example of a case where the separation fails to keep the speaker on the output consistent over the recording.

4.8.6 Performance for higher number of speakers

Finally, we also compare the performance of target speech extraction and separation on mixtures with a higher number of speakers than 2. For these experiments, we use WSJ0-*J*mix datasets with $J \in \{1, 2, 3, 4\}$. We explore cases when the number of mixed speakers in the training and test time is matched, mismatched, and also a practical setup in which we present mixtures with different numbers of speakers during the training. For target speech extraction, the architecture of the neural network is not dependent on the number of speakers, we can thus easily train and evaluate the network on different numbers of speakers. For speech separation, the number of nodes in the output layer limits how many speakers the network can separate. For training with a variable number of speakers, we thus use 4 outputs and set the reference signals to zero in case there are less than 4 speakers

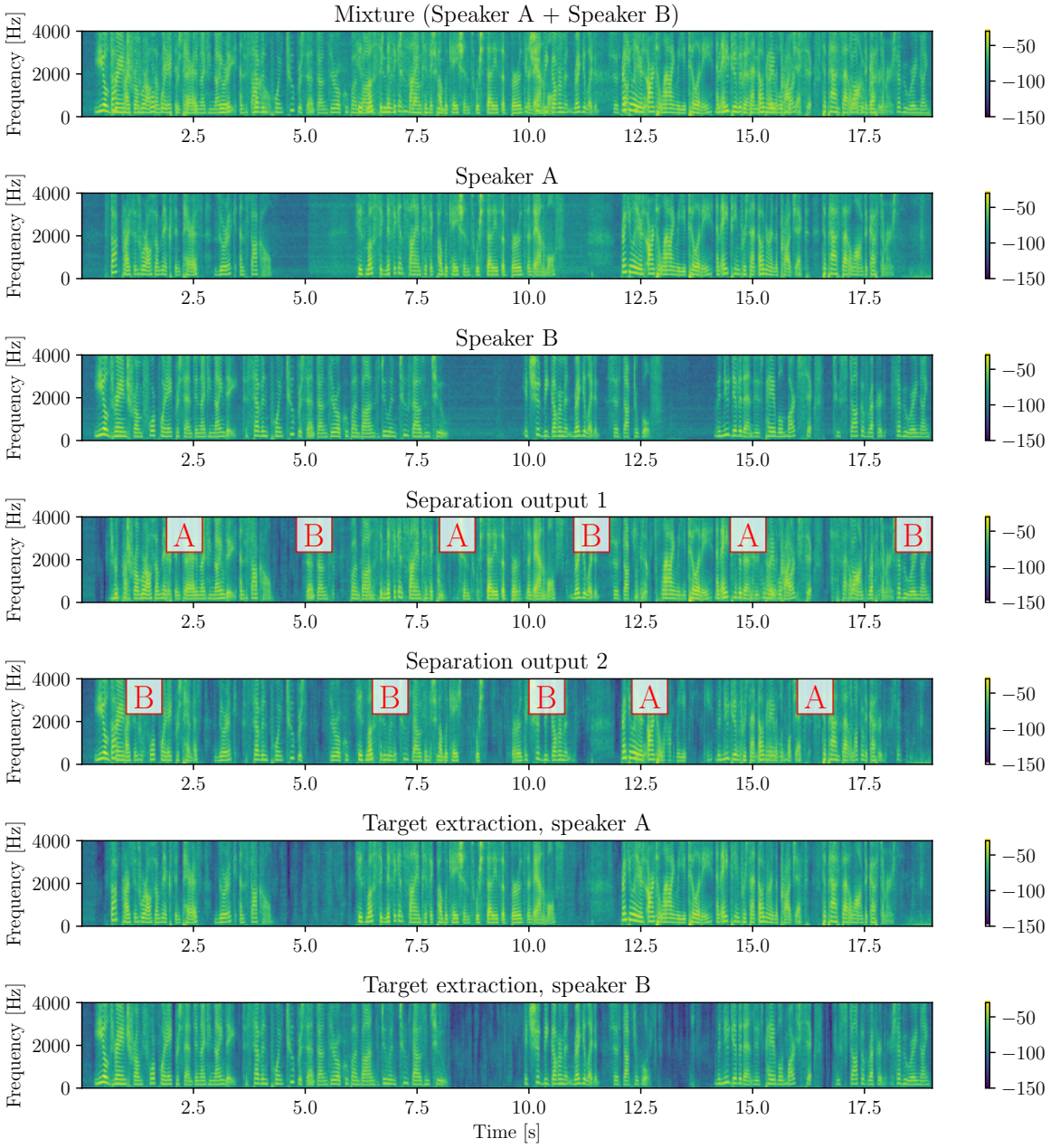


Figure 4.7: Example of outputs of target speech extraction and speech separation on a longer recording. The mixture is created as concatenation of 3 utterances of each speaker. The speech separation often switches the speakers between the outputs in the middle of the recording, as shown by labeling parts of the output as speaker A or speaker B.

Table 4.6: Performance of target speech extraction and speech separation for different number of speakers in mixtures during training and test time. The cases with matched number of speakers in train and test are emphasized in blue for better orientation in the table.

# of speakers in train mixtures	# of speakers in test mixtures: 1 SI-SDR [dB] Sep	1 SI-SDR [dB] Extr	2 ΔSI-SDR [dB] Sep	2 ΔSI-SDR [dB] Extr	3 ΔSI-SDR [dB] Sep	3 ΔSI-SDR [dB] Extr	4 ΔSI-SDR [dB] Sep	4 ΔSI-SDR [dB] Extr
2	23.43	28.98	16.56	**17.11**	4.29	4.89	2.53	2.96
3	11.50	2.20	10.89	13.52	12.58	13.82	6.76	9.59
4	8.26	9.40	7.28	11.20	9.92	12.43	9.40	10.22
1+2+3+4	35.84	**37.48**	14.25	15.98	11.96	**13.98**	4.36	**11.02**

in the mixture. With zero reference signals and SI-SDR loss, the network would tend to scale all output signals close to zero. For this reason, we also change the objective function for this separation experiment to SNR. For both extraction and separation, we clip the loss at 30 dB in the experiment with a variable number of speakers in training, to ensure that the network does not focus too much on improving the performance of single-speaker cases. We use the oracle selection for evaluation of the separation experiment, i.e. from all outputs, we choose the signal closest to the reference target speech in terms of SI-SDR.

The results are shown in Table 4.6. For testing with a single speaker in the recording, we report absolute SI-SDR, in contrast with other experiments where we report improvement over the mixture SI-SDR. For the single-speaker case, the network does not need to do any actual separation or extraction, we just hope that it will not corrupt the signal (SI-SDR values larger than circa 20 dB). We can see that this is true for models trained on two-speaker mixtures or the combined (1+2+3+4) model. Models trained on a larger number of speakers in the mixture tend to separate the input signal even when only a single speaker is present. When testing on a higher number of speakers, unsurprisingly, models trained on a matched number of speakers perform well. We can also see that the models trained on a variable number of speakers can perform well on all testing cases.

As for the comparison between extraction and separation, in matched experiments, the advantage of extraction is similar to what we have seen in two-speaker experiments. The difference in performance gets larger especially in cases when the number of training speakers is larger than in the test (results under the diagonal). In these cases, the separation models tend to over-separate the input mixture more than the extraction models. The models trained on a variable number of speakers tend to perform well on different numbers of speakers in test mixtures. One exception is the speech separation model performance on 4-speaker mixtures which is significantly lower than with the matched model. Interestingly, for TSE, the 1+2+3+4 often performs even better than the matched models. This highlights the independence of TSE on the number of speakers and shows that it is a more general model.

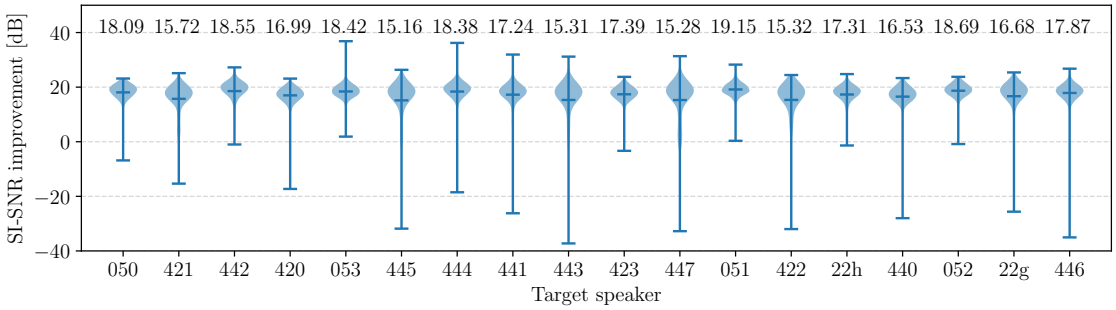


Figure 4.8: Performance of target speech extraction on WSJ0-2mix, shown separately for each target speaker in the test-set. The plot shows minimum, mean and maximum SI-SDR, together with the histogram of the results for each speaker.

4.8.7 Performance for different speakers

Target speech extraction is arguably more difficult when the target speaker has similar voice characteristics as the interfering one. In the previous section, we reported the performance aggregated over the entire test set. In this section, we look more closely at how the performance varies for different speakers.

It has been shown that gender affects the characteristics of the voice [HC99]. It is thus expected that mixtures, where target and interfering speakers are of the same gender, will be more difficult to process than when the gender is different. Table 4.7 shows the SI-SDR results for different combinations of the gender of the target and interfering speakers, for both target speech extraction and separation experiments. For all three datasets, the same gender mixtures (F-F, M-M) are clearly more difficult than different gender (F-M, M-F) ones. Interestingly, for speech separation, the F-F combination seems to be the most problematic, whereas for TSE it is in line with the M-M combination. Perhaps the additional speaker information used in the TSE method helps the network to handle the F-F case better.

Next, we explore the variance in the results for different speakers and speaker pairs. Figure 4.8 shows the SI-SDR improvement on the WSJ0-2mix test set separately for each target speaker in the set. The average results shown at the top of the plot reveal some, but not very substantial differences in performance. The plot also shows the minimum and maximum SI-SDR improvements for each speaker. We can see some extreme values for some speakers (around -40 dB meaning severe deterioration). These are caused by incorrect identification of the target speaker in some cases.

Figure 4.9 further shows the SI-SDR improvements for different speaker pairs. The speakers in this plot are ordered by gender (*050* to *441* female, *443* to *446* male). We can again see the performance decline on same-gender mixtures. However, the plot also shows that not all same-gender speaker pairs lead to bad performance. Overall, there are several speaker pairs with significantly worse performance in TSE (dark blue color). These speaker pairs mostly have bad performance also in SS. This suggests that the issue with similar speakers is rather the separation itself than the identification of the target speaker.

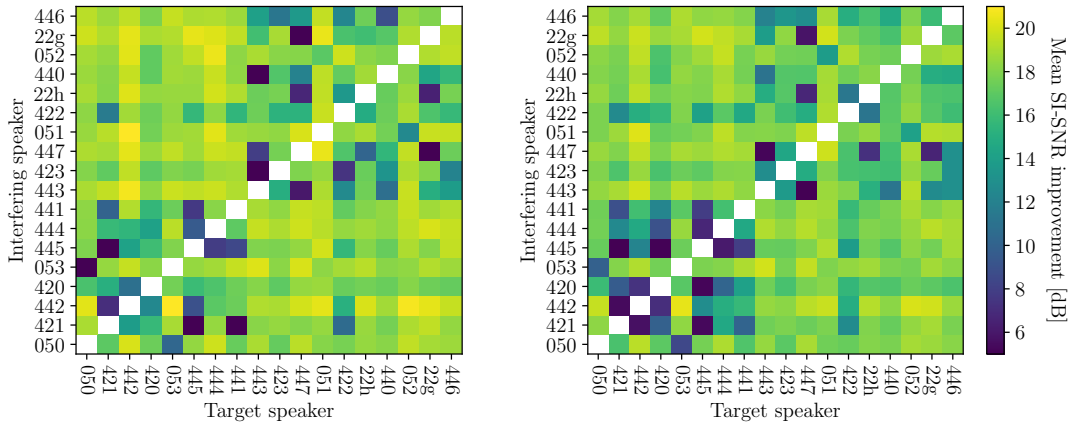


Figure 4.9: Performance of target speech extraction (left) and speech separation (right) on WSJ0-2mix for different pairs of target-interfering speakers.

Table 4.7: Performance of target speech extraction and speech separation for different gender pairs. *F* denotes female and *M* denotes male. The performance is also denoted by the color scale from red to green (separately for each row) for better orientation in the table.

	Extraction Target-Inteference				Separation Target-Inteference			
	F-F	M-M	F-M	M-F	F-F	M-M	F-M	M-F
WSJ0-2mix	15.15	15.52	18.76	18.49	13.73	15.67	18.07	17.89
WHAM	12.65	12.51	15.52	15.19	11.90	12.86	14.84	14.63
WHAMr	9.66	10.13	12.47	12.98	8.48	10.17	11.22	11.95

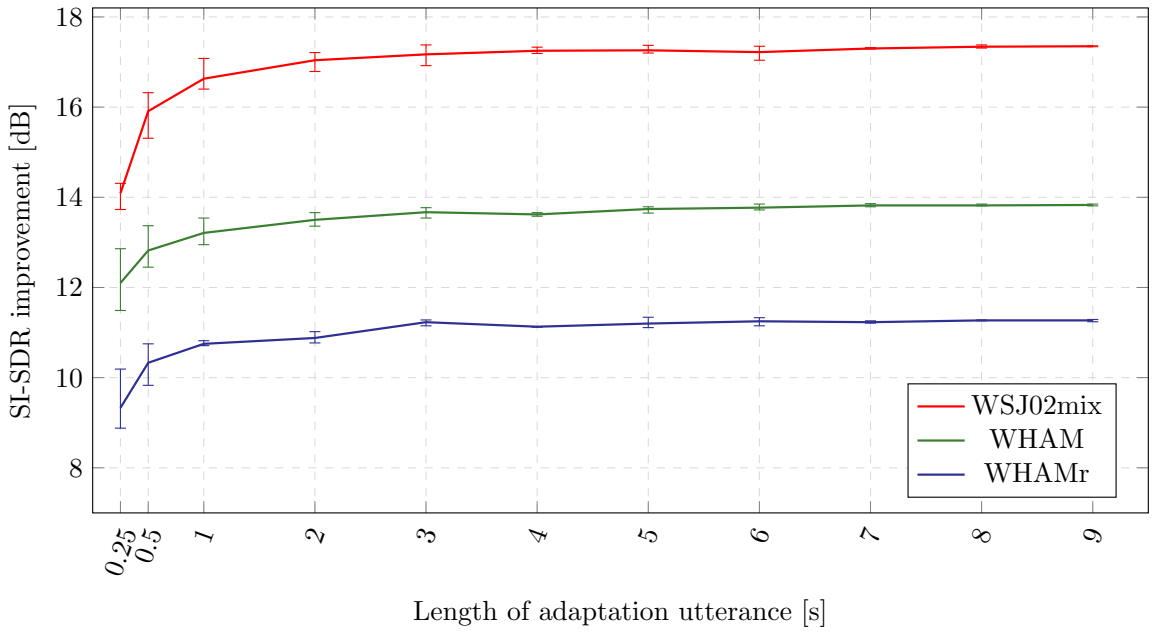


Figure 4.10: Performance of target speech extraction using different lengths of enrollment utterance during test time.

4.8.8 Length of the enrollment utterance

An important aspect to analyze for target speech extraction is the sensitivity to the length of the enrollment utterance. Depending on the application, the techniques may need to work with different lengths of enrollment utterances (e.g. one word or one sentence). Note that we do not focus on cases where the data of the target speaker are very long (e.g. one hour) — in such cases, it might be beneficial to directly re-train the network on the target speaker data.

The neural network in our experiments is trained with 0.5 s segments of enrollment utterances. During test time, we use the full lengths of the enrollment utterances, which, in our datasets, range from 1.6 seconds to 13.9 seconds, with a mean of 5.8 seconds. To analyze the sensitivity to the length, we limit the length of the enrollment utterances from 0.5 to 9 seconds. In all cases, we discard the first 0.5 seconds of the enrollment utterance to skip potential initial silence. From the rest of the utterances, we choose the segment randomly and re-run the evaluation three times to account for the random effect. We also limit the test set to only those utterances for which the original enrollment utterance is longer than 9.5 seconds so that we can evaluate the same set of utterances with all different lengths. This results in 376 utterances in the evaluation.

Figure 4.10 shows the relationship between SI-SDR improvement and the length of the enrollment utterance. We can see that after circa 2 seconds, the performance converges. For very short enrollment utterances, the performance degrades, but even for as long as 0.25 seconds, the difference to using the full length is only about 2-3 dB. Such difference is noticeable, but far from destructive. We thus conclude that the method is not overly sensitive to the length of the enrollment.

Table 4.8: Performance of target speech extraction with different methods to inform the neural network using target speaker embedding. All reported results are improvements in SI-SDR.

Informing method	WSJ0-2mix	WHAM	WHAMr
Multiplication	**17.11**	**14.05**	**11.43**
Multiplication-addition	17.05	14.03	11.35
Concatenation	16.07	13.67	10.97
Factorized layer	16.71	13.99	10.46
Attention-multiplication	15.39	12.83	10.44

4.8.9 Informing the network

In Section 4.3, we have presented several methods for informing the neural network about the target speaker using a speaker embedding. In previous experiments, we have used multiplication (see Section 4.8.3). In this section, we compare this with different methods, namely multiplication-addition, concatenation, factorized layer, and attention. The results of all methods are shown in Table 4.8. We can see that with all methods, the neural network is able to learn to extract the target speaker with fairly satisfactory performance. The best results over all three datasets are achieved by the multiplication with multiplication-addition, factorized layer and concatenation following closely. An interesting observation is that the attention-based method produces worse results than multiplication, although it has more freedom. We suspect that this is due to overfitting as the final training loss value for the attention method was comparable to multiplication and better than other methods, such as concatenation or factorized layer.

4.8.10 Speaker embedding

In Section 4.2, we reviewed different possibilities of how to represent speaker information in the enrollment utterance in a speaker embedding. There are three common choices, i.e. i-vectors, x-vectors and embeddings jointly learned with the task, that were used in the previous sections. Here, we experimentally compare these options.

Table 4.9 shows the SI-SDR improvements using different speaker embeddings. We report the results on both the test and validation sets. Note that the validation sets of all WSJ0-2mix, WHAM, and WHAMr contain the speakers seen in the training. It may thus be used to assess the performance on a closed-set of speakers, as opposed to the test set, that contains unseen speakers only. On the test set, the jointly learned embeddings lead to the best performance, followed by i-vectors. The advantage of jointly learned embeddings is especially large in the case of clean data (WSJ0-2mix).

The trends are however different on the validation set with seen speakers. There, the results of all methods are more even. This shows that in the case of using i-vectors or x-vectors as the speaker embeddings, the neural network overfits to the training speakers. We can hypothesize that this trend could be different in case of having larger speaker variability in the training. The used datasets contain 101 speakers in the training set, which might not be enough when the external embeddings are used.

From the presented results, the jointly learned embeddings seem to be well-suited for informing the neural network about the target speaker, and thus they should contain speaker information, even though the embeddings were not forced by an explicit speaker loss to

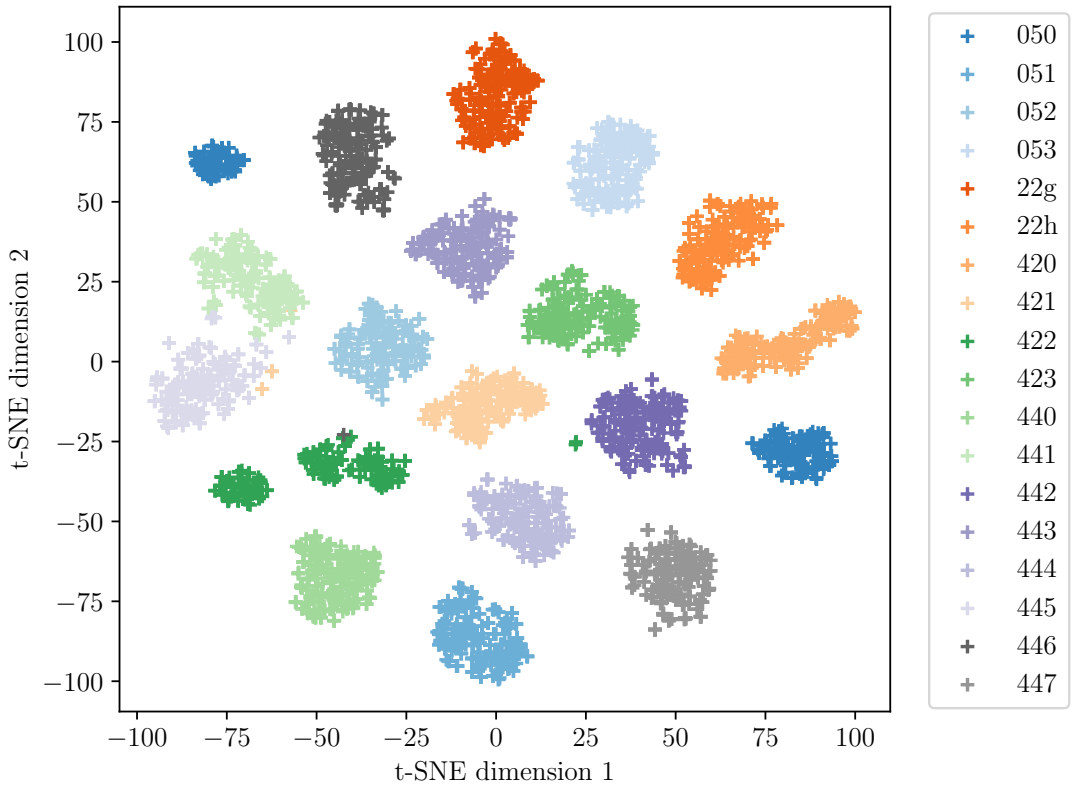


Figure 4.11: Speaker embeddings extracted from test-set of WSJ0-2mix using auxiliary network jointly learned with the task of target speech extraction. The embeddings are reduced to two dimensions with t-SNE. Each point represents one utterance and colors denote speakers.

Table 4.9: Performance of target speech extraction using different speaker embeddings, on both test and validation sets. All reported results are improvements in SI-SDR.

	Test (unseen speakers)			Validation (seen speakers)		
	WSJ0-2mix	WHAM	WHAMr	WSJ0-2mix	WHAM	WHAMr
I-vector	13.76	12.51	10.60	**18.29**	**14.63**	**12.00**
X-vector	12.59	10.50	8.44	17.99	14.28	11.90
Jointly learned	**17.11**	**14.05**	**11.43**	18.20	14.49	11.95

Table 4.10: Performance of target speech extraction with different input and output domains. For all domains, two sets of filter sizes are used denoted here by *window-length/window-shift/number-of-filters*. All reported results are improvements in SI-SDR.

	WSJ0-2mix		WHAM		WHAMr	
	16/8/512	256/64/256	16/8/512	256/64/256	16/8/512	256/64/256
Learnable	**17.11**	11.50	**14.05**	11.11	**11.43**	8.98
STFT	12.89	10.99	11.70	10.01	9.66	7.28
STFT-magnitude	11.64	11.08	10.00	9.60	7.77	6.92

model the speaker information. To confirm that the embeddings are a good representation of the speakers, we plot the embeddings extracted from WSJ0-2mix test single-speaker utterances reduced to two dimensions using the t-SNE method[7] [VdMH08]. Figure 4.11 shows the plot of embeddings, where each point corresponds to one utterance and the color labels the speaker identity. The embeddings clearly form clusters and embeddings of utterances spoken by different speakers are in most cases parts of different clusters.

4.8.11 Domain and loss

In the literature, there are two prevalent setups of the input/output domain and loss function of the neural networks. First, in earlier works, STFT or STFT magnitude was often used with bigger windows of about 200-500 samples. The loss function was usually MSE or phase-sensitive MSE (PS-MSE). Second, in recent work, learnable filters are usually used to extract the representation with short windows of about 10-40 samples. This is usually trained with SI-SDR loss. The latter is also the default setup used in this work (see Sec-

[7] t-SNE https://lvdmaaten.github.io/tsne/

Table 4.11: Performance of target speech extraction for different loss functions. In these experiments STFT representation with window length 256, window shift 64 was used.

	WSJ0-2mix		WHAM		WHAMr	
	ΔSI-SDR [db]	ΔSTOI [%]	ΔSI-SDR [db]	ΔSTOI [%]	ΔSI-SDR [db]	ΔSTOI [%]
SI-SDR	**10.99**	**16.53**	**10.01**	**15.38**	**7.28**	**14.03**
MSE	8.92	15.26	8.27	14.22	5.36	9.28
PS-MSE	9.83	15.06	7.23	12.79	4.78	7.81

tion 4.8.3). In this section, we experiment not only with these two setups, but also combine the individual settings.

First, we focus on the input and output domain and test three representations (learnable, STFT, and STFT-magnitude) with two different sizes of windows (small and large). In the small window setting, we use parameters often used with learnable filters, i.e. 16 sample windows with 8 sample shifts, 512 filters). As in STFT, the number of meaningful outputs corresponds to the number of input samples, we repeated the output values to obtain the output size 512, as in the case of learnable filters. In the case of STFT, we concatenate real and imaginary parts of the spectrum. For STFT-magnitude, we use magnitude only. In the large window setting, we use 256 sample windows with 64 sample shifts and 256 filters. The results of this experiment are shown in Table 4.10. First, the shorter window length and shift lead to substantially better quality than the longer one. The difference is especially large in the case of learnable filters but holds also for STFT and STFT-magnitude. Also, the results exhibit the same trends for all three datasets. This is not completely expected, as especially the reverberation condition could influence the choice of the window length. However, as the results show, this is not the case. The neural network still benefits from having more fine-grained information in time even for reverberant conditions. Among the three domains (learnable, STFT, and STFT-magnitude), learnable filters are clearly beneficial, especially in the case of shorter windows.

In the second set of experiments, we explored different loss functions. The earlier works using the STFT domain mostly used MSE or PS-MSE as a loss function. Here, we compare this with SI-SDR. In these experiments, we use the larger 256 sample windows and complex STFT as the input. Table 4.11 shows the results of these experiments. Here, we report both SI-SDR and STOI, to avoid the advantage of SI-SDR of having matched loss and evaluation metric. The SI-SDR loss function leads to clearly better performance in terms of both the SI-SDR and STOI metrics. Comparing MSE and PS-MSE, the only case where PS-MSE exhibits better results is WSJ0-2mix evaluated with SI-SDR metric. Note that this combination of dataset and metric is often used in the literature. In other cases, the MSE loss function works better.

Chapter 5

Multi-channel approaches to target speech extraction

In the previous chapter, we applied target speech extraction to single-channel signals only. In many real applications, multiple microphones are available to record the signal. This includes many modern smartphones, personal assistant devices, game consoles, TVs, or hearing aids. In this case, when the multi-channel signal is available, it is possible to make use of spatial information to obtain an enhanced signal of a better quality [VGP07]. Research in multi-channel processing has a long history and a large variety of methods are available [BCH08, SBA10, HUHD+21]. In this chapter, we show how neural target speech extraction can be combined with multi-channel processing methods. Note that in this work, we focus on the case when we do not have any location information about the target speaker, i.e. only the input mixture is multi-channel.

5.1 Classical beamforming

Most multi-channel methods rely on linear spatial filtering, i.e. beamforming. In this section, we summarize the most common beamforming methods.

5.1.1 Spatial filter design

Let us revisit the observed signal model in STFT domain defined in Section 3.1:

$$Y^{(m)}(n,f) = A_i^{(m)}(n,f)S_i(n,f) + \sum_{j\neq i} A_j^{(m)}(n,f)S_j(n,f) + V^{(m)}(n,f), \quad \text{(3.2 revisited)}$$

where i is the index of the target speaker, $Y^{(m)}(n,f)$ is the observed mixture at microphone m in STFT domain, $S_j(n,f)$ is the speech signal of the speaker j in STFT domain, $A_j^{(m)}(n,f)$ models the effect of the room impulse response in the frequency domain, $V^{(m)}(n,f)$ is the noise signal including the RIR from the sources of the noise to the microphone m in STFT domain, n is the index of STFT frame, and f is the index of frequency bin. In this chapter, we will use vector notation

$$\mathbf{y}(n,f) = \mathbf{a}_i(n,f)S_i(n,f) + \underbrace{\sum_{j\neq i} \mathbf{a}_j(n,f)S_j(n,f) + \mathbf{v}(n,f)}_{\mathbf{r}_i(n,f)}, \tag{5.1}$$

where $\mathbf{y}(n,f) = [Y^{(1)}(n,f), Y^{(2)}(n,f), \ldots, Y^{(M)}(n,f)]^\top$ is a vector comprising STFT coefficients in time-frame n and frequency bin f for all microphones. Analogously, $\mathbf{v}(n,f) = [V^{(1)}(n,f), V^{(2)}(n,f), \ldots, V^{(M)}(n,f)]^\top$ and $\mathbf{a}_i(n,f) = [A_i^{(1)}(n,f), A_i^{(2)}(n,f), \ldots, A_i^{(M)}(n,f)]^\top$. We additionally introduce the notation $\mathbf{r}_i(n,f)$ for all interfering signals with respect to target speaker i (i.e. other speakers and noise), and $X_i^{(m)}(n,f) = A_i^{(m)}(n,f)S_i(n,f)$ for speech of the target speaker i as received by microphone m.

In beamforming methods, we design a multi-channel filter $\mathbf{w}$, that can extract the target signal out of the mixture

$$\hat{X}_i^{(m')}(n,f) = \mathbf{w}^{\mathsf{H}}(n,f)\mathbf{y}(n,f), \tag{5.2}$$

where m' is a reference microphone and $\cdot^{\mathsf{H}}$ denotes Hermitian transpose. The filter is designed to be in some sense "optimal", i.e. leading to "best possible" reduction of the noise and preservation of the target signal. Different definitions of optimality criterion lead to different types of beamformers.

Multi-channel Wiener filter

First, minimizing the mean square error between the estimate and the reference signal leads to *Multi-channel Wiener filter* (MWF) [BCH08]

$$\mathbf{w}_{\mathrm{MWF}}(n,f) = \underset{\mathbf{w}}{\operatorname{argmin}}\, \mathbb{E}[|\hat{X}_i^{(m')}(n,f) - X_i^{(m)}(n,f)|^2] = \mathbf{\Sigma}_y^{-1}(n,f)\mathbf{\Sigma}_i(n,f)\mathbf{e} \tag{5.3}$$

where $\mathbf{e}$ is a one-hot vector denoting the reference microphone, $\mathbf{\Sigma}_y(n,f) = \mathbb{E}[\mathbf{y}(n,f)\mathbf{y}^{\mathsf{H}}(n,f)]$ is the spatial correlation matrix of the observed signal and

$$\mathbf{\Sigma}_i(n,f) = \mathbb{E}[|S_i(n,f)|^2]\mathbf{a}_i(n,f)\mathbf{a}_i^{\mathsf{H}}(n,f) \tag{5.4}$$

is the spatial correlation matrix of the target signal. The spatial correlation matrices (SCM) are $M \times M$ matrices encoding the correlation of the signals at different microphones in the frequency-domain.

Minimum variance Distortionless Response filter

Multi-channel Wiener filter, in its basic form, weights equally the effect of speech distortion and noise reduction. Often it is desirable not to cause any speech distortion for the prize of smaller noise reduction. Such optimality criterion leads to *Minimum Variance Distortionless Response* (MVDR) beamformer [GC08, SBA10]

$$\mathbf{w}_{\mathrm{MVDR}}(n,f) = \underset{\mathbf{w}}{\operatorname{argmin}}[|\mathbf{w}^{\mathsf{H}}\mathbf{r}_i(n,f)|^2] \quad \text{s.t. } \mathbf{w}^{\mathsf{H}}X_i^{(m)}(n,f) = X_i^{(m)}(n,f) \tag{5.5}$$

$$\mathbf{w}_{\mathrm{MVDR}}(n,f) = A_i^{*(m)}(n,f)\frac{\mathbf{\Sigma}_r^{-1}(n,f)\mathbf{a}_i(n,f)}{\mathbf{a}_i^{\mathsf{H}}(n,f)\mathbf{\Sigma}_r^{-1}(n,f)\mathbf{a}_i(n,f)}, \tag{5.6}$$

where $\mathbf{\Sigma}_r(n,f) = \mathbb{E}[\mathbf{r}(n,f)\mathbf{r}^{\mathsf{H}}(n,f)]$ is the SCM of the interference signal.

Note that it can be shown that both MWF and MVDR are special cases of more general parametric Multi-channel Wiener filter [SBA10], which allows tuning the amount of target signal distortion and noise reduction.

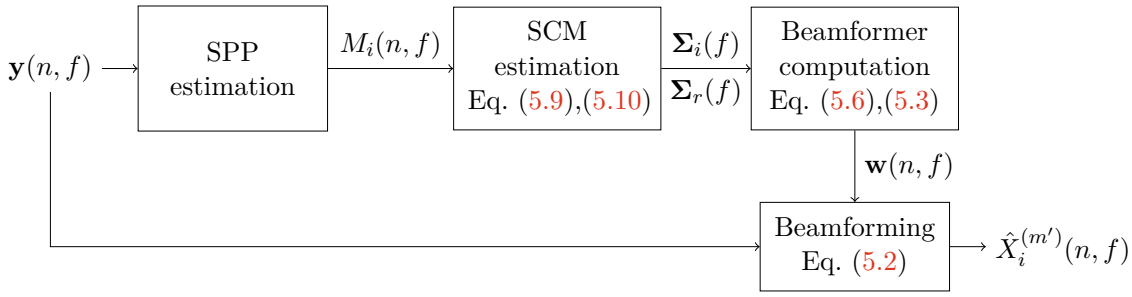


Figure 5.1: Estimation of the statistics of the target and interfering signals based on speech presence probability, estimation of beamforming coefficients and the application of beamforming.

5.1.2 Estimation of beamformer parameters

In the previous section, several types of beamforming filters with different optimality criteria were introduced. The formulations of the filters depend on statistics of the signals, namely acoustic transfer function (ATF) of the target speaker $\mathbf{a}_i(n,f)$ and spatial correlation matrices of observed speech $\mathbf{\Sigma}_y(n,f)$, target speech $\mathbf{\Sigma}_s(n,f)$ and interference $\mathbf{\Sigma}_r(n,f)$. In the following, we revise ways to estimate these statistics. We assume stationarity of the signals over N time frames and thus drop the dependency of the SCMs and ATF on time frame n. The SCM of the observed signal can then be estimated as

$$\mathbf{\Sigma}_y(f) = \frac{1}{N}\sum_{n=1}^{N}\mathbf{y}(n,f)\mathbf{y}^{\mathsf{H}}(n,f) \tag{5.7}$$

assuming ergodicity of the signal $\mathbf{y}$. The definition of $\mathbf{\Sigma}_i(f)$ suggests that $\mathbf{a}_i(f)$ can be obtained as the principal eigenvector of $\mathbf{\Sigma}_i(f)$

$$\mathbf{a}_i(f) = \mathcal{P}(\mathbf{\Sigma}_i(f)). \tag{5.8}$$

The problem thus boils down to estimation of SCMs of target $\mathbf{\Sigma}_i(f)$ and interference $\mathbf{\Sigma}_r(f)$, which cannot be obtained as in Equation 5.7 because of unavailability of isolated target and interference signals.

There are numerous methods for the estimation of these spatial correlation matrices. Here, we focus on methods based on speech presence probability (SPP), which are widely used and combine well with neural networks. Denoting the probability of the target speaker i being present in time-frequency bin (n,f) as $M_i(n,f)$ and probability of the interference being present in (n,f) as $M_r(n,f) = 1 - M_i(n,f)$, we can estimate the SCMs as:

$$\mathbf{\Sigma}_i(f) = \frac{1}{\sum_n M_i(n,f)}\sum_{n=1}^{N} M_i(n,f)\mathbf{y}(n,f)\mathbf{y}^{\mathsf{H}}(n,f) \tag{5.9}$$

$$\mathbf{\Sigma}_r(f) = \frac{1}{\sum_n M_r(n,f)}\sum_{n=1}^{N} M_r(n,f)\mathbf{y}(n,f)\mathbf{y}^{\mathsf{H}}(n,f), \tag{5.10}$$

i.e. the estimates are being updated during time-frames which are likely dominated by the target speaker or interference, respectively. We thus need an estimator of the target speech

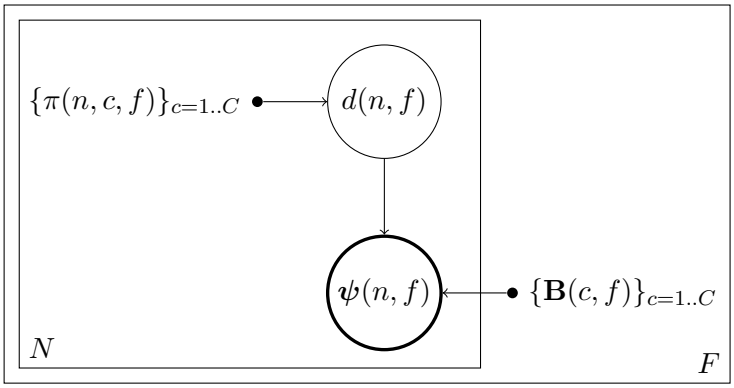


Figure 5.2: Graphical representation of the spatial CACGMM model.

presence probability to obtain the necessary statistics and subsequently the beamformer coefficients. The overall estimation of the beamforming filters is schematically depicted in Figure 5.1.

5.1.3 Spatial models for speech presence probability estimation

One popular approach for estimation of speech and interference presence probabilities are statistical signal models [IAN, MWE09, VHU10]. These were shown to be very successful, e.g. in CHiME challenges [YID+15, BHS+18] on distant automatic speech recognition, where they were often parts of the winning systems. In these models, multi-channel observations in the form of *spatial features* are modeled by a mixture model. Each component of the mixture model typically corresponds to one source. A graphical representation of the full mixture model is depicted in Figure 5.2 and corresponds to the following model of the probability density

$$p(\boldsymbol{\psi}(n,f)) = \sum_c \pi(c,n,f) p(\boldsymbol{\psi}(n,f)|d(n,f)=c), \tag{5.11}$$

where c is the index of the mixture component, $\boldsymbol{\psi}(n,f)$ is the spatial feature for time-frequency bin (n,f) and $d(n,f)$ is the latent variable modeling affiliation of the time frequency bin (n,f) to the components of the mixture. The symbol $\pi(c,n,f)$ denotes the prior probability $p(d(n,f)=c)$ of time-frequency point (n,f) belonging to mixture component c (mixture weight). In different works, the mixture weight is shared either across time [IAN] or across frequency [IAN13]. Here, we will use the variant of sharing across frequency $\pi(c,n) = \pi(c,n,f)$. In Section 5.2.3, we will introduce another variant, where $\pi(c,n,f)$ is pre-estimated and fixed. The posterior probability of the affiliation to the components given the observed spatial features $p(\mathbf{D}|\boldsymbol{\Psi})$ then corresponds to the sought presence probabilities.

To construct a spatial model, an important choice to make is the form of the spatial features and corresponding probability density to model individual components of the mixture model. Here, we will focus on *normalized observation vectors* as the spatial features, modeled by *complex angular central Gaussian* (CACG) distribution [IAN]. The normalized observation vectors are defined as

$$\boldsymbol{\psi}(n,f) = \frac{\mathbf{y}(n,f)}{|\mathbf{y}(n,f)|}. \tag{5.12}$$

The complex angular central Gaussian is defined as

$$p(\boldsymbol{\psi}(n,f)|d(n,f)=c) = \frac{(K-1)!}{2\pi^K \det \mathbf{B}(c,f)} \frac{1}{(\boldsymbol{\psi}(n,f)^{\mathsf{H}}\mathbf{B}(c,f)^{-1}\boldsymbol{\psi}(n,f))^K}, \tag{5.13}$$

where K is the number of channels (dimensionality of the spatial features) and $\mathbf{B}$ is the parameter of the distribution. The full model is then referred to as *complex angular central Gaussian mixture model* (CACGMM).

The parameters of the model $\boldsymbol{\pi}$ and $\mathbf{B}$ are estimated using Expectation Maximization algorithm. The algorithm iterates between E-step estimating the approximate posterior $q(\mathbf{D})$ with the current values of parameters

$$q(d(n,f)=c) = \frac{\pi^{(old)}(c,n,f)p(\boldsymbol{\psi}(n,f)|d(n,f)=c;\mathbf{B}^{(old)})}{\sum_{c'}\pi^{(old)}(c',n,f)p(\boldsymbol{\psi}(n,f)|d(n,f)=c';\mathbf{B}^{(old)})} \tag{5.14}$$

and M-step estimating new parameter values with the current approximate posterior

$$\mathbf{B}^{(new)}, \boldsymbol{\pi}^{(new)} = \operatorname*{argmax}_{\mathbf{B},\boldsymbol{\pi}} \mathbb{E}_{q(\mathbf{D})}[\ln p(\boldsymbol{\Psi}, \mathbf{D}; \mathbf{B}, \boldsymbol{\pi})], \tag{5.15}$$

$$\mathbf{B}(c,f) = \frac{K}{\sum_n q(d(n,f)=c)} \sum_n q(d(n,f)=c) \frac{\boldsymbol{\psi}(n,f)\boldsymbol{\psi}^{\mathsf{H}}(n,f)}{(\boldsymbol{\psi}(n,f)^{\mathsf{H}}\mathbf{B}(c,f)^{-1}\boldsymbol{\psi}(n,f))^K}, \tag{5.16}$$

$$\pi(c,n,f) = \pi(c,n) = \frac{1}{F}\sum_f q(d(n,f)=C). \tag{5.17}$$

Apart from the prior sharing over frequency bins, the model treats each frequency bin independently. As a result, the components in different frequency bins are often permuted, i.e. the first component in one frequency bin does not correspond to the first component in another frequency bin. Permutation alignment algorithms are usually used to solve this problem [SMAM04]. These algorithms compute the correct permutation based on the similarity of the posterior probabilities (masks) at different frequencies. In the case of shared mixture weight across frequency, the permutation alignment can be also used "online", i.e. in each iteration of the E-M algorithm.

5.2 Neural network-based beamforming

Following the dominance of neural networks for single-channel speech enhancement, separation, and extraction, NN-based methods emerged also for multi-channel processing. In many works, the strong modeling power of neural networks is combined with classical beamforming methods as introduced in the previous section. In this section, we overview such combinations and focus on how they can be applied for target speech extraction.

5.2.1 Neural network for speech presence probability estimation

Using neural networks as speech presence estimator for beamforming was proposed in [HDHU16, EHW+16] for speech enhancement task. In these works, the neural network directly estimates probability of speech being present $M^{(m)}(n,f)$ from the single-channel observation $y^{(m)}(n,f)$. The estimates are then aggregated over all K channels, e.g. using

mean operation

$$M^{(m)}(n,f) = f_{\text{NN}}(y^{(m)}) \tag{5.18}$$

$$M(n,f) = \frac{1}{K}\sum_{m=1}^{K} M^{(m)}(n,f). \tag{5.19}$$

The obtained estimate $M(n,f)$ then can be used to compute SCMs, as in Equations (5.9),(5.10) and construct a beamformer (Equations (5.6),(5.3)). The neural network is trained with one of the mask-based loss functions, introduced in Section 4.5. Note that the neural network in this case does not use any spatial information because it processes only a single channel at a time. This choice was made to prevent the neural network from overfitting on the microphone array configurations in the training data.

To apply this scheme to target speech extraction, we simply need to inform the neural network by the embedding extracted from the enrollment utterance

$$M_i^{(m)}(n,f) = f_{\text{NN}}(y^{(m)}, e_i) \tag{5.20}$$

$$M_i(n,f) = \frac{1}{K}\sum_{m=1}^{K} M_i^{(m)}(n,f). \tag{5.21}$$

5.2.2 Speech presence probability estimation from time-domain

The neural networks estimating speech presence as proposed in [HDHU16, EHW+16] work in frequency domain. Recent works (and our investigation in previous chapter) show that time-domain neural networks with SI-SDR loss lead to superior performance. To make use of these models, we can estimate the speech presence probability based on the estimated time-domain signal. To do that, we can re-use the definitions of ideal masks as introduced in Section 4.5. The estimated signal is then used in place of the reference. The estimated ideal mask can then be used as the presence probability

$$\hat{s}_i^{(m)} = f_{\text{NN}}(y^{(m)}, e_i) \tag{5.22}$$

$$M_i^{(m)}(n,f) = \text{ideal-mask}(\hat{s}_i^{(m)}, y^{(m)}) \tag{5.23}$$

$$M_i(n,f) = \frac{1}{K}\sum_{m=1}^{K} M_i^{(m)}(n,f), \tag{5.24}$$

where function ideal-mask computes IBM, IAM or IPSM (Equations (4.9),(4.10),(4.11), respectively), using STFT representation of both signals.

5.2.3 Integration of neural networks and spatial models

Both speech presence probability estimation based on spatial models (Section 5.1.3) and neural networks (Section 5.2.1) have their benefits and drawbacks. Neural networks have great modeling power and can make use of correlations of speech signals across time and frequency. On the other hand, spatial models have been shown to well model spatial patterns. Due to their unsupervised nature, they also do not suffer from a mismatch between training and test conditions in contrast with neural networks.

In [NIH+17], it has been proposed to combine the two approaches for speech enhancement. In this work, the neural network predicts presence probability $M_i(n,f)$, $M_r(n,f)$ for

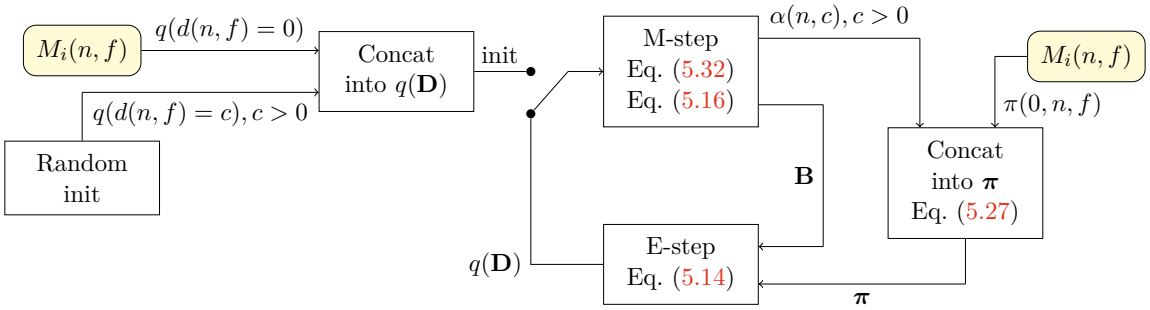


Figure 5.3: The inference of integrated target speech extraction with spatial clustering. For initialization of $q(\mathbf{D})$, left part of the scheme is used. After that, the inference iterates as shown on the right. Yellow blocks show where the masks estimated using TSE enter the inference.

both speech and noise components of the signal. These predictions are then used as the prior $\pi(c,n,f)$ on the affiliation of T-F bins to the respective components. In addition, the approximate posterior $q(\mathbf{D})$ is also initialized to the predicted presence probabilities in the first iteration of the inference

$$\pi(c,n,f) = M_c(n,f) \tag{5.25}$$

$$q^{(init)}(d(n,f)=c) = M_c(n,f). \tag{5.26}$$

In contrast with the conventional spatial clustering, the prior probabilities $\boldsymbol{\pi}$ are not re-estimated and stay fixed (Equation 5.17 is not used). The priors affect the E-step of the inference algorithm (Equation 5.14), where the contribution of the spatial model in each T-F bin is weighted according to $\boldsymbol{\pi}$. It can thus significantly influence the final result.

The method can be also readily used with time-domain neural networks by re-applying the concept introduced in the previous section to estimate the presence probabilities.

5.2.4 Integration of target speech extraction and spatial models

The above-described method for integration of the neural network with spatial clustering was proposed for speech enhancement. Similar schemes have also been employed for speech separation [DHU17]. In these, we however need an estimation of the presence probabilities for each component in the signal. This does not agree with the concept of target speech extraction where the neural network predicts an estimate of the target speaker's speech only. Here, we modify the method to deal with this case:

We aim to have the prior probability of the first component fixed to the presence probability estimated with the TSE neural network. At the same time, the priors for the other components should be re-estimated in each iteration and tied across frequency. For this, we re-parametrize the prior probability as

$$\pi(c,n,f) = \begin{cases} M_i(n,f) & \text{for } c=0, \\ \alpha(c,n) & \text{for } c>0, \end{cases} \tag{5.27}$$

where $M_i(n,f)$ is the presence probability of the target speaker as predicted by the neural network (either directly or through time-domain signal as specified in Section 5.2.2) and $\alpha(c,n)$ is a new parameter specifying the prior for the other components.

To find the update rule for $\alpha(c,n)$, we need to solve the following optimization problem

$$\boldsymbol{\alpha} = \underset{\boldsymbol{\alpha}}{\operatorname{argmax}}\, \mathbb{E}_{q(\mathbf{D})}[\ln p(\boldsymbol{\Psi}, \mathbf{D}; \mathbf{B}, \boldsymbol{\pi})] \quad \text{s.t.} \quad \sum_{c>0} \alpha(c,n) + M(n,f) = 1 \;\; \forall n, f. \tag{5.28}$$

This leads to solving the following equation using Lagrange multipliers $\lambda(n)$

$$\frac{\partial \sum_f q(d(n,f)=c) \ln \alpha(n,c) - \lambda(n)(1-\alpha(c,n))}{\partial \alpha(n,c)} = 0 \tag{5.29}$$

$$\alpha(n,c) = \frac{q(d(n,f)=c)}{\lambda(n)}. \tag{5.30}$$

Using the original constraint, we can derive the value of $\lambda(n)$

$$\sum_{c>0} \frac{q(d(n,f)=c)}{\lambda(n)} + M(n,f) = 1 \;\; \Rightarrow \;\; \lambda(n) = \frac{1 - M(n,f)}{\sum_c q(d(n,f)=c)} \tag{5.31}$$

and the final update rule

$$\alpha(n,c) = \frac{q(d(n,f)=c)}{\sum_{c>0} q(d(n,f)=c)}(1 - M(n,f)). \tag{5.32}$$

The overall iterative inference used in combination of TSE and spatial clustering is depicted in Figure 5.3.

5.3 Experiments

5.3.1 Dataset and configuration

For multi-channel experiments, we use SMS-WSJ (Spatialized Multi-Speaker Wall Street Journal) database [DHBHU19] consisting of data artificially created based on WSJ0+1. In particular, the utterances for training, validation and test were taken from *si284*, *dev93* and *eval92* parts of WSJ, respectively. The data is downsampled to 8 kHz and multi-channel mixtures are simulated using the image method [AB79]. The simulated microphone array is a circular array with a radius of 10 cm and 6 microphones. The room impulse responses are generated with T_{60} of 200 ms to 500 ms. The distance of each source from the microphone array center is 1 m to 2 m. The training, validation and test sets contain 33561, 982, and 1332 mixtures, and 283, 10, and 8 speakers, respectively. The mixtures are not fully overlapped. Instead, the longer utterance determines the length of the mixture, and the shorter utterance is padded with random offset in the beginning.

The model has an architecture described in Section 4.8.3, used also in experiments in the previous chapter. For training, we used only the fully overlapped part of the utterance, as determined by the boundaries of the original utterances[1]. As a loss function, we chose SNR, as it lead to more stable training than SI-SDR on this database. We trained the model for 135 epochs. As targets for the training, we used the reference signals including early reflections. For evaluation, we used the reference sources without any reverberation as the references and SDR metric, as recommended by the authors of SMS-WSJ database [DHBHU19].

[1] Silence parts which are inside the original utterances are kept, the mixtures thus might contain short segments of single speaker.

Table 5.1: Comparison of different STFT window sizes and shifts, using oracle MVDR beamforming with ideal binary mask. All reported results are SDR improvements over mixture.

size shift	512	1024	2048
32	12.67	**14.82**	14.71
64	12.67	**14.81**	14.71
128	12.64	**14.80**	14.70
256	12.33	14.74	14.68

Table 5.2: Comparison of different ideal masks, using oracle MVDR beamforming with STFT window size 1024 and shift 128. All reported results are SDR improvements over mixture.

	MVDR	MWF
IBM	**14.80**	12.45
IAM	12.34	11.75
IPSM	13.04	12.57

For experiments with spatial clustering, we use 100 iterations and 3 components (unless specified otherwise). We use inline permutation alignment implemented by `pb_bss`[2] toolkit [DHU17].

5.3.2 Results

Before experiments with TSE, we used oracle beamforming to compare different choices for the beamformer and window size. As we use the computation of mask from the estimated time-domain signal, the window size can be determined flexibly without dependence on the window size used in the neural network. Table 5.1 shows comparison of different STFT window sizes and shifts. These experiments are done with an MVDR beamformer using the ideal binary mask, computed using the reference signal, to estimate the beamformer weights. The results show that a rather large window with a small shift is the optimal configuration. In subsequent experiments we use the configuration 1024-128, as it has one of the best results and leads to a shorter sequence (and thus shorter processing times) than for example shift 32 or 64.

Table 5.2 shows the comparison of two different beamformers (MVDR and MWF) and three different types of masks used to estimate the beamformer weights (ideal binary mask, ideal amplitude mask, and ideal phase-sensitive mask). The MVDR beamformer consistently outperforms MWF and is best estimated by using the IBM mask. In the subsequent experiments, we will thus use the IBM mask and MVDR beamformer (unless specified otherwise).

Next, we explore the combination of beamforming and TSE (Table 5.3). We compare the SDR improvement achieved with single-channel TSE, with the combination of TSE and beamforming. The results with oracle masks can serve as a topline for the performance.

[2] pb_bss toolkit https://github.com/fgnt/pb_bss

Table 5.3: Performance of target speech extraction in both single- and multi-channel settings, using mask-based beamforming.

	ΔSDR [dB]
Single-channel TSE[3]	9.34
TSE + MVDR	**12.67**
TSE + MWF	11.75
MVDR (oracle)	14.80
MWF (oracle)	12.35

Table 5.4: Performance of target speech extraction in multi-channel setting, using mask-based beamforming and spatial clustering. Performance is also shown for different gender pairs of target and interfering speaker. *F* denotes female, *M* denotes male speaker.

ΔSDR [dB]	all	F-F	M-M	F-M	M-F
TSE + MVDR	12.67	11.11	11.15	13.76	13.89
CACGMM + MVDR	11.54	11.29	12.02	11.61	11.29
TSE + CACGMM + MVDR	**13.54**	**12.82**	**12.36**	**14.34**	**14.17**
MVDR (oracle)	14.80	14.96	14.88	14.83	14.82

The TSE mask-based beamforming improves over the single-channel performance by about 2 dB to 4 dB, which brings it more than halfway to the oracle beamforming performance. As in the oracle experiments, MVDR performs better than MWF, although the gap is smaller than in the oracle case.

Finally, we compare the performance with spatial clustering and explore the combination of TSE with spatial clustering. The results are shown in Table 5.4. The spatial clustering with CACGMM by itself slightly under-performs the beamforming based on TSE. However, in mixtures consisting of speakers of the same gender (*F-F*, *M-M*), CACGMM works better. This is because TSE works with spectral information, therefore it has more difficulties when the two speakers have a similar voice. The combination of TSE and CACGMM combines the advantages of both methods and outperforms both of them. In the case of different-gender mixtures (*F-M*, *M-F*), the performance is approaching that of MVDR based on an oracle mask.

Figure 5.4 shows an example of masks estimated by different methods. In this case, the mask predicted by CACGMM is inaccurate at some parts, especially in lower frequencies. TSE mostly correctly identifies the target speaker, however, also includes small noise (e.g. in the final silence part of the signal). TSE combined with CACGMM corrects these mistakes and leads to the most accurate mask.

In Section 3.3, we claimed the independence on the number of components as one of the advantages of TSE compared to speech separation. However, in the combination with spatial clustering, it is necessary to set the number of components of CACGMM, which could violate this claim. In the next experiment, we show that the spatial clustering is actually not sensitive to the number of components and that it can be set to a higher number with-

[3]Note that this result does not correspond to results in experiments presented in previous chapter as here, we use SMS-WSJ dataset.

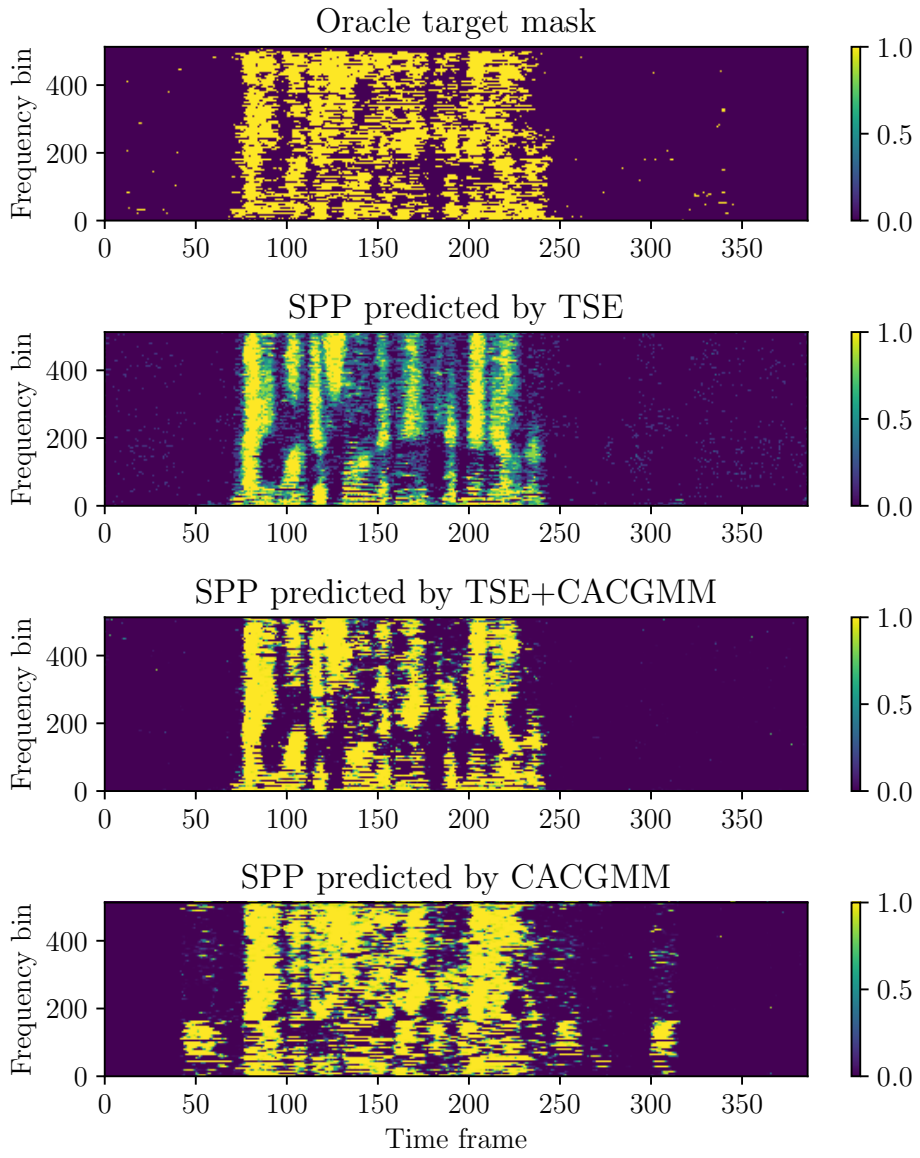


Figure 5.4: Example of time-frequency masks (i.e. speech presence probabilities (SPP)) estimated by spatial clustering and target speech extraction.

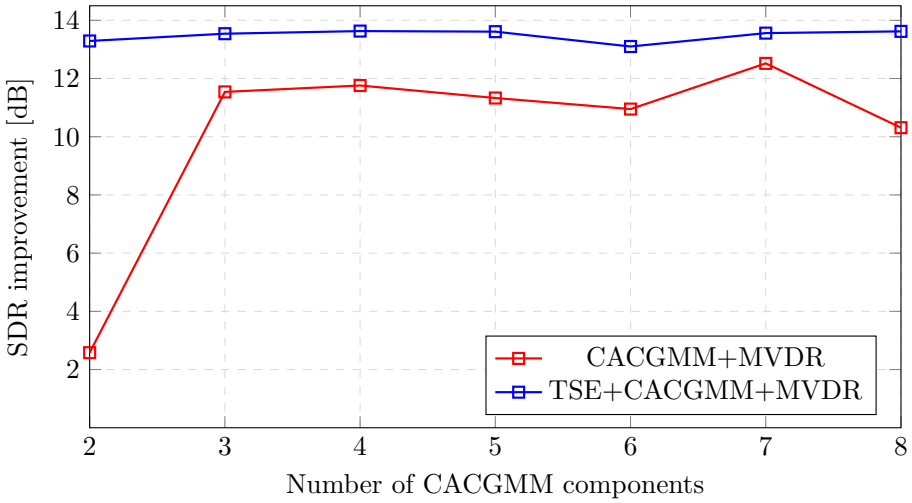


Figure 5.5: Performance of spatial clustering with and without target speech extraction as a function of number of components of CACGMM.

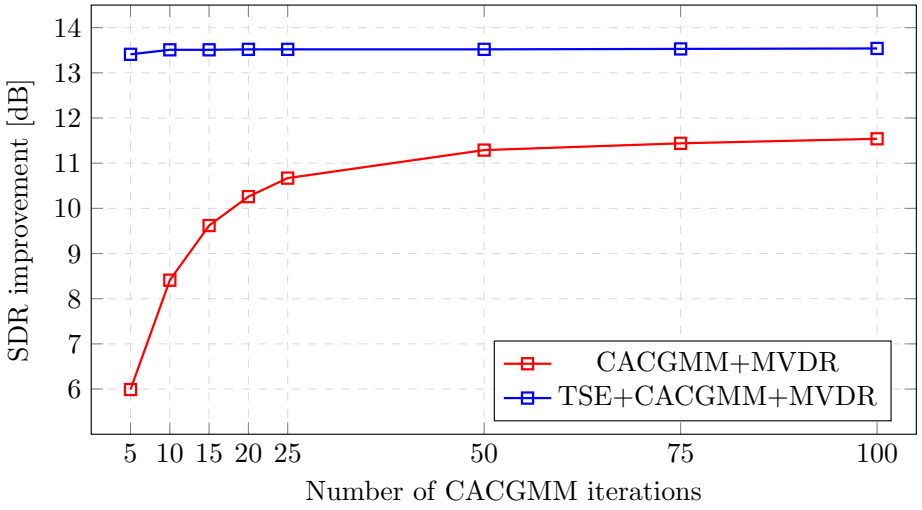


Figure 5.6: Performance of spatial clustering with and without target speech extraction as a function of number of iterations in CACGMM inference.

out hurting the performance. Figure 5.5 shows the SDR improvements of TSE combined with spatial clustering for different numbers of components in CACGMM. Note that the correct number of components is three, for two speakers and noise. The performance stays approximately constant even when setting more than twice more components than present in the mixture.

In the last experiment, we explore the performance as a function of the number of iterations of the spatial clustering. Figure 5.6 shows that as few as five iterations are already sufficient for a good performance (13.41 dB). This is in contrast with CACGMM, for which the performance increases steeply until 50 iterations (11.29 dB). Of course, for TSE+CACGMM+MVDR, part of the computational burden is on the forward pass through the neural network, which is not present in CACGMM+MVDR.

Chapter 6

Application of target speech extraction

In previous chapters, we considered target speech extraction as an isolated task, evaluated based on the quality of the output signal. Although the output of the extraction itself is useful in some applications (e.g. hearing aids), we often aim to improve the performance of other tasks. In this chapter, we focus on two such tasks, i.e. automatic speech recognition (ASR) and speaker diarization. Note that this chapter should not serve as an exhaustive overview of the research in either of the two tasks. We rather aim to provide a high-level overview of how the tasks can be solved and how target speech extraction can be used to improve their performance.

6.1 Automatic speech recognition

6.1.1 Single-speaker automatic speech recognition

Automatic speech recognition aims to automatically transcribe a speech signal into text, i.e. recognize what has been said. In the past, the research progressed from simple tasks such as recognizing spoken digits [DBB52] to more complicated scenarios with a large vocabulary and conversational speech [You96]. Recently, the focus has also moved to scenarios with a higher amount of interference such as noise, reverberation or interfering speakers [WMB+20].

Nowadays, there are two main directions in ASR research: hybrid [HDY+12] and end-to-end ASR systems [WWL19]. In the hybrid setting, the system consists of several modules focusing on different parts of the task, such as acoustics, language, and pronunciation. In contrast, end-to-end approaches aim to tackle the entire problem with one model. This model is predominantly a neural network that accepts a sequence of features at its input and outputs a sequence of characters. Although the research in end-to-end systems is booming today, the hybrid systems are still on-par with end-to-end on many tasks [ALM20, LWG+20]. In this work, we perform our experiments using a hybrid system, although it could be analogously applied also to end-to-end ones [DWO+19].

The hybrid system has three main components - the acoustic model, language model, and pronunciation model. The language model encodes the language information, that is which sequences of words are likely to appear in the language or domain. Typically, it is either a simple n-gram probabilistic model or a neural network predicting the next word. The pronunciation model maps between words and phonemes and is mostly represented by

a dictionary encoding this mapping. Our main focus in this work is on the acoustic model mapping between acoustic features and fundamental speech units, such as phonemes.

The most commonly used acoustic features are Mel-filterbanks or Mel-frequency cepstral coefficients (MFCC) which mimic the properties of human perception. The acoustic model itself is a neural network mapping the sequence of the acoustic features into a sequence of Hidden Markov model (HMM) state probabilities. The loss function to train the network is so-called lattice-free maximum mutual information (LF-MMI) loss [PPG+16] defined for one utterance as

$$\mathcal{L}_{\text{MMI}} = -\log \sum_{\omega \in \text{num}} p(\omega|O) = -\log \frac{\sum_{\omega \in \text{num}} P(\omega) \exp(\sum_n \psi_{n,\omega_n})}{\sum_{\omega' \in \text{den}} P(\omega') \exp(\sum_n \psi_{n,\omega'_n})}, \tag{6.1}$$

where num is the set of HMM state sequences corresponding to the ground-truth transcription of the utterance and den is the set of all possible HMM state sequences (in practice, only approximated). Further, O is the sequence of input features and $\psi_{n,s}$ is the potential predicted by the neural network for time-frame n and state s based on the input features.

The acoustic, language and pronunciation models are combined into a recognition network, usually represented as a weighted finite-state transducer [Moh97]. Obtaining a hypothesis given a sequence of acoustic features then corresponds to finding the best path in the transducer, usually referred to as decoding.

6.1.2 Combination with target speech extraction

Automatic speech recognition models degrade significantly in presence of interfering speakers. This type of interference is especially harmful as, in contrast with background noise, the interfering speech has the same characteristics as the target speech, so the ASR system cannot differentiate well between the target and interference. In this work, we aim to tackle this problem by employing a pre-processing done by target speech extraction. The resulting system thus consists of two stages: First, target speech extraction takes the mixed speech signal and the enrollment utterance and extracts the speech signal of the target speaker. Second, the extracted speech is processed by the ASR system, i.e. acoustic features are extracted, passed through the acoustic model, and decoded to obtain the final hypothesis.

Both stages of this modular system are typically trained separately with different objectives. The target speech extraction is trained with loss functions measuring discrepancy between the estimated and the reference signal, as presented in Section 4.5. The ASR system is typically trained on single-speaker signals with the objective of matching reference transcription, as shown in Equation (6.1). This may be sub-optimal as the model used for target speech extraction has limited capacity and thus needs to do trade-offs for minimizing the loss. The trade-offs might be different for optimizing the output signal purity as opposed to optimizing the transcription by ASR. In practice, this usually manifests itself as small artifacts, such as speech distortion, present in the output of TSE, which then hurt the accuracy of the ASR.

These problems can be solved by considering both systems jointly during the training. There are different options of what we can do:

1. *Fine-tuning the acoustic model on the outputs of the TSE system.* By doing this, the acoustic model may learn to disregard the artifacts in TSE's output.

2. *Fine-tuning TSE system with the ASR objective.* In this case, we need to compute the gradients of the LF-MMI loss with respect to the parameters of the TSE network.

This means backpropagating through the acoustic model and the feature extraction stage. In this case, the TSE can learn to output a signal that is optimal given the ASR system as opposed to the signal purity measures (for instance a different trade-off between interference reduction and speech distortion).

3. *Fine-tuning both systems jointly.* Similarly to the previous step, we update TSE with the ASR objective. However, we also keep updating the ASR system itself. This option gives the most freedom to the models to adjust to each other. However, it is also the least interpretable as the boundary between both models becomes blurred.

6.1.3 Evaluation of automatic speech recognition

To evaluate the ASR system, the hypothesis and reference sequences of words are compared. The widely used metric is word error rate (WER). This measures the percentage of correctly recognized words. More precisely, it is defined as

$$WER = \frac{S + I + D}{R} = \frac{S + I + D}{S + D + C}, \tag{6.2}$$

where:

- S is the number of substituted words
- I is the number of inserted words
- D is the number of deleted words
- R is the number of words in reference transcription
- C is the number of correctly recognized words.

To compute the above entities, dynamic string alignment is used to align the hypothesis and the reference transcription.

6.2 Experiments with automatic speech recognition

6.2.1 Dataset and configuration

We perform experiments on WSJ0-2mix dataset introduced in Section 4.8.1. For the test, we use the *max* version of the mixtures, where the mixture is of the length of the longest utterance. This is necessary not to cut any word in half and thus to obtain meaningful ASR results. The target speech extraction system corresponds to the settings described in Section 4.8.3.

For training of the ASR system, we used PyChain toolkit[1] [SWPK20] and followed the WSJ recipe[2]. We trained the system on clean single-speaker data from the training and validation set of WSJ0-2mix. The acoustic model was a time-delay neural network (TDNN) [PPK15] with 5 layers. The basic idea of TDNN architecture is depicted in Figure 6.1. The convolutional layers have filters with kernel of size 3, stride $[1, 1, 1, 1, 3]$ and dilation $[1, 1, 3, 3, 3]$ in 5 layers respectively. Each layer is followed by a dropout with a probability

[1] PyChain toolkit https://github.com/YiwenShaoStephen/pychain

[2] PyChain WSJ recipe https://github.com/YiwenShaoStephen/pychain_example/tree/master/examples/wsj

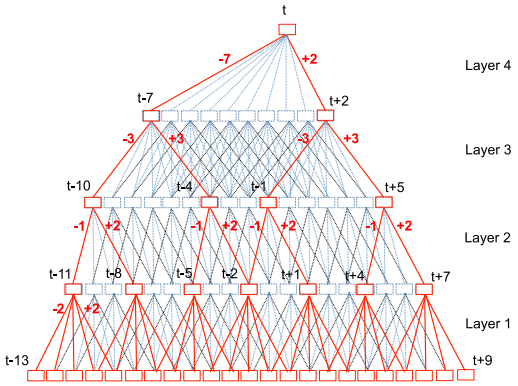


Figure 6.1: Time-delay neural network architecture. Image from [PPK15].

of 0.2. The input of the model are 40-dimensional MFCC features with global mean and variance normalization. The system is trained for 20 epochs with Adam optimizer and learning rate 1×10^{-3} and the best model is chosen based on validation loss. For fine-tuning, we use the learning rate 1×10^{-5} in case of ASR fine-tuning and 1×10^{-4} for TSE fine-tuning and joint fine-tuning. These values were found empirically according to the validation loss. Both TSE and ASR are initialized from the pre-trained models.

For decoding, tri-gram language model is used with CMU pronunciation dictionary. This is subsequently re-scored with a four-gram language model. For building the numerator and denominator graph, 84 mono-phone units are used.

6.2.2 Results

Table 6.1 shows the results of the ASR experiments measured by WER. First, the performance of the ASR system on single-speaker data and on unprocessed mixtures serves as the top- and bottom-line of the performance. Recognizing extracted signals obtained with a TSE system trained with SI-SDR loss leads to 20.64% WER, which is a great improvement from the mixture performance of 81.94%. The last three rows show the fine-tuning of either the TSE, the ASR system, or both jointly. All three options further reduce the error. We however see that fine-tuning the ASR system is more efficient than fine-tuning the TSE. One possible reason for this is that it is easier for the ASR system to adjust to the artifacts produced by the TSE, than for the TSE to stop producing such artifacts. Another explanation might be simple optimization issues as the error based on the ASR loss needs to be backpropagated through more transformations to reach the TSE model parameters. Furthermore, the best performance is achieved when both systems are updated jointly. The WER of 15.86%, in this case, is getting relatively close to 10.75% WER of recognizing clean single-speaker signals.

To get a better idea of how the behavior of the model changes during the fine-tuning, we can explore the SI-SDR of the signals extracted by the TSE block. The original pre-trained system achieves 17.17 dB on the *max* test-set of WSJ0-2mix. We can see that fine-tuning the system with the ASR objective hurts the SI-SDR. The degradation is much stronger in the case when only TSE is fine-tuned and thus the ASR system cannot adjust to the extracted signals. Although the SI-SDR value is as low as 5.93 dB, the system achieves better ASR performance. This shows that the SI-SDR does not perfectly reflect how well the signal will be recognized by the ASR system. Figure 6.2 shows an example of a signal segment,

Table 6.1: Automatic speech recognition results on WSJ0-2mix.

Method	fine-tuning TSE	fine-tuning ASR	WER [%]	ins [%]	del [%]	sub [%]	ΔSI-SDR [dB]
Single-speaker	-	-	10.75	0.63	2.12	7.99	∞
Mixtures	-	-	81.94	7.80	42.06	32.09	0.00
Extracted	✗	✗	20.64	2.20	6.38	12.06	**17.17**
Extracted	✓	✗	18.07	2.38	4.66	11.03	5.93
Extracted	✗	✓	17.09	2.78	**3.73**	10.57	**17.17**
Extracted	✓	✓	**15.86**	**1.61**	4.45	**9.80**	11.78

that is not well recognized by the ASR system when extracted by the pre-trained system, although having a high SI-SDR value. After fine-tuning the TSE block, the SI-SDR value substantially deteriorates, however, the ASR hypothesis improves. In the spectrogram, we can see that the signal extracted with the pre-trained system is very close to the reference. In some regions, the system however over-removes parts of the target signal. Although this difference is very small, it is detrimental for the ASR, in contrast with the output of the re-trained system, which leaves more interference in the signal.

6.3 Speaker diarization

6.3.1 Task of speaker diarization

Speaker diarization is the task of determining "who spoke when" [ABE+12]. The input is an audio recording of a conversation and the diarization system should determine how many speakers are present and label the segments spoken by each of these speakers. Such a system is useful for instance as a pre-processing for subsequent ASR, which usually assumes the segments and corresponding speaker identities as given. In some works, the diarization task is handled together with the task of voice activity detection (VAD), in others, the ground-truth (oracle) VAD is used. In our work, we will focus on the latter.

The diarization works today mostly fall into one of two categories: diarization based on clustering [SGR14, LPDB22], and end-to-end diarization [HFW+20]. The clustering-based approaches today commonly work with x-vectors (Section 4.2), as these encode well the speaker information, as shown in speaker verification literature. The recording is split into short segments, x-vector embedding is extracted from each segment, and finally, the x-vectors are clustered. This can be done for example by Agglomerative hierarchical clustering (AHC) or using Bayesian HMM and variational Bayes inference (VBx) [LPDB22]. The VBx method will be detailed in the next sections. End-to-end diarization models rely on a neural network that directly maps the acoustic features to speaker activities. Although the end-to-end approaches are quickly rising nowadays, the clustering-based approaches still provide competitive results [LPDB22].

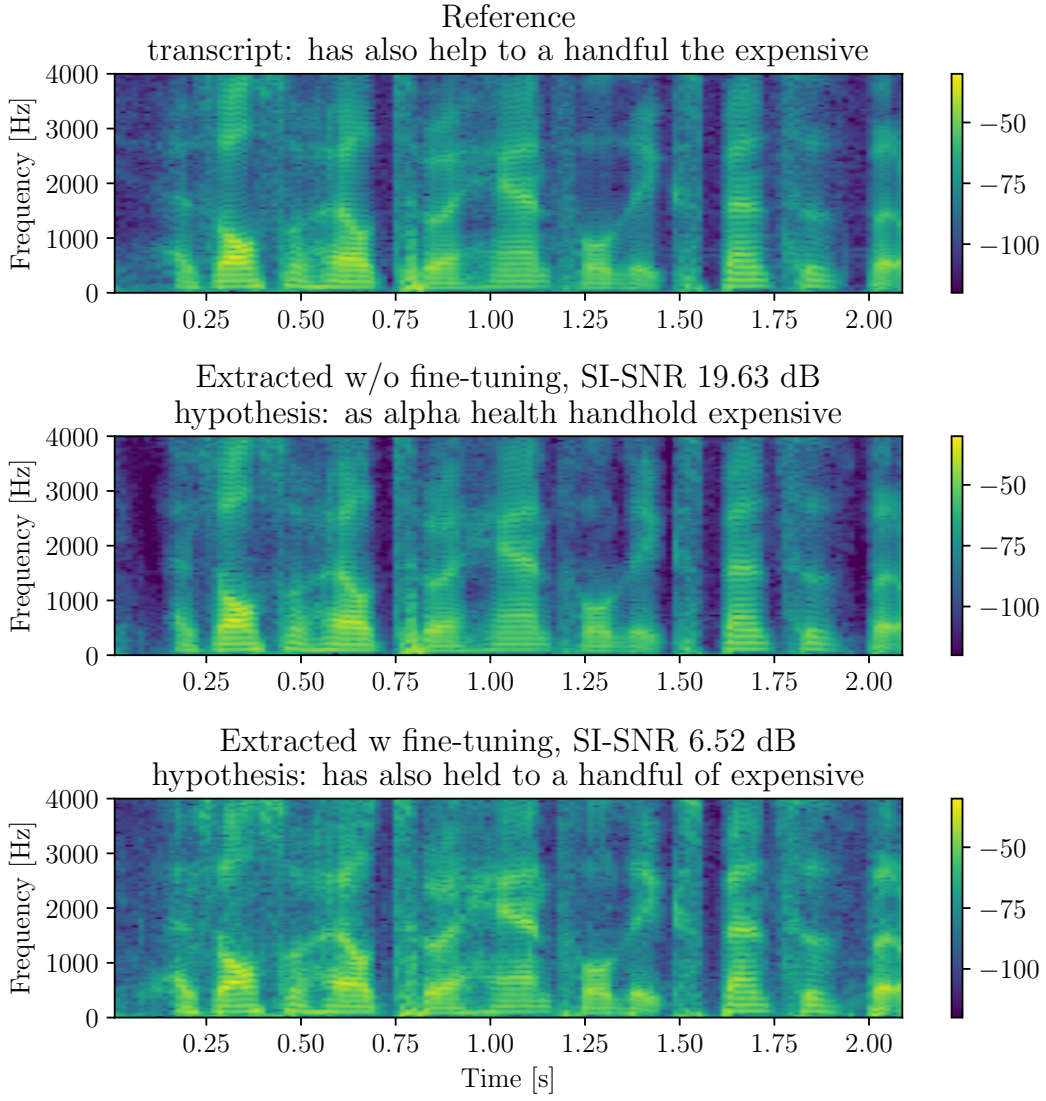


Figure 6.2: Example of utterance segment where ASR hypothesis of the original TSE output is wrong despite high SI-SDR metric. ASR hypothesis after fine-tuning TSE block improves despite the degradation in SI-SDR.

6.3.2 Evaluation of speaker diarization

The most popular evaluation metric for speaker diarization is the Diarization error rate (DER). This is defined as:

$$DER = \frac{SR + FA + Miss}{Total\_speech}, \tag{6.3}$$

where:

- SER is speaker error, amount of time attributed to incorrect speakers,
- FA is false alarm, the amount of time which is attributed to a speaker in the non-speech region or attributed to more speakers than present,
- $Miss$ is missed speech, the amount of time speech is not attributed to any speaker or multi-speaker speech is attributed to a lower number of speakers,
- $Total\_speech$ is total amount of speech, accounting for speaker overlaps.

6.3.3 Bayesian HMM-clustering of x-vector sequences (VBx)

In this work, we build upon the VBx clustering method, which has been shown to provide excellent results across several tasks [LPDB22] and was often used in leading systems in diarization evaluations [LWD+20, LGM+21]. The input of the VBx method is a sequence of x-vectors, extracted from typically around 1 s long segments. The goal is to infer the number of speakers and cluster the segments.

The VBx approach assumes that the sequence of x-vectors $\mathbf{X}$ is generated by a Hidden Markov Model (HMM). Each state of the HMM corresponds to one speaker. The topology of the HMM is depicted in Figure 6.3. The switching between speakers is controlled by the transition probabilities π_s. These are learned and will turn zero in case there are fewer speakers than the states in the HMM. In this way, VBx can determine the number of speakers in the recording. The latent variable $\mathbf{Z}$ represents the assignment of segments to the states (i.e. speakers). The emission probability distribution for each state is derived from a pre-trained Probabilistic linear discriminant analysis (PLDA) model. On a high level, each state emission probability distribution is represented by a speaker latent vector $\mathbf{y}_s$ of the same dimensionality as the x-vector. For details about the emission distributions and the whole model, we refer to [LWD+20]. The entire model can be described by the joint probability factorization:

$$p(\mathbf{X}, \mathbf{Z}, \mathbf{Y}) = p(\mathbf{X}|\mathbf{Z}, \mathbf{Y})p(\mathbf{Z})p(\mathbf{Y}) = \prod_t p(\mathbf{x}_t|z_t) \prod_t p(z_t|z_{t-1}) \prod p(\mathbf{y}_s), \tag{6.4}$$

where $p(\mathbf{x}_t|z_t = s) = p(\mathbf{x}_t|\mathbf{y}_s)$ is the emission probability distribution of state s derived from PLDA model, $p(z_t|z_{t-1})$ are the transition probabilities and $p(\mathbf{y}_s)$ is the prior distribution on speaker latent vectors.

The goal of the inference with the model is to obtain $p(\mathbf{Z}|\mathbf{X})$, i.e. the speaker activities. For this, Variational Bayes (VB) inference is used with the mean-field approximation $q(\mathbf{Z}, \mathbf{Y}) = q(\mathbf{Z})q(\mathbf{Y})$. Variational Bayes maximizes the evidence lower bound objective

$$\mathcal{L}_{\text{ELBO}} = F_A \mathbb{E}_{q(\mathbf{Z},\mathbf{Y})}[\ln p(\mathbf{X}|\mathbf{Y}, \mathbf{Z})] + F_B \mathbb{E}_{q(\mathbf{Y})}\left[\ln \frac{p(\mathbf{Y})}{q(\mathbf{Y})}\right] + \mathbb{E}_{q(\mathbf{Z})}\left[\ln \frac{p(\mathbf{Z})}{q(\mathbf{Z})}\right]. \tag{6.5}$$

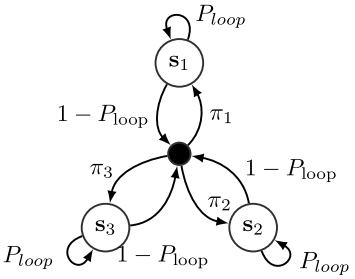


Figure 6.3: Topology of Hidden Markov model used in VBx method for speaker diarization. P_{loop} denotes probability of self-loop, π_i is the probability of entering state s_i. Image from [LPDB22].

The above equation is the proper lower bound for the likelihood in the case of $F_A = F_B = 1$. Empirically, tuning the values of the factors F_A,F_B can however help improve the performance. The inference maximizing Equation (6.5) proceeds by iterative updates of $q(\mathbf{Y})$ and $q(\mathbf{Z})$, that is updating the speaker models and updating the speaker activity, respectively. We refer to [LWD+20] for the readers interested in the update formulae.

One drawback of the VBx method is that it does not deal with overlap. Each frame is assumed to be generated from one speaker only. Some recent evaluations of diarization in challenging conditions show that a big part of the remaining error is indeed from the segments with more than one speaker [LGM+21]. Methods that can deal with overlap are thus needed.

6.3.4 Combination with target speech extraction

As mentioned in the previous section, VBx diarization has issues in the case when multiple speakers are overlapping. A target speech extraction system could be used to help to resolve this issue. On the other hand, in target speech extraction, there is a need for enrollment utterance. One way to obtain enrollment utterances in the case of conversational data is to identify the speakers' segments using diarization. The diarization and target speech extraction systems can thus benefit each other, and it is natural to combine them.

The simplest way to use target speech extraction for diarization purposes is by following three steps. First, running a preliminary VBx diarization system to identify segments that can be used as enrollment utterances. Second, using the TSE system with the enrollment utterances to extract the speech of each speaker from the recording. Third, running a voice activity detector on the output to obtain the activity of each speaker. Such a process is however problematic for two reasons:

1. *Absent speaker problem.* The TSE system, as we have defined it in previous chapters, is trained on recordings where the target speaker is always present. For instance, given the mixture of speaker-A + speaker-B, we ask the system to extract either speaker-A or speaker-B, but not a different speaker-C. The system is thus trained to resolve the issue of *Which of the speakers in the input is the target one?*, but not *Is the target speaker present in the mixture?*. The latter is arguably a more difficult task. Training of the TSE system including the absent speaker case is possible and has been explored in the literature [BXLS21]. It however hurts the performance in the usual present-speaker cases.

2. *Solving more difficult task.* In the above-described process, the TSE step needs to internally solve both diarization (where each speaker is speaking) and the extraction in the overlapped parts. This is apparently a more difficult task than the diarization itself. It is thus likely that a system aiming to solve diarization directly will always work better than a system that has the additional burden of outputting the exact signals.

For the above reasons, we focus here on a different way to combine the systems, where both work in tandem, and solve only the task for which they are specialized. This combination is carried out in the following steps:

1. Preliminary VBx diarization is run to identify segments that can be used as enrollment utterances.

2. TSE system uses the enrollment utterances to extract each speakers' speech from the recording.

3. VBx diarization is run again on all extracted signals jointly.

4. Speaker activities obtained on each extracted signal in the previous step are combined with union operation.

Figure 6.4 schematically depicts an example.

Using the above steps, it does not matter if TSE extracts the wrong speaker for the absent speaker cases, as the final diarization step can assign them to the correct speaker cluster. On the other hand, in the overlapped parts which VBx alone cannot solve, TSE will extract different speakers in the respective outputs, and using the final diarization and union combination will lead to identifying both speakers in the final output. Both systems here are used in line with their specialization, i.e. TSE helps by extracting speakers from overlap regions and VBx clusters single-speaker segments.

For step 3 of the process, we need to run the diarization on all extracted signals jointly. Practically, this can be simply done by concatenating the extracted signals and running the standard VBx algorithm on the result. We observed that when doing such prolongation of the input sequence we need to divide the factor F_A by the number of concatenated signals. This counteracts the artificial dependencies introduced into the sequence.

Furthermore, in step 3 of the process, the final diarization run can re-use the results of the initial run. In particular, the speaker models inferred by the initial diarization can be used to initialize speaker models in the final diarization. We will compare three different cases: 1) the final diarization is run from scratch, 2) we initialize the speaker models, 3) we initialize speaker models and do only one step of inference of the speaker activities. The final option is especially interesting, as it does not run the iterative diarization, but only re-estimates $q(\mathbf{Z})$ based on the existing speaker models. It thus does not add much computational overhead.

As shown before, the outputs of TSE may contain small artifacts which are not well processed by a system that was not trained on these extracted recordings. To combat this problem, we follow a strategy suggested in speech enhancement [DSA20]: we add the original observed signal with a small weight to the extracted signal. In our case, it means that in case of overlap, the final signal will contain both speakers, but the one extracted by the TSE system will be much stronger.

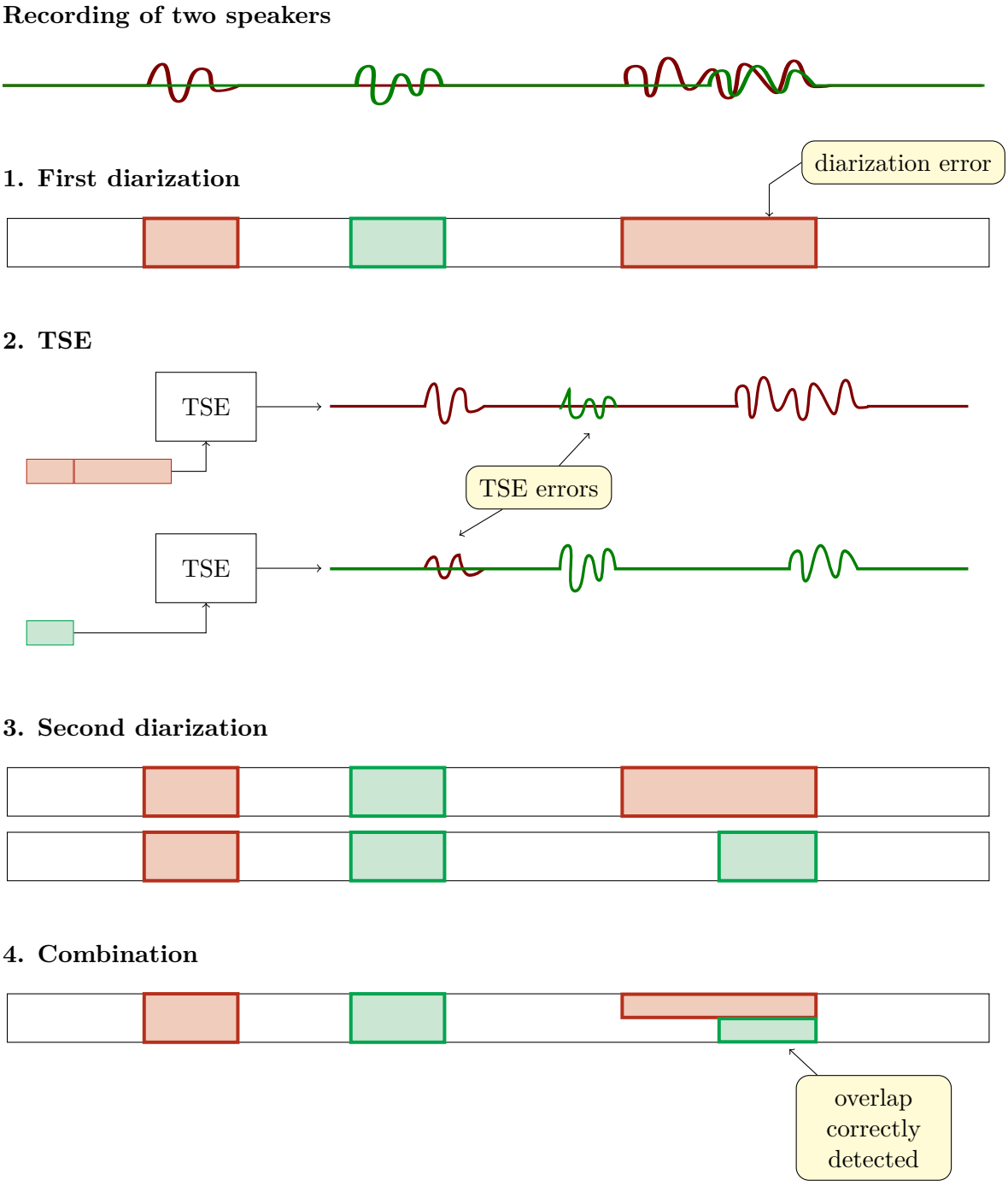


Figure 6.4: Example of combined processing of TSE and VBx diarization. VBx in the first step makes a diarization error as it cannot handle overlap. TSE makes small mistakes in areas of absent target speaker. The combination of both is able to correct these mistakes and leads to correct labeling of the overlapped part.

6.4 Experiments with speaker diarization

6.4.1 Datasets and configuration

For the diarization experiments, we use NIST SRE 2000 CALLHOME dataset [CCD+08] widely used in the diarization field. It consists of 500 recordings of conversational telephone speech. There are 2 to 7 speakers per recording, although the majority of the recordings contain 2 to 3 speakers. Overall, the length of the data is 15 hours, discounting silence parts. In all experiments, we use ground-truth (oracle) VAD labels. Three different setups are considered for evaluation, as in [LPDB22]: *fair*, which uses 0.25 s collar discounting small imprecision in segment boundaries, *full* with no collar, and *forgiving*, where 0.25 s collar is used and overlapped speech is not evaluated.

We follow the VBx experiments on CALLHOME described in [LPDB22] using the published recipe[3]. The used x-vector extractor corresponds to the one described in Section 4.8.3. The PLDA model is trained on x-vectors extracted from the same datasets as used for training of the x-vector extractor.

For the diarization itself, x-vectors are extracted from segments 1.5 s long with 0.25 s shift. They are centered, whitened and length-normalized. Agglomerative hierarchical clustering is run on the x-vectors to provide initialization of the speaker activities in VBx. For VBx itself, the dimensionality of x-vectors is reduced to 128 with LDA. The hyperparameters F_A, F_B, P_{loop} are set to 0.4, 17, 0.4, respectively, as found optimal in [LPDB22]. For the experiments combining TSE and VBx, we tune F_A and F_B using a two-fold partition of CALLHOME (note that this tuning however brings only marginal difference compared to the original values). For step 3 of the combined TSE and VBx experiments, we additionally modify F_A as described in Section 6.3.4.

For target speech extraction, we used the same architecture and training process as described in Section 4.8.3. However, as we cannot train on a matched dataset (full CALLHOME is used as a test-set), we train on four datasets to provide enough variability:

1.–2. *WSJ0-1mix* and *WSJ0-2mix* as described in Section 4.8.1.

3. *Libri2mix* dataset [CPC+20] contains simulated two-speaker mixtures based on LibriSpeech [PCPK15] corpus. It roughly follows conventions used when creating WSJ0-2mix. It however contains a wider range of 921 speakers (compared to 101 in WSJ0-2mix) in 364 hours of data. We use the clean version of the dataset without added noises, as CALLHOME is also relatively clean.

4. *VoxCeleb2Mix* is a dataset of artificial mixtures of two speakers we created based on VoxCeleb2 [CNZ18b]. We chose 6800 speakers for training, 152 for validation, and 100 for test-set. We roughly filtered the data according to simple SNR estimator [KS08] to choose the cleaner segments of the dataset. We simulated 50000 mixtures for training, 5000 for validation, and 3000 for test-set. The two speakers are mixed with SNR ranging from −5 dB to 5 dB.

During training, we randomly choose one of the four datasets for each element of the batch. The amount of data used from each dataset is thus balanced.

[3] VBx recipe https://github.com/BUTSpeechFIT/VBx

Database	ΔSI-SDR / SI-SDR[§] [dB]
WSJ0-1mix	37.67[§]
WSJ0-2mix	16.51
Libri2mix	14.63
Voxceleb2mix	9.74

Table 6.2: Target speech extraction results on test sets of four different databases. Reported numbers are SI-SDR improvements apart from single-speaker data, where absolute SI-SDR is used, denoted by [§].

6.4.2 Results

Since for the following experiments we trained a model on a combination of artificial datasets, we first report the performance on the respective test sets. Table 6.2 shows the SI-SDR results. The result on WSJ0-1mix shows that the model does not degrade single-speaker speech. On both WSJ0-2mix and Libri2mix, the model achieves good performance. In the case of WSJ0-2mix, the result is comparable to the improvement achieved by the model trained on matched dataset only (17.11 dB). The performance on Voxceleb2mix is worse; this is caused by more challenging conditions in the dataset. Although during its creation, we roughly filtered out very noisy recordings, the dataset may still contain data with reverberation or weaker noise. Furthermore, as the speakers in VoxCeleb are recorded in many conditions, the enrollment utterance can have different conditions than the mixture. Given these challenges, we find the performance on Voxceleb2mix satisfactory.

Next, we present the results of the diarization task itself. Table 6.3 shows the DER over different configurations of our system. All results are confirmed with the VBx baseline. For the combined experiments, we test four different settings. First, we do not add the original mixture to the extracted outputs and run the final diarization without any initialization. Second, we add the mixture with a weight of 0.2. This setting is then repeated with initialization of the speaker models and initialization with only one step of inference. We start by analyzing the results of the conditions considering overlap (*fair*, *full*). The results clearly show that the addition of the original mixture to the extracted output is necessary. This suggests that the diarization system does not handle well the extracted signals and maybe would benefit from re-training on the extracted signals (as in the previous ASR experiments). Here, we however explore the more simple solution of the addition of the observed signal. The three setups of the initialization of the speaker models perform very comparably. The best setup is to initialize the speaker models and run the iterative inference. However, the difference from performing only one step of the inference of the activities is rather negligible. This is good news, as this setting does not require much computation done in the final diarization step.

When we enable the system to detect also the overlapped parts, it is expected to get some degradation on single-speaker parts. The results on *fair* and *full* conditions combine both single-speaker and overlapped parts, thus include also the degradation of single-speaker regions. This degradation can be seen in the results on *forgiving* condition, that does not score the overlapped parts. Here, the combined system leads to about 1.5% DER degradation.

We can further analyze the results by breaking down the DER into different parts: missed speech (*miss*), false alarms (*fa*) and speaker error (*spke*). This break-down is shown

Table 6.3: Diarization results on CALLHOME using VBx and combination of VBx and target speech extraction.

Method	add mixture	speaker model init	DER [%] fair	full	forgiving
VBx baseline [LPDB22]	-	-	14.21	21.77	**4.42**
VBx + TSE	0	✗	24.30	30.84	22.63
VBx + TSE	0.2	✗	12.62	20.43	6.02
VBx + TSE	0.2	✓	**12.47**	**20.35**	5.94
VBx + TSE	0.2	✓(1 step)	12.57	20.42	5.92

Table 6.4: Break-down of diarization results on CALLHOME using VBx and combination of VBx and target speech extraction with fair evaluation.

Method	add mixture	speaker model init	Error rate [%] miss	fa	spke
VBx baseline [LPDB22]	-	-	10.11	**0.00**	4.10
VBx + TSE	0	✗	**4.73**	13.64	5.92
VBx + TSE	0.2	✗	7.04	1.72	3.86
VBx + TSE	0.2	✓	7.03	1.70	**3.75**
VBx + TSE	0.2	✓(1 step)	7.06	1.68	3.83

in Table 6.4. The biggest improvement of the combined system is caused by reducing missed speech. On the other hand, the false alarm rate increases. The increase in false alarm rate explains the degradation of DER in the *fair* condition.

Finally, we look at how the amount of overlap in the recording influences the results. Figure 6.5 shows the average DER for recordings with different amounts of overlap. For the combined system, we used the one with speaker model initialization, as it showed the best result. The results clearly show an upward trend with a higher amount of overlap, which confirms the difficulty of handling overlaps properly. Applying the combined system of TSE+VBx reduces the errors with a higher reduction on the files with higher amounts of overlap, as we would expect.

6.5 Using diarization labels to fine-tune speech recognition

In Section 6.2, we showed that training TSE neural network with the ASR loss can significantly improve the final ASR performance. For this type of fine-tuning, we however need transcriptions for the data on which we fine-tune. In this section, we explore whether we can fine-tune the TSE system using only speaker labels. In practice, these speaker labels can be obtained for example by using a diarization system.

For fine-tuning with speaker labels, we use a loss function composed of two parts — a speaker identity loss and a mixture consistency loss. The speaker identity loss forces the output to have the characteristics of the desired speaker. To evaluate how well the speaker characteristics match, we use x-vectors and PLDA. The mixture consistency loss forces all signals extracted from the mixture to sum back to the mixture. Such constraint naturally arises from the assumed mixing model and further restricts the network output.

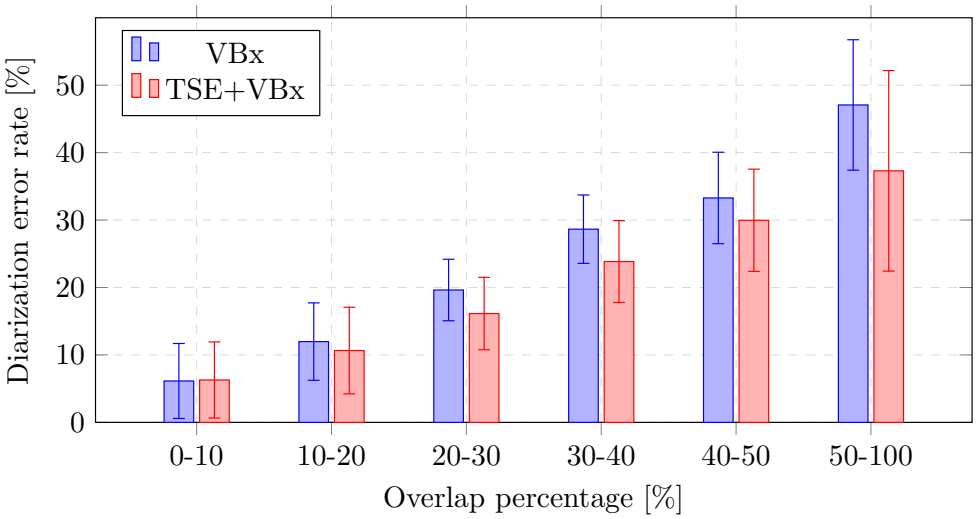


Figure 6.5: Break-down of diarization results on CALLHOME-fair for different amounts of overlap in the recordings. Error bars correspond to the standard deviation of the per-recording results in each overlap condition.

We can see the loss function as weakly supervised, as it does not require the strong supervision of parallel single-speaker data, but instead uses the weak supervision in the form of speaker labels. Weakly supervised loss functions have been previously explored also in the area of universal sound separation [PWLR20a, PWLR20b, KWS+20]. Notably, in [PWLR20a, PWLR20b] authors propose an objective function consisting of sound event classification and mixture consistency, similarly to our proposed objective function. Direct application of the loss from [PWLR20a, PWLR20b] to our task would however require training a speaker classifier on the adaptation data; we avoid this by using the generative PLDA model, that can be trained on a disjoint set of speakers and is considered state-of-the-art in speaker verification.

6.5.1 Weakly supervised loss with speaker labels

The proposed loss function uses supervision in the form of speaker characteristics. The speaker characteristics are obtained from a set of N_i segments of speech of the target speaker i, denoted as $\mathcal{S}_i^{\text{tgt}} = \{s_i^{(tgt,1)}, s_i^{(tgt,2)}, \ldots, s_i^{(tgt,N_i)}\}$. If using segmented speech corpus such as WSJ, the segments can be simply different utterances from the same speaker. In the case of long recordings such as meetings, the segments can be obtained from other parts of the recording where the speaker is speaking, after applying diarization.

We devise a loss function $\mathcal{L}_{\text{spk}}$ forcing the output of TSE for each speaker i to have the same speaker characteristics as $\mathcal{S}_i^{\text{tgt}}$. Forcing the correct speaker information however does not tie the output of TSE to the input mixture in any way. For this reason, we add an additional loss $\mathcal{L}_{\text{mix}}$ encouraging the mixture consistency. The full weakly supervised loss function is then

$$\mathcal{L}_{\text{wsup}}(\hat{s}_i(t), \mathcal{S}_i^{\text{tgt}}) = \lambda_{\text{spk}}\mathcal{L}_{\text{spk}} + \lambda_{\text{mix}}\mathcal{L}_{\text{mix}}, \tag{6.6}$$

where λ_{spk} and λ_{mix} are hyper-parameters weighting the parts of the loss.

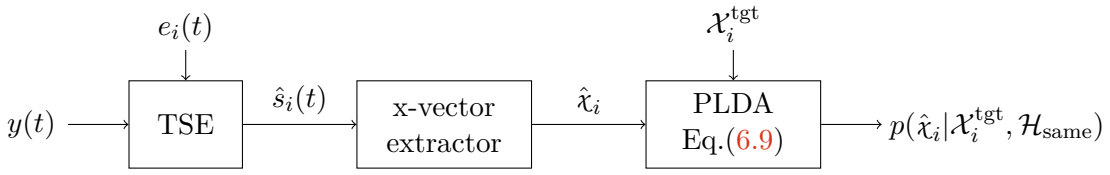


Figure 6.6: Speaker identity part of the auxiliary weakly supervised loss.

Speaker identity loss

To encourage the TSE estimate $\hat{s}_i$ to have the same speaker characteristics as $\mathcal{S}_i^{\text{tgt}}$, we employ concepts from speaker identification. Namely, we use x-vectors to represent the speaker characteristics and PLDA to model the x-vectors. Let us denote the x-vector extracted from $\hat{s}_i(t)$ as $\hat{\chi}_i$ and the set of x-vectors extracted from $\mathcal{S}_i^{\text{tgt}}$ as $\mathcal{X}_i^{\text{tgt}}$.

In PLDA, the distribution of x-vectors is modeled as

$$p(\mathbf{r}) = \mathcal{N}(\mathbf{r}; \mathbf{m}, \boldsymbol{\Sigma}_{\text{ac}}) \tag{6.7}$$

$$p(\chi|\mathbf{r}) = \mathcal{N}(\chi; \mathbf{r}, \boldsymbol{\Sigma}_{\text{wc}}), \tag{6.8}$$

where χ is the x-vector, $\mathbf{r}$ is the speaker mean, $\mathbf{m}$ is the global mean and $\boldsymbol{\Sigma}_{\text{ac}}$, $\boldsymbol{\Sigma}_{\text{wc}}$ are the across-speaker and within-speaker co-variance matrices. In the loss function, we aim to maximize the likelihood of the estimated x-vector $\hat{\chi}_i$ given the x-vectors $\mathcal{X}_i^{\text{tgt}}$, under the hypothesis $\mathcal{H}_{\text{same}}$ that both have been generated from the same speaker

$$p(\hat{\chi}_i|\mathcal{X}_i^{\text{tgt}}, \mathcal{H}_{\text{same}}) = \int p(\hat{\chi}_i|\mathbf{r}) p(\mathbf{r}|\mathcal{X}_i^{\text{tgt}})\, d\mathbf{r} = \mathcal{N}(\hat{\chi}_i; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{6.9}$$

$$\boldsymbol{\Sigma}_i = (\boldsymbol{\Sigma}_{\text{ac}}^{-1} + N_i \boldsymbol{\Sigma}_{\text{wc}}^{-1})^{-1} \tag{6.10}$$

$$\boldsymbol{\mu}_i = \boldsymbol{\Sigma}_i \left(\boldsymbol{\Sigma}_{\text{ac}}^{-1}\mathbf{m} + N_i \boldsymbol{\Sigma}_{\text{wc}}^{-1} \tilde{\mathbf{x}}_i\right), \tag{6.11}$$

where $\tilde{\mathbf{x}}_i$ and N_i are the mean and the number of x-vectors in $\mathcal{X}_i^{\text{tgt}}$. The equality follows Equation (213) in [Mur07] for computing predictive posterior distribution in case of multivariate Gaussian prior and Gaussian likelihood. The loss function is then the inverse log-likelihood summed over all speakers in the mixture

$$\mathcal{L}_{\text{spk}} = -\sum_{i=0}^{I-1} \log p(\hat{\chi}_i|\mathcal{X}_i^{\text{tgt}}, \mathcal{H}_{\text{same}}). \tag{6.12}$$

Note that $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can be pre-computed for each speaker in advance. The evaluation of the loss function is then the evaluation of the Gaussian p.d.f. of the estimated x-vector $\hat{\chi}_i$. Figure 6.6 shows the process for computing the speaker identification loss. The extraction of the x-vector including the feature extraction needs to be implemented in a differentiable way, which is possible using a toolkit such as PyTorch [PGM+19].

Mixture consistency loss

The mixture consistency loss reflects a property that should hold for the extracted sources, i.e. summing back to the original signal. This directly follows from the assumed mixing

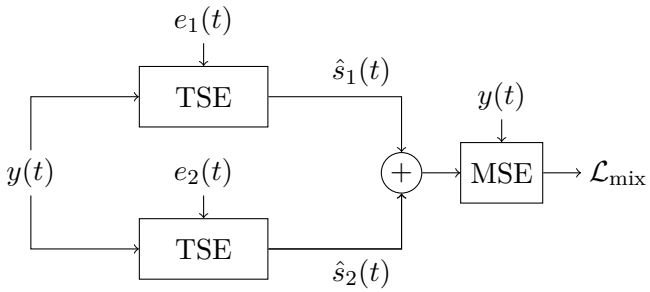


Figure 6.7: Mixture consistency part of the auxiliary weakly supervised loss.

model. To enforce this, we minimize the mean-square error between the sum of the extracted signals and the observed mixture in time-domain

$$\mathcal{L}_{\text{mix}} = ||y(t) - \sum_{i=0}^{I-1} \hat{s}_i(t)||^2. \tag{6.13}$$

Note that we neglect the noise factor $v(t)$ in the loss. This could possibly lead to the network learning to include the noise in the extracted sources. Extending the mixture consistency with a noise model is a possible future direction, that could be beneficial for very noisy conditions. The mixture consistency loss function is schematically depicted in Figure 6.7.

6.5.2 Overall steps

The proposed weakly supervised loss can be used for re-training a target speech extraction system on data, where for each mixture $y(t)$ and corresponding enrollment utterance $e_i(t)$, there is a set of segments $\mathcal{S}_i^{\text{tgt}}$ spoken by the target speaker. Here, we describe the steps we follow in our experiments with the proposed loss:

1. Train TSE with supervised loss. This step requires parallel single-speaker recording $s_i(t)$ for each mixture $y(t)$ and represents the baseline.
2. Re-train TSE initialized in step 1 with weakly supervised loss $\mathcal{L}_{\text{wsup}}(\hat{s}_i(t), \mathcal{S}_i^{\text{tgt}})$ on the same training data as in step 1, but using only weak supervision in the form of segments $\mathcal{S}_i^{\text{tgt}}$ spoken by the target speaker. The segments are taken from other utterances of the target speaker in the training data.
3. Re-train TSE with weakly supervised loss $\mathcal{L}_{\text{wsup}}(\hat{s}_i(t), \mathcal{S}_i^{\text{tgt}})$ on the target data. The target data consists of long recordings, so segments of the target speaker $\mathcal{S}_i^{\text{tgt}}$ are found by diarization.

The goal of step 2 is to evaluate the effect of the loss itself compared to the supervised loss, while in step 3, we explore whether it is possible to use the loss to adapt the system to the target data.

6.5.3 Dataset and configuration

The work described in this section was done as part of 2020 Seventh Frederick Jelinek Memorial Summer Workshop[4]. As a consequence, the model and training data used here

[4]2020 Seventh Frederick Jelinek Memorial Summer Workshop https://www.clsp.jhu.edu/workshops/20-workshop/

are not exactly consistent with the experiments in the rest of the thesis. The overall target speech extraction methodology however remains the same.

We use two different sources of data, i.e. artificially mixed short utterances for training, and long meeting-like recordings for testing and adaptation. The artificially mixed data are based on LibriSpeech dataset [PCPK15]. We simulate mixtures of two speakers. We will denote the mixed data as LibriSpeech-mix[5]. LibriSpeech dataset is also used to get enrollment utterances in all experiments. For testing and adaptation, we use the LibriCSS dataset [CYL+20] containing multi-channel recordings simulating conversations. For our experiments, we use the first channel only. Each recording was created using multiple utterances from LibriSpeech [PCPK15] from multiple speakers. The utterances were played back from a loudspeaker in a room. The recordings can be grouped into six different overlap conditions from 0% to 40% of overlap. Alternatively, the recordings can be grouped into 10 different sessions, where each session contains different speakers. Each session then contains one recording for each overlap condition. We use *session0* as the development set and the remaining sessions as the evaluation set.

The main network used for TSE corresponds to frequency-domain SpeakerBeam [ŽDK+19] (Section 4.7.1). It consists of 3 BLSTM layers, each with 600 units and 2 fully connected layers with ReLU activation. We apply the multiplication operation after the first BLSTM layer. The auxiliary network has 2 fully connected layers with 64 units, ReLU activation after the first layer. For supervised training, we used Adam optimizer with learning rate 1×10^{-3} and gradient clipping 1. We trained the network for 450k iterations with batch size 36. We used STFT with window size and shift of 512 and 128 samples. When extracting the target speech, we apply TSE by chunks of 10 seconds with 5-second shift.

We use x-vector extractor and PLDA model from VBx recipe[6], as described in [LPDB22] and Sections 4.8.3, 6.4.1.

For re-training the network with the proposed loss $\mathcal{L}_{\text{wsup}}$, we use Adam optimizer, with learning rate 1×10^{-6} and gradient clipping 1. We perform 80k iterations with batch size 1. We weigh both parts of the loss equally, setting $\lambda_{\text{spk}} = \lambda_{\text{mix}} = 0.5$, unless stated otherwise.

We use the hybrid HMM-DNN model from [RDC+21]. The acoustic model is a 17-layer factored TDNN [PCW+18] trained using the lattice-free MMI objective [PPG+16] (Section 6.1.1). The model was trained on the 960h Librispeech data with 3x speed perturbation, and additionally fine-tuned for 1 epoch on reverberated Librispeech data. We use the official 3-gram language model provided with Librispeech for decoding.

6.5.4 Results

We evaluate the target speech extraction performance with speech recognition on the LibriCSS dataset. We show results on the evaluation set (*all*) and on the condition with the highest amount of overlap (*OV40*). We compare four different setups: (1) unprocessed data without TSE applied, (2) baseline TSE trained with the supervised loss $\mathcal{L}_{\text{sup}}$ on LibriSpeech-mix, (3) TSE re-trained with the weakly supervised loss $\mathcal{L}_{\text{wsup}}$ on LibriSpeech-mix, and (4) TSE re-trained on target LibriCSS data with $\mathcal{L}_{\text{wsup}}$. The setups (2)-(4) correspond to steps 1-3 as described in Section 6.5.2. Note that for (4), there are no parallel data available.

For the experiments, it is necessary to have diarization outputs — first, for ASR decoding, and second, to get speaker labels when adapting to the target data. To avoid the influence of the diarization errors, we first perform experiments using oracle diarization.

[5] Note that this does not exactly correspond to LibriMix dataset described in Section 6.4.1.

[6] VBx recipe https://github.com/BUTSpeechFIT/VBx

Table 6.5: Speech recognition performance in terms of Word error rate (WER) on single-channel LibriCSS data using oracle diarization.

		$\mathcal{L}_{\text{wsup}}$	on target data	WER [%] all	WER [%] OV40
(1)	Mixtures	-	-	26.2	41.6
(2)	TSE	✗	✗	24.1	33.2
(3)	TSE	✓	✗	20.8	31.1
(4)	TSE	✓	✓	**20.3**	**30.2**

Table 6.6: Speech recognition performance in terms of Word error rate (WER) on single-channel LibriCSS data using RPN and TS-VAD diarization.

		$\mathcal{L}_{\text{wsup}}$	on target data	RPN all	RPN OV40	TS-VAD all	TS-VAD OV40
	DER [%]	-	-	9.5	14.2	7.6	9.5
	WER [%]						
(1)	Mixtures	-	-	31.2	47.2	28.4	42.4
(2)	TSE	✗	✗	30.1	42.3	27.4	36.3
(3)	TSE	✓	✗	27.0	40.0	24.3	33.7
(4)	TSE	✓	✓	**26.6**	**39.4**	**24.0**	**33.0**

The results are shown in Table 6.5. Comparing the results on unprocessed data (row (1)) with applying baseline TSE trained with the supervised loss $\mathcal{L}_{\text{sup}}$ (row (2)), we can see that the target speech extraction improves the ASR performance significantly, especially when a higher amount of overlap is present. Re-training the TSE system with the weakly supervised loss $\mathcal{L}_{\text{wsup}}$ on the original artificially mixed data (row (3)) improves the performance further. By exploring the outputs of original and re-trained TSE, we can see that after re-training with $\mathcal{L}_{\text{wsup}}$, the resulting speech contains slightly more noise, but less speech distortion. Such outputs may be more favorable for the ASR system. Note that the network in (2) is fully converged and training it longer with supervised loss does not yield better results. The improvements in (3) are thus not simply caused by longer training. Finally, re-training TSE directly on the target evaluation set, leads to further improvement, showing that it is possible to adapt to the target conditions using the speaker labels only.

Although not directly comparable, a separation using a similar network architecture and hybrid ASR back-end achieved a WER of 35.5% on OV40 condition in [CYL+20]. The performance of the proposed system could also be improved using more sophisticated network architectures for both front-end and back-ends as in [CYL+20].

In the second set of experiments, we used outputs of the diarization system rather than the oracle ground-truth diarization, to see whether errors in the speaker labels have a detrimental effect on the adaptation. We used two different diarization systems, i.e. region proposal network (RPN) [HWF+20] and target-speaker voice activity detection (TS-VAD) [MKP+20]. Table 6.6 shows the results of the ASR, together with the diarization error rates (DER) of the diarization systems. We can see that in both cases, the trends are similar as

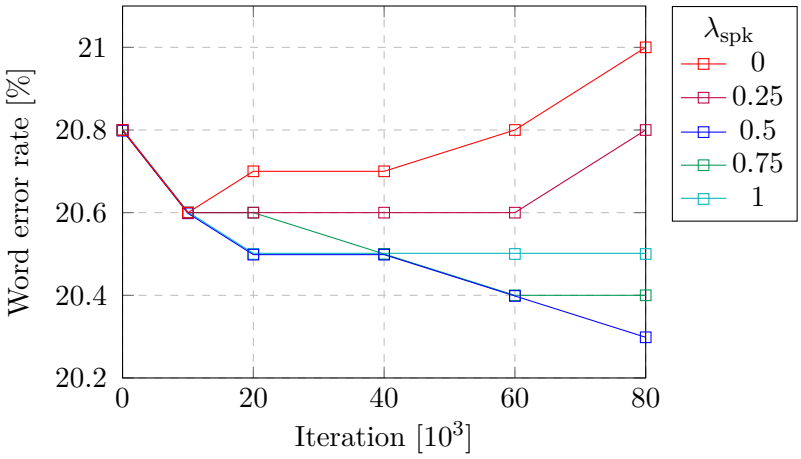


Figure 6.8: Speech recognition performance as a function of number of iterations during adaptation and different values of weight λ_{spk}. The mixture consistency weight λ_{mix} is set to $1 - \lambda_{\text{spk}}$.

with the oracle diarization. Including the proposed loss brings from 2.3% to 3.1% WER improvement, and adapting to the target data further improves the performance by 0.3-0.7% WER. The adaptation to the target data is thus not significantly affected by the diarization errors.

To understand better how the two parts of the loss function, defined in Section 6.5.1, affect the training, we experimented with different weights $\lambda_{\text{spk}}, \lambda_{\text{mix}}$ during the adaptation stage. Figure 6.8 shows the speech recognition performance as a function of the number of iterations performed. We set the weights so that $\lambda_{\text{mix}} + \lambda_{\text{spk}} = 1$. The results show greater importance of the speaker identity loss $\mathcal{L}_{\text{spk}}$. When the mixture consistency is dominant in the loss, the performance starts to worsen after 10k iterations. The speaker consistency loss leads to improvements even by itself, however, the best results are still obtained when both parts of the loss are balanced.

Chapter 7

Conclusion

This thesis presents target speech extraction as a way to tackle issues with interfering speakers which arise in many speech technologies. We offer target speech extraction as an alternative to speech separation, having several benefits, such as the needlessness of counting speakers in the mixture, avoiding permutation problems, or better consistency of the output for longer recordings. Our work shows the advantage of TSE over SS, analyzes its different aspects, suggests possible ways to combine it with multi-channel processing, and proposes application to automatic speech recognition and speaker diarization. We conclude the findings of the individual parts of our investigation below.

We first compared target speech extraction with speech separation; this showed the advantage of target speech extraction, especially in difficult conditions. Our experiments also showed better robustness of TSE to changing number of speakers in the mixture and stable performance with the increasing length of the recording. We also show the method is not too sensitive to the length of enrollment utterance of the target speaker. In further analysis, we show one weakness of both TSE and SS — decreased performance when the voice characteristics of the speakers in the mixture are very similar. We also studied how different aspects affect the system, including choice of speaker representation, the method to inform the network, input/output domain, or loss function.

We further pointed out the possibility to combine TSE with multi-channel methods and achieve an improved quality of the extracted speech signal. The first explored approach is mask-based beamforming with mask estimated using the TSE output. This simple approach already improves the performance substantially. Secondly, we demonstrate that the TSE can be combined with spatial clustering, which brings a further increase in performance. Although spatial clustering comes with prolonged inference time, we conclude that in combination with TSE, the iterative inference converges to a good solution much sooner. Additionally, according to our results, the combination of TSE and spatial clustering is not very sensitive to the number of components; this preserves the independence of TSE on the number of speakers in the mixture.

Finally, we present how TSE can be used to improve ASR performance. Further improvement of the accuracy is brought by fine-tuning either the TSE or ASR component, or ideally both of these jointly. We show that the fine-tuning of the TSE jointly with ASR leads to less aggressive suppression of the non-target parts of the signal, which leads to better recognition. We also introduced a way to combine TSE with clustering-based speaker diarization; this makes use of the strengths of both methods and leads to improved diarization performance on overlapped parts. Lastly, we proposed a weakly supervised auxiliary loss based on speaker labels, which can be used to fine-tune TSE for improved ASR

performance. We demonstrated that the loss can be also used for adaptation to a target domain since it requires only speaker labels as supervision.

7.1 Future directions

Robustness and real recordings

The target speech extraction proves to be a promising approach to pre-process multi-talker recordings. The biggest challenge today is to carry the performance over to realistic recordings. Most of the experiments in this thesis, and also generally in the literature, are obtained on artificially mixed recordings. This applies to both target speech extraction and speech separation. When presented with real recordings in challenging conditions, both methods often fail. This can be observed for instance from the recent CHiME-6 challenge [WMB+20] of automatic speech recognition and diarization in everyday environments, where none of the participants utilized recent neural-network-based separation or extraction methods[1].

One obstacle to applying these methods to real recordings is the difficulty of evaluation. For real recordings, parallel single-speaker data are not available, and as such, objective measures such as SI-SDR cannot be evaluated. It is possible to evaluate on a downstream task, such as ASR, but these results are difficult to analyze as many factors are influencing the final accuracy. One step towards solving this is the recent REAL-M dataset [SRCG21] that offers real recordings together with a pre-trained estimator of SI-SDR performance without the need for ground-truth. This could enable us to more closely analyze the performance of TSE on real recordings.

Another promising direction to improve the robustness is unsupervised training or adaptation on the real recordings. We made one step towards this direction in the last part of the thesis, where we presented a weakly supervised loss. Other approaches are appearing in speech separation literature, such as unsupervised training on mixed data [WTE+20], or utilizing spatial information as supervision [SWLRP19]. Some works attempted to use adversarial training, however without much success [Hos19]. This direction, in our opinion, is still open and could be explored further.

Finally, while the TSE and SS problems are nowadays mostly tackled using discriminative techniques, it is possible to approach them using generative models. We have explored this direction in the context of multi-channel speech separation in [ŽDB+21] and showed that it is possible to solve the separation problem as inference in the factorial generative model. Such a model offers higher interpretability and as a consequence, the possibility to adapt to different noise conditions. The model also includes a latent speaker variable which opens a way to apply it for target speech extraction.

Latency

Another issue not addressed in most of the current methods is latency. In some applications, this does not pose a problem, in others, such as hearing devices, the latency is a key factor. Advancements in neural network architectures which bring improved accuracy however often come with increased latency. The research direction of improving latency while keeping the same or slightly decreased performance is not very well explored. There are exceptions, for instance in the field of sound source separation [TWJS21] or challenges

[1] Proceedings of CHiME-2020 workshop https://chimechallenge.github.io/chime2020-workshop/programme.html

in speech enhancement with limited latency [RGC+20, GBC+21]. We however believe that this direction could be explored further.

Speaker discrimination

The final direction that could improve the current target speech extraction systems is better speaker discriminability. There are two aspects that should be addressed:

- *Better robustness to intra-speaker variability.* When the speakers are in different conditions, or in a different emotional state the system should still correctly identify them.
- *Better discrimination among speakers.* Especially when applied to more challenging conditions, the system can confuse different speakers. Improved discrimination among speakers would also enable tackling the absent speaker case, as discussed in Section 6.3.4.

These two aspects however are opposing and need to be traded off. To advance in this direction, simply training on datasets with more intra- and inter-speaker variability could help. Other than that, advances in speaker verification can motivate further research in TSE.

Bibliography

[AB79] Jont B Allen and David A Berkley. Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America*, 65(4):943–950, 1979.

[ABE+12] Xavier Anguera, Simon Bozonnet, Nicholas Evans, Corinne Fredouille, Gerald Friedland, and Oriol Vinyals. Speaker diarization: A review of recent research. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):356–370, 2012.

[ACZ19] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. My Lips Are Concealed: Audio-Visual Speech Enhancement Through Obstructions. In *Proc. Interspeech 2019*, pages 4295–4299, 2019.

[AD20] Ali Aroudi and Simon Doclo. Cognitive-driven binaural beamforming using eeg-based auditory attention decoding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:862–875, 2020.

[ALM20] Andrei Andrusenko, Aleksandr Laptev, and Ivan Medennikov. Towards a Competitive End-to-End Speech Recognition for CHiME-6 Dinner Party Transcription. In *Proc. Interspeech 2020*, pages 319–323, 2020.

[ALMD19] Axel Ahrens, Kasper Duemose Lund, Marton Marschall, and Torsten Dau. Sound source localization with varying amount of visual information in virtual reality. *PloS one*, 14(3):e0214603, 2019.

[AM08] Ibrahim Almajai and Ben Milner. Using audio-visual features for robust voice activity detection in clean and noisy speech. In *2008 16th European Signal Processing Conference*, pages 1–5. IEEE, 2008.

[And17] Asger Heidemann Andersen. Speech intelligibility prediction for hearing aid systems. 2017.

[Bar] Jon Barker. Distant microphone speech recognition in everyday environments: from chime-5 to chime-6. https://youtu.be/UQci8LgwZOc?t=2491.

[BC94] Guy J Brown and Martin Cooke. Computational auditory scene analysis. *Computer Speech & Language*, 8(4):297–336, 1994.

[BCH08] Jacob Benesty, Jingdong Chen, and Yiteng Huang. *Microphone array signal processing*, volume 1. Springer Science & Business Media, 2008.

[BHS+18] Christoph Boeddeker, Jens Heitkaemper, Joerg Schmalenstroeer, Lukas Drude, Jahn Heymann, and Reinhold Haeb-Umbach. Front-end processing for the chime-5 dinner party scenario. In *CHiME5 Workshop, Hyderabad, India*, volume 1, 2018.

[BMVW15] Jon Barker, Ricard Marxer, Emmanuel Vincent, and Shinji Watanabe. The third 'chime' speech separation and recognition challenge: Dataset, task and baselines. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 504–511. IEEE, 2015.

[BP92] AWa Bronkhorst and RJTJotASoA Plomp. Effect of multiple speechlike maskers on binaural speech recognition in normal and impaired hearing. *The Journal of the Acoustical Society of America*, 92(6):3132–3139, 1992.

[Bre94] Albert S Bregman. *Auditory scene analysis: The perceptual organization of sound*. MIT press, 1994.

[Bro15] Adelbert W Bronkhorst. The cocktail-party problem revisited: early processing and selection of multi-talker speech. *Attention, Perception, & Psychophysics*, 77(5):1465–1487, 2015.

[BWVT18] Jon Barker, Shinji Watanabe, Emmanuel Vincent, and Jan Trmal. The Fifth 'CHiME' Speech Separation and Recognition Challenge: Dataset, Task and Baselines. In *Proc. Interspeech 2018*, pages 1561–1565, 2018.

[BXLS21] Marvin Borsdorf, Chenglin Xu, Haizhou Li, and Tanja Schultz. Universal Speaker Extraction in the Presence and Absence of Target Speakers for Speech of One and Two Talkers. In *Proc. Interspeech 2021*, pages 1469–1473, 2021.

[CCCK20] Soo-Whan Chung, Soyeon Choe, Joon Son Chung, and Hong-Goo Kang. FaceFilter: Audio-Visual Speech Separation Using Still Images. In *Proc. Interspeech 2020*, pages 3481–3485, 2020.

[CCD+08] Fabio Castaldo, Daniele Colibro, Emanuele Dalmasso, Pietro Laface, and Claudio Vair. Stream-based speaker segmentation using speaker factors and eigenvoices. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4133–4136. IEEE, 2008.

[CK17] Jen-Tzung Chien and Kuan-Ting Kuo. Variational recurrent neural networks for speech separation. In *Proc. Interspeech 2017*, pages 1193–1197, 2017.

[CLM17] Zhuo Chen, Yi Luo, and Nima Mesgarani. Deep attractor network for single-microphone speaker separation. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, 2017.

[CNZ18a] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. VoxCeleb2: Deep Speaker Recognition. In *Proc. Interspeech 2018*, pages 1086–1090, 2018.

[CNZ18b] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. VoxCeleb2: Deep Speaker Recognition. In *Proc. Interspeech 2018*, pages 1086–1090, 2018.

[CPC+20] Joris Cosentino, Manuel Pariente, Samuele Cornell, Antoine Deleforge, and Emmanuel Vincent. Librimix: An open-source dataset for generalizable speech separation. *arXiv preprint arXiv:2005.11262*, 2020.

[ÇS06] Özgür Çetin and Elizabeth Shriberg. Analysis of overlaps in meetings by dialog factors, hot spots, speakers, and collection site: Insights for automatic speech recognition. In *Ninth international conference on spoken language processing*, 2006.

[CXY+18] Zhuo Chen, Xiong Xiao, Takuya Yoshioka, Hakan Erdogan, Jinyu Li, and Yifan Gong. Multi-channel overlapped speech recognition with location guided speech extraction network. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 558–565. IEEE, 2018.

[CYL+20] Zhuo Chen, Takuya Yoshioka, Liang Lu, Tianyan Zhou, Zhong Meng, Yi Luo, Jian Wu, Xiong Xiao, and Jinyu Li. Continuous speech separation: Dataset and analysis. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7284–7288. IEEE, 2020.

[DBB52] Ken H Davis, R Biddulph, and Stephen Balashek. Automatic recognition of spoken digits. *The Journal of the Acoustical Society of America*, 24(6):637–642, 1952.

[DG19] David Ditter and Timo Gerkmann. Influence of Speaker-Specific Parameters on Speech Separation Systems. In *Proc. Interspeech 2019*, pages 4584–4588, 2019.

[DHBHU19] Lukas Drude, Jens Heitkaemper, Christoph Boeddeker, and Reinhold Haeb-Umbach. Sms-wsj: Database, performance measures, and baseline recipe for multi-channel source separation and recognition. *arXiv preprint arXiv:1910.13934*, 2019.

[DHU17] Lukas Drude and Reinhold Haeb-Umbach. Tight integration of spatial and spectral features for bss with deep clustering embeddings. In *INTERSPEECH 2017, Stockholm, Schweden*, 2017.

[DKD+10] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2010.

[DKHN15] Marc Delcroix, Keisuke Kinoshita, Takaaki Hori, and Tomohiro Nakatani. Context adaptive deep neural networks for fast acoustic model adaptation. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4535–4539. IEEE, 2015.

[DMS+21] Chengyun Deng, Shiqian Ma, Yongtao Sha, Yi Zhang, Hui Zhang, Hui Song, and Fei Wang. Robust Speaker Extraction Network Based on Iterative Refined Adaptation. In *Proc. Interspeech 2021*, pages 3530–3534, 2021.

[DOŽ+20] Marc Delcroix, Tsubasa Ochiai, Kateřina Žmolíková, Keisuke Kinoshita, Naohiro Tawara, Tomohiro Nakatani, and Shoko Araki. Improving speaker discrimination of target speech extraction with time-domain speakerbeam. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 691–695. IEEE, 2020.

[DSA20] Alexandre Défossez, Gabriel Synnaeve, and Yossi Adi. Real Time Speech Enhancement in the Waveform Domain. In *Proc. Interspeech 2020*, pages 3291–3295, 2020.

[DTDL16] Jun Du, Yanhui Tu, Li-Rong Dai, and Chin-Hui Lee. A regression approach to single-channel speech separation via high-resolution deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(8):1424–1437, 2016.

[DTX+14] Jun Du, Yanhui Tu, Yong Xu, Lirong Dai, and Chin-Hui Lee. Speech separation of a target speaker based on deep neural networks. In *2014 12th International Conference on Signal Processing (ICSP)*, pages 473–477. IEEE, 2014.

[DWO+19] Marc Delcroix, Shinji Watanabe, Tsubasa Ochiai, Keisuke Kinoshita, Shigeki Karita, Atsunori Ogawa, and Tomohiro Nakatani. End-to-end speakerbeam for single channel target speech recognition. In *Interspeech*, pages 451–455, 2019.

[DŽO+19] Marc Delcroix, Kateřina Žmolíková, Tsubasa Ochiai, Keisuke Kinoshita, Shoko Araki, and Tomohiro Nakatani. Compact network for SpeakerBeam target speaker extraction. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6965–6969. IEEE, 2019.

[DŽO+21] Marc Delcroix, Kateřina Žmolíková, Tsubasa Ochiai, Keisuke Kinoshita, and Tomohiro Nakatani. Speaker activity driven neural speech extraction. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6099–6103. IEEE, 2021.

[EHW+16] Hakan Erdogan, John R. Hershey, Shinji Watanabe, Michael I. Mandel, and Jonathan Le Roux. Improved MVDR Beamforming Using Single-Channel Mask Prediction Networks. In *Proc. Interspeech 2016*, pages 1981–1985, 2016.

[EHWLR15] Hakan Erdogan, John R Hershey, Shinji Watanabe, and Jonathan Le Roux. Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 708–712. IEEE, 2015.

[Ell96] Daniel Patrick Whittlesey Ellis. *Prediction-driven computational auditory scene analysis*. PhD thesis, Massachusetts Institute of Technology, 1996.

[FBD09] Cédric Févotte, Nancy Bertin, and Jean-Louis Durrieu. Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis. *Neural computation*, 21(3):793–830, 2009.

[FEDS07] Jonathan B Fritz, Mounya Elhilali, Stephen V David, and Shihab A Shamma. Auditory attention—focusing the searchlight on sound. *Current opinion in neurobiology*, 17(4):437–455, 2007.

[FKL+20] Y. Fan, J.W. Kang, L.T. Li, K.C. Li, H.L. Chen, S.T. Cheng, P.Y. Zhang, Z.Y. Zhou, Y.Q. Cai, and D. Wang. Cn-celeb: A challenging chinese speaker recognition dataset. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7604–7608, 2020.

[FLRH13] Cédric Févotte, Jonathan Le Roux, and John R Hershey. Non-negative dynamical system with application to speech and audio. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3158–3162. IEEE, 2013.

[GBC+21] Simone Graetzer, Jon Barker, Trevor J. Cox, Michael Akeroyd, John F. Culling, Graham Naylor, Eszter Porter, and Rhoddy Viveros Muñoz. Clarity-2021 Challenges: Machine Learning Challenges for Advancing Hearing Aid Processing. In *Proc. Interspeech 2021*, pages 686–690, 2021.

[GC08] Sharon Gannot and Israel Cohen. Adaptive beamforming and postfiltering. In *Springer handbook of speech processing*, pages 945–978. Springer, 2008.

[GCSP13] Elana Zion Golumbic, Gregory B Cogan, Charles E Schroeder, and David Poeppel. Visual input enhances selective speech envelope tracking in auditory cortex at a "cocktail party". *Journal of Neuroscience*, 33(4):1417–1426, 2013.

[GCZ+19] Rongzhi Gu, Lianwu Chen, Shi-Xiong Zhang, Jimeng Zheng, Yong Xu, Meng Yu, Dan Su, Yuexian Zou, and Dong Yu. Neural Spatial Filter: Target Speaker Speech Separation Assisted with Directional Information. In *Proc. Interspeech 2019*, pages 4290–4294, 2019.

[GSP18] Aviv Gabbay, Asaph Shamir, and Shmuel Peleg. Visual Speech Enhancement. In *Proc. Interspeech 2018*, pages 1170–1174, 2018.

[GXW+20] Meng Ge, Chenglin Xu, Longbiao Wang, Eng Siong Chng, Jianwu Dang, and Haizhou Li. SpEx+: A Complete Time Domain Speaker Extraction Network. In *Proc. Interspeech 2020*, pages 1406–1410, 2020.

[HC99] Helen M Hanson and Erika S Chuang. Glottal characteristics of male speakers: Acoustic correlates and comparison with female data. *The Journal of the Acoustical Society of America*, 106(2):1064–1077, 1999.

[HC02] John Hershey and Michael Casey. Audio-visual sound separation via hidden markov models. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2002.

[HCLRW16] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 31–35. IEEE, 2016.

[HDHU16] Jahn Heymann, Lukas Drude, and Reinhold Haeb-Umbach. Neural network based spectral mask estimation for acoustic beamforming. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 196–200. IEEE, 2016.

[HDY+12] Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[HFFHU19] Jens Heitkaemper, Thomas Fehér, Michael Freitag, and Reinhold Haeb-Umbach. A study on online source extraction in the presence of changing speaker positions. In *International Conference on Statistical Language and Speech Processing*, pages 198–209. Springer, 2019.

[HFW+20] Shota Horiguchi, Yusuke Fujita, Shinji Watanabe, Yawen Xue, and Kenji Nagamatsu. End-to-End Speaker Diarization for an Unknown Number of Speakers with Encoder-Decoder Based Attractors. In *Proc. Interspeech 2020*, pages 269–273, 2020.

[HJB+20] Jens Heitkaemper, Darius Jakobeit, Christoph Boeddeker, Lukas Drude, and Reinhold Haeb-Umbach. Demystifying tasnet: A dissecting approach. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6359–6363. IEEE, 2020.

[HKHJS14] Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis. Deep learning for monaural speech separation. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1562–1566. IEEE, 2014.

[HLZ20] Shulin He, Hao Li, and Xueliang Zhang. Speakerfilter: Deep learning-based target speaker extraction using anchor speech. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 376–380, 2020.

[Hos19] Yedid Hoshen. Towards unsupervised single-channel blind source separation using adversarial pair unmix-and-remix. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3272–3276. IEEE, 2019.

[HROK10] John R Hershey, Steven J Rennie, Peder A Olsen, and Trausti T Kristjansson. Super-human multi-talker speech recognition: A graphical modeling approach. *Computer Speech & Language*, 24(1):45–66, 2010.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HUHD+21] Reinhold Haeb-Umbach, Jahn Heymann, Lukas Drude, Shinji Watanabe, Marc Delcroix, and Tomohiro Nakatani. Far-field automatic speech recognition. *Proceedings of the IEEE*, 109(2):124–148, 2021.

[HUWN+19] Reinhold Haeb-Umbach, Shinji Watanabe, Tomohiro Nakatani, Michiel Bacchiani, Bjorn Hoffmeister, Michael L. Seltzer, Heiga Zen, and Mehrez Souden. Speech processing for digital home assistants: Combining signal processing with deep-learning techniques. *IEEE Signal Processing Magazine*, 36(6):111–124, 2019.

[HWF+20] Zili Huang, Shinji Watanabe, Yusuke Fujita, Paola García, Yiwen Shao, Daniel Povey, and Sanjeev Khudanpur. Speaker diarization with region proposal network. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6514–6518. IEEE, 2020.

[HXS+20] Yunzhe Hao, Jiaming Xu, Jing Shi, Peng Zhang, Lei Qin, and Bo Xu. A Unified Framework for Low-Latency Speaker Extraction in Cocktail Party Environments. In *Proc. Interspeech 2020*, pages 1431–1435, 2020.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[IAN] Nobutaka Ito, Shoko Araki, and Tomohiro Nakatani. Complex angular central gaussian mixture model for directional statistics in mask-based microphone array signal processing. In *2016 24th European Signal Processing Conference (EUSIPCO)*, pages 1153–1157. IEEE.

[IAN13] Nobutaka Ito, Shoko Araki, and Tomohiro Nakatani. Permutation-free convolutive blind source separation via full-band clustering based on frequency-independent source presence priors. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3238–3242, 2013.

[IRC+16] Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R. Hershey. Single-Channel Multi-Speaker Separation Using Deep Clustering. In *Proc. Interspeech 2016*, pages 545–549, 2016.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KBM+11] Martin Karafiát, Lukáš Burget, Pavel Matějka, Ondřej Glembek, and Jan Černocký. ivector-based discriminative adaptation for automatic speech recognition. In *2011 IEEE Workshop on Automatic Speech Recognition Understanding*, pages 152–157, 2011.

[KHO+06] Trausti Kristjansson, John Hershey, Peder Olsen, Steven Rennie, and Ramesh Gopinath. Super-human multi-talker speech recognition: The ibm 2006 speech separation challenge system. In *Ninth International Conference on Spoken Language Processing*, 2006.

[KS08] Chanwoo Kim and Richard M Stern. Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis. In *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[KTJ16] Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen. Speech intelligibility potential of general and specialized deep neural network based speech enhancement systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(1):153–167, 2016.

[KVv+21] Martin Karafiát, Karel Veselý, Jan Černocký, Ján Profant, Jiří Nytra, Miroslav Hlaváček, and Tomáš Pavlíček. Analysis of x-vectors for low-resource speech recognition. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6998–7002. IEEE Signal Processing Society, 2021.

[KWS+20] Qiuqiang Kong, Yuxuan Wang, Xuchen Song, Yin Cao, Wenwu Wang, and Mark D Plumbley. Source separation with weakly labelled data: An approach to computational auditory scene analysis. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 101–105. IEEE, 2020.

[KYT+17] Morten Kolbæk, Dong Yu, Zheng-Hua Tan, Jesper Jensen, Morten Kolbaek, Dong Yu, Zheng-Hua Tan, and Jesper Jensen. 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, 2017.

[LBK+20] Fernando Loizides, Sara Basson, Dimitri Kanevsky, Olga Prilepova, Sagar Savla, and Susanna Zaraysky. Breaking boundaries with live transcribe: Expanding use cases beyond standard captioning scenarios. In *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '20, New York, NY, USA, 2020. Association for Computing Machinery.

[LCH+17] Yi Luo, Zhuo Chen, John R Hershey, Jonathan Le Roux, and Nima Mesgarani. Deep clustering and conventional networks for music separation: Stronger together. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 61–65. IEEE, 2017.

[LCY20] Yi Luo, Zhuo Chen, and Takuya Yoshioka. Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 46–50. IEEE, 2020.

[LGM+21] Federico Landini, Ondřej Glembek, Pavel Matějka, A. Johan Rohdin, Lukáš Burget, Mireia Sánchez Diez, and Anna Silnova. Analysis of the BUT diarization system for Voxconverse challenge. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5819–5823. IEEE Signal Processing Society, 2021.

[Lia13] Hank Liao. Speaker adaptation of context dependent deep neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7947–7951, 2013.

[LM19] Yi Luo and Nima Mesgarani. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM transactions on audio, speech, and language processing*, 27(8):1256–1266, 2019.

[Loi07] Philipos C Loizou. *Speech enhancement: theory and practice*. CRC press, 2007.

[LPDB22] Federico Landini, Ján Profant, Mireia Diez, and Lukáš Burget. Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: theory, implementation and analysis on standard tasks. *Computer Speech & Language*, 71:101254, 2022.

[LRHW15] Jonathan Le Roux, John R Hershey, and Felix Weninger. Deep nmf for speech separation. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 66–70. IEEE, 2015.

[LRWEH19] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R Hershey. Sdr–half-baked or well done? In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 626–630. IEEE, 2019.

[LS01] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[LVRH16] Colin Lea, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks: A unified approach to action segmentation. In *European Conference on Computer Vision*, pages 47–54. Springer, 2016.

[LWD+20] Federico Landini, Shuai Wang, Mireia Diez, Lukáš Burget, Pavel Matějka, Kateřina Žmolíková, Ladislav Mošner, Anna Silnova, Oldřich Plchot, Ondřej Novotný, et al. BUT system for the second Dihard speech diarization challenge. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6529–6533. IEEE, 2020.

[LWG+20] Jinyu Li, Yu Wu, Yashesh Gaur, Chengyi Wang, Rui Zhao, and Shujie Liu. On the Comparison of Popular End-to-End Models for Large Scale Speech Recognition. In *Proc. Interspeech 2020*, pages 1–5, 2020.

[LZY19] Wenjie Li, Pengyuan Zhang, and Yonghong Yan. Tenet: target speaker extraction network with accumulated speaker embedding for automatic speech recognition. *Electronics Letters*, 55(14):816–819, 2019.

[MC12] Nima Mesgarani and Edward F Chang. Selective cortical representation of attended speaker in multi-talker speech perception. *Nature*, 485(7397):233–236, 2012.

[MCHC20] Seongkyu Mun, Soyeon Choe, Jaesung Huh, and Joon Son Chung. The sound of my voice: Speaker representation loss for target voice separation. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7289–7293, 2020.

[MKP+20] Ivan Medennikov, Maxim Korenevsky, Tatiana Prisyach, Yuri Khokhlov, Mariya Korenevskaya, Ivan Sorokin, Tatiana Timofeeva, Anton Mitrofanov, Andrei Andrusenko, Ivan Podluzhny, Aleksandr Laptev, and Aleksei Romanenko. Target-Speaker Voice Activity Detection: A Novel Approach for Multi-Speaker Diarization in a Dinner Party Scenario. In *Proc. Interspeech 2020*, pages 274–278, 2020.

[Moh97] Mehryar Mohri. Finite-state transducers in language and speech processing. *Computational linguistics*, 23(2):269–311, 1997.

[MPG+20] Pavel Matějka, Oldřich Plchot, Ondřej Glembek, Lukáš Burget, Johan Rohdin, Hossein Zeinali, Ladislav Mošner, Anna Silnova, Ondřej Novotnỳ, Mireia Diez, et al. 13 years of speaker recognition research at but, with longitudinal analysis of nist sre. *Computer Speech & Language*, 63:101035, 2020.

[MS12] Gautham J. Mysore and Maneesh Sahani. Variational inference in non-negative factorial hidden markov models for efficient audio source separation. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, page 1499–1506, Madison, WI, USA, 2012. Omnipress.

[MSCM12] Pejman Mowlaee, Rahim Saeidi, Mads Græsbøll Christensen, and Rainer Martin. Subjective and objective quality assessment of single-channel speech separation algorithms. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 69–72. IEEE, 2012.

[MSF+19] Matthew Maciejewski, Gregory Sell, Yusuke Fujita, Leibny Paola Garcia-Perera, Shinji Watanabe, and Sanjeev Khudanpur. Analysis of robustness of deep single-channel speech separation using corpora constructed from multiple domains. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 165–169. IEEE, 2019.

[MTZ+21] Daniel Michelsanti, Zheng-Hua Tan, Shi-Xiong Zhang, Yong Xu, Meng Yu, Dong Yu, and Jesper Jensen. An overview of deep-learning-based audio-visual speech enhancement and separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021.

[Mur07] Kevin P Murphy. Conjugate bayesian analysis of the gaussian distribution. *def*, 1(2σ2):16, 2007.

[MWE09] Michael I Mandel, Ron J Weiss, and Daniel PW Ellis. Model-based expectation-maximization source separation and localization. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(2):382–394, 2009.

[MWMLR20] Matthew Maciejewski, Gordon Wichern, Emmett McQuinn, and Jonathan Le Roux. Whamr!: Noisy and reverberant single-channel speech separation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 696–700. IEEE, 2020.

[NAW20] Eliya Nachmani, Yossi Adi, and Lior Wolf. Voice separation with an unknown number of multiple speakers. In *International Conference on Machine Learning*, pages 7164–7175. PMLR, 2020.

[NCZ17] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. VoxCeleb: A Large-Scale Speaker Identification Dataset. In *Proc. Interspeech 2017*, pages 2616–2620, 2017.

[NIH+17] Tomohiro Nakatani, Nobutaka Ito, Takuya Higuchi, Shoko Araki, and Keisuke Kinoshita. Integrating dnn-based and spatial clustering-based mask estimation for robust mvdr beamforming. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 286–290, 2017.

[NYN+15] Kuniaki Noda, Yuki Yamaguchi, Kazuhiro Nakadai, Hiroshi G Okuno, and Tetsuya Ogata. Audio-visual speech recognition using deep learning. *Applied Intelligence*, 42(4):722–737, 2015.

[ODK+19a] Tsubasa Ochiai, Marc Delcroix, Keisuke Kinoshita, Atsunori Ogawa, and Tomohiro Nakatani. Multimodal SpeakerBeam: Single Channel Target Speech Extraction with Audio-Visual Speaker Clues. In *Proc. Interspeech 2019*, pages 2718–2722, 2019.

[ODK+19b] Tsubasa Ochiai, Marc Delcroix, Keisuke Kinoshita, Atsunori Ogawa, and Tomohiro Nakatani. A unified framework for neural speech separation and extraction. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6975–6979. IEEE, 2019.

[OKS18] Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda. Attentive Statistics Pooling for Deep Speaker Embedding. In *Proc. Interspeech 2018*, pages 2252–2256, 2018.

[PB92] Douglas B Paul and Janet Baker. The design for the wall street journal-based csr corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*, 1992.

[PCC+20] Manuel Pariente, Samuele Cornell, Joris Cosentino, Sunit Sivasankaran, Efthymios Tzinis, Jens Heitkaemper, Michel Olvera, Fabian-Robert Stöter, Mathieu Hu, Juan M. Martín-Doñas, David Ditter, Ariel Frank, Antoine Deleforge, and Emmanuel Vincent. Asteroid: the PyTorch-based audio source separation toolkit for researchers. In *Proc. Interspeech*, 2020.

[PCPK15] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an ASR corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

[PCW+18] D. Povey, Gaofeng Cheng, Yiming Wang, K. Li, Hainan Xu, M. Yarmohammadi, and S. Khudanpur. Semi-orthogonal low-rank matrix factorization for deep neural networks. In *INTERSPEECH*, 2018.

[PGB+11] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number CONF. IEEE Signal Processing Society, 2011.

[PGM+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[Plc14] Oldřich Plchot. *Extensions to Probabilistic Linear Discriminant Analysis for Speaker Recognition*. Ph.d. thesis, Brno University of Technology, Faculty of Information Technology, 2014.

[PPG+16] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur. Purely Sequence-Trained Neural Networks for ASR Based on Lattice-Free MMI. In *Proc. Interspeech 2016*, pages 2751–2755, 2016.

[PPK15] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. A time delay neural network architecture for efficient modeling of long temporal contexts. In *Proc. Interspeech 2015*, pages 3214–3218, 2015.

[PSDV+18] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[PT94] Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

[PWLR20a] Fatemeh Pishdadian, Gordon Wichern, and Jonathan Le Roux. Finding strength in weakness: Learning to separate sounds with weak supervision. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2386–2399, 2020.

[PWLR20b] Fatemeh Pishdadian, Gordon Wichern, and Jonathan Le Roux. Learning to separate sounds from weakly labeled scenes. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 91–95. IEEE, 2020.

[QCY18] Yanmin Qian, Xuankai Chang, and Dong Yu. Single-channel multi-talker speech recognition with permutation invariant training. *Speech Communication*, 104:1–11, 2018.

[RBHH01] Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221)*, volume 2, pages 749–752. IEEE, 2001.

[RDC+21] Desh Raj, Pavel Denisov, Z. Chen, H. Erdogan, Zili Huang, Mao-Kui He, Shinji Watanabe, Jun Du, T. Yoshioka, Yi Luo, N. Kanda, Jinyu Li, S. Wisdom, and J. Hershey. Integration of speech separation, diarization, and recognition for multi-speaker meetings: System description, comparison, and analysis. In *IEEE Spoken Language Technology Workshop (SLT)*, 2021.

[RGC+20] Chandan K.A. Reddy, Vishak Gopal, Ross Cutler, Ebrahim Beyrami, Roger Cheng, Harishchandra Dubey, Sergiy Matusevych, Robert Aichner, Ashkan Aazami, Sebastian Braun, Puneet Rana, Sriram Srinivasan, and Johannes Gehrke. The INTERSPEECH 2020 Deep Noise Suppression Challenge: Datasets, Subjective Testing Framework, and Challenge Results. In *Proc. Interspeech 2020*, pages 2492–2496, 2020.

[RGC21] Chandan KA Reddy, Vishak Gopal, and Ross Cutler. Dnsmos: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6493–6497. IEEE, 2021.

[RSK+21] Neville Ryant, Prachi Singh, Venkat Krishnamohan, Rajat Varma, Kenneth Church, Christopher Cieri, Jun Du, Sriram Ganapathy, and Mark Liberman. The Third DIHARD Diarization Challenge. In *Proc. Interspeech 2021*, pages 3570–3574, 2021.

[RWL+21a] Rajeev Rikhye, Quan Wang, Qiao Liang, Yanzhang He, and Ian McGraw. Multi-user voicefilter-lite via attentive speaker embedding. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 275–282, 2021.

[RWL+21b] Rajeev Rikhye, Quan Wang, Qiao Liang, Yanzhang He, Ding Zhao, Yiteng Huang, Arun Narayanan, and Ian McGraw. Personalized Keyphrase Detection Using Speaker and Environment Information. In *Proc. Interspeech 2021*, pages 4204–4208, 2021.

[SBA10] Mehrez Souden, Jacob Benesty, and SofiÈne Affes. On optimal frequency-domain multichannel linear filtering for noise reduction. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(2):260–276, 2010.

[SBD18] Robin Scheibler, Eric Bezzam, and Ivan Dokmanić. Pyroomacoustics: A python package for audio room simulation and array processing algorithms. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 351–355, 2018.

[SCB08] Barbara G Shinn-Cunningham and Virginia Best. Selective attention in normal and impaired hearing. *Trends in amplification*, 12(4):283–299, 2008.

[Ser14] B Series. Method for the subjective assessment of intermediate quality level of audio systems. *International Telecommunication Union Radiocommunication Assembly*, 2014.

[SFM+14] Paris Smaragdis, Cedric Fevotte, Gautham J Mysore, Nasser Mohammadiha, and Matthew Hoffman. Static and dynamic source separation using nonnegative factorizations: A unified view. *IEEE Signal Processing Magazine*, 31(3):66–75, 2014.

[SGR14] Gregory Sell and Daniel Garcia-Romero. Speaker diarization with plda i-vector scoring and unsupervised calibration. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 413–417. IEEE, 2014.

[SGRS+18] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333, 2018.

[SLRH+18] Shane Settle, Jonathan Le Roux, Takaaki Hori, Shinji Watanabe, and John R Hershey. End-to-end multi-speaker speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4819–4823. IEEE, 2018.

[Sma07] Paris Smaragdis. Convolutive speech bases and their application to supervised speech separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(1):1–12, 2007.

[SMAM04] Hiroshi Sawada, Ryo Mukai, Shoko Araki, and Shoji Makino. A robust and precise method for solving the permutation problem of frequency-domain blind source separation. *IEEE transactions on speech and audio processing*, 12(5):530–538, 2004.

[Sny20] David Snyder. *X-vectors: Robust neural embeddings for speaker recognition*. PhD thesis, Johns Hopkins University, 2020.

[SR14] Pawel Swietojanski and Steve Renals. Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 171–176. IEEE, 2014.

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

[SRCG21] Cem Subakan, Mirco Ravanelli, Samuele Cornell, and François Grondin. Real-m: Towards speech separation on real mixtures. *arXiv preprint arXiv:2110.10812*, 2021.

[SSM+18] Gregory Sell, David Snyder, Alan McCree, Daniel Garcia-Romero, Jesús Villalba, Matthew Maciejewski, Vimal Manohar, Najim Dehak, Daniel Povey, Shinji Watanabe, and Sanjeev Khudanpur. Diarization is hard: Some experiences and lessons learned for the jhu team in the inaugural dihard challenge. In *Proc. Interspeech 2018*, pages 2808–2812, 2018.

[SSM+19] Igor Szöke, Miroslav Skácel, Ladislav Mošner, Jakub Paliesek, and Jan Černockỳ. Building and evaluation of a real room impulse response dataset. *IEEE Journal of Selected Topics in Signal Processing*, 13(4):863–876, 2019.

[SSNP13] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny. Speaker adaptation of neural network acoustic models using i-vectors. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 55–59. IEEE, 2013.

[SV17] Paris Smaragdis and Shrikant Venkataramani. A neural network alternative to non-negative audio models. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 86–90. IEEE, 2017.

[SWH16] Robert C Streijl, Stefan Winkler, and David S Hands. Mean opinion score (mos) revisited: methods and applications, limitations and alternatives. *Multimedia Systems*, 22(2):213–227, 2016.

[SWLRP19] Prem Seetharaman, Gordon Wichern, Jonathan Le Roux, and Bryan Pardo. Bootstrapping single-channel source separation via unsupervised spatial clustering on stereo mixtures. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 356–360. IEEE, 2019.

[SWPK20] Yiwen Shao, Yiming Wang, Daniel Povey, and Sanjeev Khudanpur. PyChain: A Fully Parallelized PyTorch Implementation of LF-MMI for End-to-End ASR. In *Proc. Interspeech 2020*, pages 561–565, 2020.

[THHJ10] Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *2010 IEEE international conference on acoustics, speech and signal processing*, pages 4214–4217. IEEE, 2010.

[TWJS21] Efthymios Tzinis, Zhepei Wang, Xilin Jiang, and Paris Smaragdis. Compute and memory efficient universal sound source separation. *Journal of Signal Processing Systems*, pages 1–15, 2021.

[VdMH08] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[VGF06] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *IEEE transactions on audio, speech, and language processing*, 14(4):1462–1469, 2006.

[VGP07] Emmanuel Vincent, Rémi Gribonval, and Mark D Plumbley. Oracle estimators for the benchmarking of source separation algorithms. *Signal Processing*, 87(8):1933–1950, 2007.

[VHU10] Dang Hai Tran Vu and Reinhold Haeb-Umbach. Blind speech separation employing directional statistics in an expectation maximization framework. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 241–244. IEEE, 2010.

[Vir06] Tuomas Virtanen. Speech recognition using factorial hidden markov models for separation in the feature space. In *Ninth International Conference on Spoken Language Processing*, 2006.

[Vir07] Tuomas Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 15(3):1066–1074, 2007.

[VWŽ+16] Karel Veselý, Shinji Watanabe, Katerina Žmolíková, Martin Karafiát, Lukáš Burget, and Jan Honza Černockỳ. Sequence summarizing neural network for speaker adaptation. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5315–5319. IEEE, 2016.

[WAF+19] Gordon Wichern, Joe Antognini, Michael Flynn, Licheng Richard Zhu, Emmett McQuinn, Dwight Crow, Ethan Manilow, and Jonathan Le Roux. WHAM!: Extending Speech Separation to Noisy Environments. In *Proc. Interspeech 2019*, pages 1368–1372, 2019.

[WCS+18] Jun Wang, Jie Chen, Dan Su, Lianwu Chen, Meng Yu, Yanmin Qian, and Dong Yu. Deep extractor network for target speaker recovery from single channel speech mixtures. In *Proc. Interspeech 2018*, pages 307–311, 2018.

[WCX+19] Peidong Wang, Zhuo Chen, Xiong Xiao, Zhong Meng, Takuya Yoshioka, Tianyan Zhou, Liang Lu, and Jinyu Li. Speech separation using speaker inventory. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 230–236, 2019.

[WDDL16] Y. Wang, J. Du, L. R. Dai, and C. H. Lee. Unsupervised single-channel speech separation via deep neural network for different gender mixtures. In *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1–4, 2016.

[WG15] Chunyang Wu and Mark JF Gales. Multi-basis adaptive neural network for rapid adaptation in speech recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4315–4319. IEEE, 2015.

[WMB+20] Shinji Watanabe, Michael Mandel, Jon Barker, Emmanuel Vincent, Ashish Arora, Xuankai Chang, Sanjeev Khudanpur, Vimal Manohar, Daniel Povey, Desh Raj, et al. Chime-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings. *arXiv preprint arXiv:2004.09249*, 2020.

[WMS+20] Quan Wang, Ignacio Lopez Moreno, Mert Saglam, Kevin Wilson, Alan Chiao, Renjie Liu, Yanzhang He, Wei Li, Jason Pelecanos, Marily Nika, and

Alexander Gruenstein. VoiceFilter-Lite: Streaming Targeted Voice Separation for On-Device Speech Recognition. In *Proc. Interspeech 2020*, pages 2677–2681, 2020.

[WMW+19] Quan Wang, Hannah Muckenhirn, Kevin Wilson, Prashant Sridhar, Zelin Wu, John R. Hershey, Rif A. Saurous, Ron J. Weiss, Ye Jia, and Ignacio Lopez Moreno. VoiceFilter: Targeted Voice Separation by Speaker-Conditioned Spectrogram Masking. In *Proc. Interspeech 2019*, pages 2728–2732, 2019.

[WRH18] Zhong-Qiu Wang, Jonathan Le Roux, and John R. Hershey. Alternative objective functions for deep clustering. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 686–690, 2018.

[WTE+20] Scott Wisdom, Efthymios Tzinis, Hakan Erdogan, Ron J. Weiss, Kevin Wilson, and John R. Hershey. Unsupervised speech separation using mixtures of mixtures. In *ICML 2020 Workshop on Self-Supervision for Audio and Speech*, 2020.

[WWL19] Dong Wang, Xiaodong Wang, and Shaohe Lv. An overview of end-to-end automatic speech recognition. *Symmetry*, 11(8):1018, 2019.

[WWPM18] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4879–4883. IEEE, 2018.

[WWW16] Donald S. Williamson, Yuxuan Wang, and DeLiang Wang. Complex ratio masking for monaural speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(3):483–492, 2016.

[WYSD15a] Chao Weng, Dong Yu, Michael L. Seltzer, and Jasha Droppo. Deep neural networks for single-channel multi-talker speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(10):1670–1679, 2015.

[WYSD15b] Chao Weng, Dong Yu, Michael L Seltzer, and Jasha Droppo. Deep neural networks for single-channel multi-talker speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(10):1670–1679, 2015.

[WZ13] Yu-Xiong Wang and Yu-Jin Zhang. Nonnegative matrix factorization: A comprehensive review. *IEEE Transactions on Knowledge and Data Engineering*, 25(6):1336–1353, 2013.

[XCY+19a] Xiong Xiao, Zhuo Chen, Takuya Yoshioka, Hakan Erdogan, Changliang Liu, Dimitrios Dimitriadis, Jasha Droppo, and Yifan Gong. Single-channel speech extraction using speaker inventory and attention network. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 86–90, 2019.

[XCY+19b] Xiong Xiao, Zhuo Chen, Takuya Yoshioka, Hakan Erdogan, Changliang Liu, Dimitrios Dimitriadis, Jasha Droppo, and Yifan Gong. Single-channel speech extraction using speaker inventory and attention network. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 86–90, 2019.

[XRCL19a] Chenglin Xu, Wei Rao, Eng Siong Chng, and Haizhou Li. Optimization of speaker extraction neural network with magnitude and temporal spectrum approximation loss. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6990–6994, 2019.

[XRCL19b] Chenglin Xu, Wei Rao, Eng Siong Chng, and Haizhou Li. Time-domain speaker extraction network. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 327–334, 2019.

[YAA+19] Takuya Yoshioka, Igor Abramovski, Cem Aksoylar, Zhuo Chen, Moshe David, Dimitrios Dimitriadis, Yifan Gong, Ilya Gurvich, Xuedong Huang, Yan Huang, Aviv Hurvitz, Li Jiang, Sharon Koubi, Eyal Krupka, Ido Leichter, Changliang Liu, Partha Parthasarathy, Alon Vinnikov, Lingfeng Wu, Xiong Xiao, Wayne Xiong, Huaming Wang, Zhenghao Wang, Jun Zhang, Yong Zhao, and Tianyan Zhou. Advances in online audio-visual meeting transcription. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 276–283, 2019.

[YCQ17] Dong Yu, Xuankai Chang, and Yanmin Qian. Recognizing Multi-Talker Speech with Permutation Invariant Training. In *Proc. Interspeech 2017*, pages 2456–2460, 2017.

[YEC+18] Takuya Yoshioka, Hakan Erdogan, Zhuo Chen, Xiong Xiao, and Fil Alleva. Recognizing overlapped speech in meetings: A multichannel separation approach using neural networks. In *Proc. Interspeech 2018*, pages 3038–3042, 2018.

[YID+15] Takuya Yoshioka, Nobutaka Ito, Marc Delcroix, Atsunori Ogawa, Keisuke Kinoshita, Masakiyo Fujimoto, Chengzhu Yu, Wojciech J Fabian, Miquel Espi, Takuya Higuchi, et al. The ntt chime-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 436–443. IEEE, 2015.

[YKTJ17] Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 241–245. IEEE, 2017.

[You96] Steve Young. A review of large-vocabulary continuous-speech. *IEEE signal processing magazine*, 13(5):45, 1996.

[YSD+12] Takuya Yoshioka, Armin Sehr, Marc Delcroix, Keisuke Kinoshita, Roland Maas, Tomohiro Nakatani, and Walter Kellermann. Making machines

understand us in reverberant rooms: Robustness against reverberation for automatic speech recognition. *IEEE Signal Processing Magazine*, 29(6):114–126, 2012.

[ZB11] Sue Anne Zollinger and Henrik Brumm. The Lombard effect. *Current Biology*, 21(16):R614–R615, 2011.

[ŽDB+21] Kateřina Žmolíková, Marc Delcroix, Lukáš Burget, Tomohiro Nakatani, and Jan Honza Černocky. Integration of variational autoencoder and spatial clustering for adaptive multi-channel neural speech separation. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 889–896. IEEE, 2021.

[ŽDK+17a] Kateřina Žmolíková, Marc Delcroix, Keisuke Kinoshita, Takuya Higuchi, Atsunori Ogawa, and Tomohiro Nakatani. Learning speaker representation for neural network based multichannel speaker extraction. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 8–15. IEEE, 2017.

[ŽDK+17b] Kateřina Žmolíková, Marc Delcroix, Keisuke Kinoshita, Takuya Higuchi, Atsunori Ogawa, and Tomohiro Nakatani. Speaker-aware neural network based beamformer for speaker extraction in speech mixtures. In *Interspeech*, pages 2655–2659, 2017.

[ŽDK+18] Kateřina Žmolíková, Marc Delcroix, Keisuke Kinoshita, Takuya Higuchi, Tomohiro Nakatani, and Jan Černockỳ. Optimization of speaker-aware multichannel speech extraction with asr criterion. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6702–6706. IEEE, 2018.

[ŽDK+19] Kateřina Žmolíková, Marc Delcroix, Keisuke Kinoshita, Tsubasa Ochiai, Tomohiro Nakatani, Lukáš Burget, and Jan Černockỳ. SpeakerBeam: Speaker aware neural network for target speaker extraction in speech mixtures. *IEEE Journal of Selected Topics in Signal Processing*, 13(4):800–814, 2019.

[ŽDR+21] Kateřina Žmolíková, Marc Delcroix, Desh Raj, Shinji Watanabe, and Jan Černocký. Auxiliary Loss Function for Target Speech Extraction and Recognition with Weak Supervision Based on Speaker Characteristics. In *Proc. Interspeech 2021*, pages 1464–1468, 2021.

[ZG21] Neil Zeghidour and David Grangier. Wavesplit: End-to-end speech separation by speaker clustering. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:2840–2849, 2021.

[ZGS20] Jianshu Zhao, Shengzhou Gao, and Takahiro Shinozaki. Time-Domain Target-Speaker Speech Separation with Waveform-Based Speaker Embedding. In *Proc. Interspeech 2020*, pages 1436–1440, 2020.

[ZHZ20] Zining Zhang, Bingsheng He, and Zhenjie Zhang. X-TaSNet: Robust and Accurate Time-Domain Speaker Extraction Network. In *Proc. Interspeech 2020*, pages 1421–1425, 2020.

[ZLD21] Cătălin Zorilă, Mohan Li, and Rama Doddipatla. An investigation into the multi-channel time domain speaker extraction network. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 793–800, 2021.

[ZSS18] Hossein Zeinali, Hossein Sameti, and Themos Stafylakis. DeepMine Speech Processing Database: Text-Dependent and Independent Speaker Verification and Speech Recognition in Persian and English . In *Proc. The Speaker and Language Recognition Workshop (Odyssey 2018)*, pages 386–392, 2018.

[ZW16] Xiao-Lei Zhang and DeLiang Wang. A deep ensemble learning method for monaural speech separation. *IEEE/ACM transactions on audio, speech, and language processing*, 24(5):967–977, 2016.

[ZWS+19] Hossein Zeinali, Shuai Wang, Anna Silnova, Pavel Matějka, and Oldřich Plchot. But system description to voxceleb speaker recognition challenge 2019. In *Proceedings of The VoxCeleb Challange Workshop 2019*, pages 1–4, 2019.