TOWARDS SINGLE-CHANNEL SPEECH SEPARATION IN NOISE AND REVERBERATION

by

Matthew Maciejewski

A dissertation submitted to The Johns Hopkins University

in conformity with the requirements for the degree of

Doctor of Philosophy

Baltimore, Maryland

October 2021

Abstract

Many speech technologies, such as automatic speech recognition and speaker identification, are conventionally designed to only work on single speech streams. As a result, these systems can suffer severely degraded performance in cases of overlapping speech, i.e. when two or more people are speaking at the same time. Speech separation systems aim to address this problem by taking a recording of a speech mixture and outputting a single recording for each speaker in the mixture, where the interfering speech has been removed. The advancements in speech technology provided by deep neural networks have extended to speech separation, resulting in the first effectively functional single-channel speech separation systems. As performance of these systems has improved, there has been a desire to extend their capabilities beyond the clean studio recordings using close-talking microphones that the technology was initially developed on. In this dissertation, we focus on the extension of these technologies to the noisy and reverberant conditions more representative of real-world applications. Contributions of this dissertation include producing and releasing new data appropriate for training and evaluation of single-channel speech separation techniques, performing benchmark experiments to establish the degradation of conventional methods in more realistic settings, theoretical analysis of the impact, and development of new techniques targeted at improving system performance in these adverse conditions.

Dissertation Committee

Sanjeev Khudanpur (Advisor)
Associate Professor
Center for Language and Speech Processing
Department of Computer Science
Department of Electrical and Computer Engineering
Johns Hopkins University

Shinji Watanabe (Co-advisor)
Associate Professor
Language Technologies Institute
Carnegie Mellon University

Adjunct Associate Professor
Center for Language and Speech Processing
Department of Electrical and Computer Engineering
Johns Hopkins University

Najim Dehak
Associate Professor
Center for Language and Speech Processing
Department of Electrical and Computer Engineering
Johns Hopkins University

Acknowledgments

I would like to thank my family for giving me everything I could need to achieve my dreams while giving me the freedom to pursue them.

I would like to thank my friends for always being there for me, even when I was too drained to offer much in return.

I would like to thank my advisors and mentors for enduring my frustrations and nevertheless putting in the effort to guide me in the ways that I needed.

I would like to thank my labmates and fellow students for your boundless help not only in getting things done but also in commiseration.

There are far too many people to thank individually who have positively touched my life through all these years of work, and for that I am immensely grateful. It is not without each and every one of you that I would be where I am today. Through the greatest trials and tribulations we reflect on all the good we have in our lives. Please know that I have nothing but the deepest appreciation for all that you have done for me. Thank you.

Dedication

John J. Godfrey

1941–2020

This work is dedicated to Jack, who had a profound impact not only on my research and education but also on me as a friend.

I first met Jack in 2014 while attending the CLSP workshop in Prague during the summer before I started graduate school. For all the wonders the Czech Republic had to offer, I was just as enamored by Jack's charisma and stories of his youth in New York City. It was easy to attach myself to him for my first project, despite his propensity for continuing to talk with me about whatever—be it the project, his career, politics, linguistics, career/life advice, football, genealogy, fishing, or whatever book he had read most recently—for quite literally hours after our meetings ended. Though I sometimes wanted to get back to work, I nevertheless always enjoyed our discussions and listening to all he had to say, and these talks eventually led to a true friendship. He

and I kept in touch, even after he finally retired properly and moved down to Louisiana to be with family. I only wish he had had more time to enjoy his retirement. I cannot imagine Jack will ever slip from my memory and can think of no better way to honor him and his impact on my life than to dedicate this work to him. I am sure that it would mean a lot to him.

Table of Contents

List of Tables

List of Figures

Abbreviations

ASR: Automatic Speech Recognition

BLSTM: Bi-directional Long Short-Term Memory

CASA: Computational Auditory Scene Analysis

ch5-2mix: CHiME-5 Mixture Dataset (Section 4.2)

CHiME: Computational Hearing in Multisoure Environments

CLSP: Center for Language and Speech Processing

DFT: Discrete Fourier Transform

DNN: Deep Neural Network

DPCL: Deep Clustering

DSER: Discounted Source-to-Error Ratio

EER: Equal Error Rate

ESSER: Estimated Source-to-Separation Error Ratio

HMM: Hidden Markov Model

IBM: Ideal Binary Mask

ICA: Independent Component Analysis

ICASSP: International Conference on Acoustics, Speech and Signal Processing

IEEE: Institute of Electrical and Electronics Engineers

IRM: Ideal Ratio Mask

ISCA: International Speech Communication Association

MERL: Mitsubish Electric Research Laboratories

MSE: Mean Squared Error

MUSAN: A Music, Speech, and Noise Corpus

mx6-2mix: Mixer 6 Mixture Dataset (Section 4.2)

NMF: Non-negative Matrix Factorization

no-2mix: Noisy Oracle Mixture Dataset (Section 6.3.1)

PESQ: Perceptual Evaluation of Speech Quality

PLDA: Probabilistic Linear Descriminant Analysis

PIT: Permutation Invariant Training

ReLU: Rectified Linear Unit

RSAN: Recurrent Selective Attention Network

SAD: Speech Activity Detection

SAR: Source-to-Artifact Ratio

SDR: Source-to-Distortion Ratio

SDRi: Source-to-Distortion Ratio improvement

SID: Speaker Identification

SIR: Source-to-Interference Ratio

SI-SDR: Scale-Invariant Source-to-Distortion Ratio (2.12)

SI-SDRi: Scale-Invariant Source-to-Distortion Ratio improvement

SI-SNR: Scale-Invariant Signal-to-Noise Ratio (typically called SI-SDR)

SNR: Signal-to-Noise Ratio

STFT: Short-Time Fourier Transform

STOI: Short-Time Objective Intelligibility

TasNet: Time-domain Audio Separation Network

TCN: Temporal Convolutional Network

uPIT: utterance-level Permutation Invariant Training

WER: Word Error Rate

WHAM!: WSJ0 Hipster Ambient Mixtures

WHAMR!: WHAM with Reverberation (Section 4.3)

WPE: Weighted Prediction Error

WSJ: Wall Street Journal Speech Corpus

wsj0-2mix: Wall Street Journal Mixture Dataset

Notation

$\hat{}$: denotes an estimate

$x(t)$, $\mathbf{x} \in \mathbb{R}^T$: mixture waveform

$s_k(t)$, $\mathbf{s}_k \in \mathbb{R}^T$: source waveform

$n(t)$, $\mathbf{n} \in \mathbb{R}^T$: noise waveform

$n_k(t)$, $\mathbf{n}_k \in \mathbb{R}^T$: noise waveform associated with k^{th} source

$n_\Sigma(t)$, $\mathbf{n}_\Sigma \in \mathbb{R}^T$: noise waveform of sum of individual source noises

$e(t)$, $\mathbf{e} \in \mathbb{R}^T$: waveform error residual

T: length of waveform in samples

t: sample index

K: total speaker/source count

k: speaker/source index

$X(d,f)$, $\mathbf{X} \in \mathbb{C}^{D \times F}$: mixture spectral feature matrix

$S_k(d,f)$, $\mathbf{S}_k \in \mathbb{C}^{D \times F}$: source spectral feature matrix

$M_k(d,f)$, $\mathbf{M}_k \in [0,1]^{D \times F}$: source spectral feature amplitude mask

$\mathbb{1}_k(d,f)$: source spectral occupancy indicator

D: number of frames in spectral domain representation

d: spectral domain frame index

F: number of spectral filters

f: spectral filter index

L: length of spectral filters in samples

P: period of spectral filter sliding window in samples

$\mathbf{w}_d \in [0,\infty)^F$: filter weight vector at frame d

$\mathbf{x}_d \in \mathbb{R}^L$: sliding window segment of the mixture waveform

$\mathbf{U} \in \mathbb{C}^{L\times F}$: matrix of analysis spectral filters

$\mathbf{V} \in \mathbb{C}^{L\times F}$: matrix of synthesis spectral filters

$H(\,\cdot\,)$: spectral rectifying function (e.g. magnitude)

E: embedding dimension

$\mathbf{Z} \in \mathbb{R}^{(DF)\times E}$: Deep Clustering embedding matrix

$\mathbf{Y} \in \mathbb{R}^{(DF)\times K}$: Deep Clustering spectral source occupancy encoding matrix

π: permutation map from estimated source index to oracle source index

$a_k(t)$: source room impulse response

C_k: source room impulse response length

λ: tuning parameter

Chapter 1

Introduction

An inevitable property of multi-party conversations is that more than one person will end up speaking simultaneously for some portion of time [1–4]. Many speech technologies, including conventional automatic speech recognition (ASR) and speaker identification (SID) systems, are not designed to function on overlapping speech and can suffer severe performance degradation under such conditions. In addition, state-of-the-art diarization systems, which label when each participant in a conversation is speaking, provide very limited handling of overlap, if at all, despite the correct labeling of overlapping speech being a component in the task and having a corresponding penalty in the evaluation metric [5–7].

Speech separation techniques aim to solve this problem by producing a separate waveform for each speaker in an audio recording with multiple talkers. Some

promising breakthroughs have been made recently in speech separation using deep neural networks (DNNs) [8–17]. These studies, however, have been limited to very controlled conditions consisting of synthetically-mixed recordings of non-overlapping speech from the Wall Street Journal (WSJ0) corpus [18], which consists of talkers reading news articles into a close-talking microphone in quiet, anechoic recording environments. Conversational speech is often recorded with a table or room microphone, resulting in far-field speech recordings often featuring reduced signal strength, increased noise, and reverberation [1, 2, 4]—a common condition that current speech separation techniques are unable to handle [19].

A common technique for mitigating performance degradation of audio processing involving distinctly localized sources among interfering signals is the use multiple microphones, referred to as "multi-channel" techniques. Audio coming from directions outside of the target signal can be suppressed with methods such as beamforming. However, it is not always possible to have access to more than one microphone, necessitating the development of "single-channel" techniques.

In this dissertation we analyze performance of state-of-the-art single-channel speech separation methods in noisy and reverberant conditions and propose methods to improve performance in those conditions, and begin to close the gap in performance between near- and far-field speech.

1.1 Contributions

The primary goal of this dissertation is to *develop novel techniques to improve speech separation in noisy and revereberant recording conditions*.

One contribution of this work is to establish a formulation of speech separation in noisy and reverberant conditions and discuss theoretical reasons as for why speech separation is particularly challenging in such conditions. In order to perform experimental analysis of the conventional speech separation systems, this work creates and publicly releases a number of datasets representing various conditions and establishes benchmarks in those conditions, resulting in multiple publications [19, 20]. These datasets aid reproducibility in the field, and experimental recipes for [20] have been included in the widely-used Asteroid [21] and ESPnet [22] toolkits. This work includes simple techniques used to improve performance of the systems in those conditions. Another contribution is an analysis of two different paradigms for creating data for training and evaluation data for speech separation systems in noisy conditions, and experimental evaluation of the impact of the paradigms on both training and evaluation. This dissertation also proposes a solution to improve performance of systems trained in the paradigm which allows for training with in-domain recordings [23]. A final contribution of this work is an exploration into the use of speaker verification as downstream evaluation of speech separation system performance [24].

1.2 Dissertation Organization

This introductory chapter serves as a brief overview of the dissertation and its position in the field. The following two chapters serve as a survey of the task, including discussion of the problems, context, theoretical formulations, and related work. Chapter 2 focuses on the overall single-channel speech separation problem, while Chapter 3 focuses on the extension to conditions with noise and reverberation. Chapter 4 is focused on experimental documentation of the impact of noise and reverberation on speech separation systems, including creation of datasets necessary for such evaluations. Chapter 5 focuses on basic approaches used to improve performance in these conditions. Chapters 6 and 7 focus on analysis and a proposed solution, respectively, regarding an issue with the creation of training and evaluation data for noisy speech separation. Chapter 8 explores speaker verification as a downstream task for evaluation of speech separation systems, in part to address issues regarding noise and reverberation in ground truth. Finally, Chapter 9 concludes the dissertation with a summary and discussion of future work.

Chapter 2

Single-Channel Speech Separation

2.1 Overview

Single-channel speech separation refers specifically to the task of estimating multiple output waveforms from a single input recording in which multiple people speak simultaneously, with each estimate containing the speech of only one of the speakers in the input recording. In contrast to multi-channel techniques, where multiple microphones capture the speech and give access to directional information, single-channel speech separation must use only the structure of speech and must leverage inter-speaker differences, relying heavily on the fact that the speech of each speaker is sparse in a time-frequency domain. In other words, if a mixture of multiple speakers is segmented spectrally, for example with a simple Short-Time Fourier Transform

(STFT) with reasonable parameters, there is low probability that multiple speakers will contribute significant energy to any particular time-frequency bin. This not only makes it easier to partition the signal in a spectral representation, but also makes the latent speech signals more easy to identify in training and inference. Prior to the proliferation of DNN-based methods fueled by large amounts of labeled data, source separation techniques were typically based on either known properties of the speech signals or inspired by the human auditory perceptual system's ability to track sources in overlapped speech.

Examples of such conventional methods are Computational Auditory Scene Analysis (CASA) [25], Factorial Hidden Markov Models (HMMs), Independent Component Analysis (ICA) [26], and Non-negative Matrix Factorization (NMF) [27]. These methods typically are founded in signal processing and rely on statistical properties of the signals to separate the sources. The biggest challenge with this class of techniques for speech separation, compared to other separation tasks such as removing noise from speech, is that speech signals from two different speakers can have very similar statistical properties. The approaches do utilize the structure and continuity constraints of speech in time and frequency, which leads to some level of success in speech separation, but their performance is largely surpassed by the newer deep learning techniques.

While some state of the art techniques do estimate the speech source waveforms

directly, the majority of DNN speech separation techniques rely on a spectral masking approach. These techniques are based on first projecting the mixture waveform using an analysis transform into a two-dimensional spectral domain with resolution in both time and frequency. In earlier techniques, the spectral representation used was the Short-Time Fourier Transform (STFT), but more recently learned transforms are used. Next, a neural network takes this mixture spectral representation and produces a mask for each speaker, with values ranging from 0 to 1. Each of these masks are then independently multiplied with the mixture representation to mask out the interfering sources, resulting in an estimate of the source spectra of individual speakers. Finally, a synthesis transform is used to convert the spectral representations back into estimated source waveforms.

The differences between the various techniques fall into one of three main categories: differences in the spectral feature transformation, differences in the type or topology of the neural network used to produce masks, and differences in the loss functions used in training the network.

2.2 Related Work

The advent of deep learning-based single-channel speech separation effectively began with two foundational works, both based on a Short-Time Fourier Transform (STFT) spectral masking approach, which have largely defined the field ever since. One

of these techniques is Deep Clustering (DPCL) [8, 9]. This approach is based on attempting to partition each STFT bin of the mixture spectrum according to the speaker whose energy dominates that bin. The partitioning is accomplished by first embedding each STFT bin into a high-dimensional space in which the bins cluster according to speaker. In addition, this work released the wsj0-2mix and wsj0-3mix datasets, artificial mixtures created by summing together recordings from the Wall Street Journal speech corpus [18]. This dataset quickly became the standard for single-channel speech separation and has been used in nearly every study since. The other foundational technique is Permutation Invariant Training (PIT) [10, 11], which solves the problem of the unknowable order of sources from which to backpropagate the loss, by evaluating the loss for every permutation, and only backpropagating from the permutation that produces the smallest loss. This technique is foundational to the field and, after the field moved away from the STFT, is used in nearly every system. It has even been used in non-separation multi-speaker tasks such as multi-speaker speech recognition [28, 29] and speaker diarization [30].

Another seminal work in single-channel speech separation was the development of TasNet [15, 16]. The development of TasNet was to replace the STFT with sliding-window projections of the signal onto a set of learned analysis and synthesis bases to serve as a comparable spectral transform to the STFT. This approach provided significant gains, generally attributed to the capabilities of the network to model

phase information directly and optimize the network according to a waveform-level objective.

This waveform-level objective, Scale-Invariant Signal-to-Distortion Ratio (SI-SDR), best described and analyzed in [31], has also served as a significant work in the field. SI-SDR currently serves as the most common evaluation metric and is used as an objective function in most systems, including in techniques not based on the TasNet learned spectral masking paradigm [32]. SI-SDR plays a central role in this dissertation and is described further in Section 2.4.3.

2.3 Problem Formulation

The basic formulation of the speech separation problem is

$$x(t) = \sum_{k=1}^{K} s_k(t), \text{ for } t = 1, \ldots, T, \tag{2.1}$$

where $x(t)$ represents the mixture waveform, and consists of a basic sum of each of K speech signals $s_k(t)$, each having originated from a different speaker. The speech separation task is then to produce estimates $\hat{s}_k(t)$ of the speech signals for each of the K speakers.

The basic spectral masking formulation begins with a sliding window segmentation

of the original mixture into D vectors $\mathbf{x}_* \in \mathbb{R}^{1\times L}$ of length L over a stride of length P:

$$\mathbf{x}_d = x(t), \text{ for } t \in [dP, dP+L). \tag{2.2}$$

Here d represents the index of the segments. These are then projected onto a set $\mathbf{U}$ of F basis vectors of length L, which is then passed through a function with non-negative range, producing a vector $\mathbf{w}_d \in [0, \text{inf})^{1\times F}$, where F is the number of analysis basis vectors. This can be reformulated in matrix notation as:

$$\mathbf{w}_d = H(\mathbf{x}_d\mathbf{U}), \tag{2.3}$$

where $\mathbf{U} \in \mathbb{C}^{L\times F}$ is the set of F vectors of length L, with the resulting vector $\mathbf{w}$ being the corresponding weights when the signal is projected onto these basis vectors. The function H ensures the weights are non-negative.

These weight vectors can be concatenated together to form a two-dimensional matrix $\mathbf{X} \in [0, \text{inf})^{D\times F}$, a time series of features with each feature coefficient representing occupancy of some particular frequency content. For example, when using the Short-Time Fourier Transform (STFT), $\mathbf{U}$ is the Fourier basis, $H(\,\cdot\,)$ is the magnitude operation, and $\mathbf{X}$ represents the magnitude spectrum of the mixture.

The problem is then formulated as a matrix estimation problem. The goal is to, from the mixture spectrum $\mathbf{X}$, estimate a soft binary mask $\hat{\mathbf{M}}_k \in [0,1]^{D\times F}$ for each of

the K speech sources, with values close to 1 in bins where the mixture representation is dominated by the target speaker and values close to 0 in bins where the representation is dominated by other speakers:

$$\left(\hat{\mathbf{M}}_1, \ldots, \hat{\mathbf{M}}_K\right) = \text{DNN}(\mathbf{X}). \tag{2.4}$$

With element-wise multiplication of each mask with the mixture spectrum, the estimates of the corresponding speech source spectra are:

$$\hat{\mathbf{S}}_k = \hat{\mathbf{M}}_k \circ \mathbf{X}, \text{ for } k = 1, \ldots, K. \tag{2.5}$$

For reconstruction of the signals, the new estimated source weights are multiplied onto a matched set of F synthesis basis vectors $\mathbf{V} \in \mathbb{C}^{L \times F}$ and summed after applying the corresponding strides to produce estimates of the source waveforms $\hat{s}_k(t)$. It is worth noting, however, that in the particular case of the STFT, although the mask estimate is generated using the magnitude spectrum, the mask is multiplied onto the complex spectrum for synthesis. This is done to simplify the task, as although synthesizing the audio using the original phase of the mixture is not correct, initial attempts to learn complex masks were less successful than amplitude masks (in part due to phase information not having the same sparsity assumptions as magnitude does), and using the original phase is likely better than synthesizing with random or constant phase.

More recent works involving phase estimation have been successful [33, 34], though were largely set aside in favor of better-performing methods that model waveforms directly.

2.4 Experimental Configuration

2.4.1 Models

The overwhelming majority of single-channel speech separation models presented in the literature have two primary points of variation: the feature transforms and the network that generates the mask from the spectral representation. All models proposed in this dissertation fall under this architectural paradigm.

The two types of feature transforms used are the short-time Fourier transform and learned features. In the case where the STFT is used for the features, the spectral transforms $\mathbf{U}$ and $\mathbf{V}$ described in Section 2.3 are simply the windowed complex Discrete Fourier Transform (DFT) bases. The function $H(\cdot)$ is the magnitude operator. One of the biggest benefits of using the STFT is that since it is a fixed transform, the target speech signals (i.e. ground truth single-speaker waveforms $s_k(t)$) can be projected into this domain as well, and a spectral-based loss function can be used. The primary downside of using the STFT is that in practice, the network estimates spectral amplitude masks, relying on using the original mixture phase for reconstruction, which

ultimately produces phase errors.

The newer type of feature transforms used are the "TasNet" [15] learned basis vectors. For this method, both the **U** and **V** transform matrices are fully learned. As they are real-valued, the function $H(\cdot)$ is just a simple rectifying function, typically a Rectified Linear Unit (ReLU). One of the primary motivations for the TasNet bases is that they can directly use a waveform-level objective and learn to model phase information accordingly. A downside, however, is that since the transform is learned and has its parameters updated throughout training, the internal spectral representation is unknown *a priori* and the ground truth target cannot be computed in spectral form like with the STFT. And, the loss function must in part be at the waveform level so as to result in proper learning of the synthesis bases.

The two types of time series-modeling mask-estimation networks we used are Bi-directional Long Short-Term Memory (BLSTM) networks and Temporal Convolutional Networks (TCN). These serve as the DNN featured in equation (2.4), with each featuring a sigmoid at the end to ensure production of masks with values between 0 and 1. Both BLSTMs and TCNs are standard architectures that are used in modeling time series of features such as our mixture spectrum. BLSTMs have a higher memory overhead and are slower to train, but have hypothetically arbitrary memory along the time axis. In contrast, while TCNs are faster and use less memory, they have a fixed context in the time dimension.

All networks used in our experiments effectively fall into a combination of the above feature transforms and internal masking architectures. The Deep Clustering [8], Permutation Invariant Training [11], and Recurrent Selective Attention Network [14] systems are all based on STFT features combined with a BLSTM. The TasNet-BLSTM [15] and ConvTasNet [16] systems use TasNet learned features with a BLSTM and TCN respectively.

2.4.2 Training

2.4.2.1 Loss Functions

The STFT-based loss functions are based on a consequence of the sparsity of speech in the spectral domain, namely:

$$\|X(d,f)\| \approx \max_k \|S_k(d,f)\|, \tag{2.6}$$

where d and f are the time and filter indices for each spectral bin respectively. The goal is then to estimate a set of binary indicator functions $\mathbb{1}_{1:K}$, that map from the mixture amplitude spectrum to the occupancy of each time-frequency bin by a given source. This can be used as a multiplicative mask to produce an estimate $\hat{S}_k(d,f)$ of

the individual source spectrum $S_k(d,f)$ from the mixture spectrogram:

$$\hat{S}_k(d,f) = \mathbb{1}_k(d,f)X(d,f). \tag{2.7}$$

For Deep Clustering (DPCL) the loss is based around *affinity matrices*, $DF \times DF$ binary-valued matrices that indicate *pairs of STFT coefficients* that are dominated by the same source. The loss function used is based on the squared Frobenius norm between the "oracle" and estimated affinity matrices:

$$\text{Loss}^{\text{DPCL}}(\mathbf{V}) = \|\mathbf{Z}\mathbf{Z}^\top - \mathbf{Y}\mathbf{Y}^\top\|_F^2. \tag{2.8}$$

The affinity matrix can be generated using the outer product of an indicator matrix $\mathbf{Y} \in \{0,1\}^{(DF)\times K}$ with itself, where the k-th column of $\mathbf{Y}$ is a DF-dimensional binary vector encoding the STFT coefficients belonging to source k. The matrix $\mathbf{Z} \in \mathbb{R}^{(DF)\times E}$ is produced by the DPCL network, consisting of a E-dimensional embedding vector for each STFT coefficient, which produces an affinity matrix estimate through the self-outer product.

The training target for the mask-based losses are approximations of $\hat{S}_k(d,f)$ tailored for performance, such as the Ideal Binary Mask (IBM) and Ideal Ratio Mask (IRM) among others [35], and are defined as follows:

$$M_k^{\text{IBM}}(d,f) = \begin{cases} 1, & \arg\max_{k'} |S_{k'}(d,f)| = k \ ; \\ 0, & \text{otherwise} \end{cases} \tag{2.9}$$

$$M_k^{\text{IRM}}(d,f) = \frac{|S_k(d,f)|}{\sum_{k'=1}^{K} |S_{k'}(d,f)|} \ . \tag{2.10}$$

These mask targets are then used in the loss function.

These approaches typically train using the mean squared error loss between the estimated (masked) spectra of perceived sources and the ground truth source magnitude spectra, including a key permutation step to match estimated and oracle masks.

$$\text{Loss}^{\text{uPIT}}(\hat{\mathbf{M}}, \pi) = \frac{1}{DFK} \sum_{k=1}^{K} \| \hat{\mathbf{M}}_k \circ |\mathbf{X}| - |\mathbf{S}_{\pi_s}| \|_F^2, \tag{2.11}$$

where $\hat{\mathbf{M}}_k \in [0,1]^{D \times F}$ is the estimated mask for source s, and $|\mathbf{X}|, |\mathbf{S}_s| \in \mathbb{R}_{\geq 0}^{D \times F}$ are the Short-Time Fourier Fransform (STFT) magnitudes for the mixture and source k, respectively. The summation over K represents the different sources in a mixture. D and F denote the numbers of frames and frequency bins, respectively. π is the permuted source sequence of oracle magnitude spectra, chosen to match the sequence of estimated masks, where π_k returns the k-th element of π, i.e. the ground truth source index matching the k-th estimated mask. This is typically chosen for each set of outputs $\hat{\mathbf{M}}_1, \ldots, \hat{\mathbf{M}}_K$ by using the permutation that produces the lowest loss for each training pass.

For waveform-level objectives, the typical loss used is negative SI-SDR. SI-SDR serves as the current typical evaluation metric, computing the log power ratio of the source to the error between the source and estimate. This is then negated to serve as a loss function. Further details on SI-SDR will be presented in Section 2.4.3, in the discussion of evaluation metrics. This loss function is applied using the same permutation-invariant manner described above.

2.4.2.2 Data Requirements and Design

One of the defining aspects of the data used in DNN-based single-channel speech separation is the fact that it generally must use artificially-created mixtures. This is relevant not only for training of systems but also the conventional methods of evaluation. The crux of the issue is that in order to provide training targets for separating mixtures, and to evaluate the closeness of estimates to the desired signal, the desired single-speaker speech signal must be known. Being able to produce the correct answers from a natural mixture of speech would itself be to have perfectly solved the speech separation task. As a result, the data solution is to take single-speaker recordings and artificially sum them. Audio waves from independent sources combine to first approximation in an additive manner, so this is a sufficiently appropriate simulation of real mixtures, but has the benefit of the DNN training process—and evaluation dataset—having perfect ground truth information.

In terms of the specifics of dataset creation, there are a few considerations that must be made. Two of the less impactful ones are the relative signal amplitude and sample rate. It is typical to control the signal-to-noise ratio between the speech signals to mimic the way in which the relative volumes of multiple speakers will differ. It is worth noting, however, that the conventionally-used datasets do not account for variable relative volumes over time. The sample rates used for experiments have traditionally been 8 kHz to reduce processing power, but more recently 16 kHz data has become more common to match typical speech technology conditions.

One of the most important decisions in creating data for speech separation is the style of overlap. The question is what regions of the trial recording contain each speaker and how much of the recording is overlapped. There are three primary ways in which data is created: The first is to combine two utterances and truncate the longer to the length of the shorter, typically referred to as the *min* condition, where the mixture recording contains 100% overlap. The second involves the same combination, but without truncation, resulting in a region at the end of the mixture recording where only one speaker's speech is present, referred to as the *max* condition. The third is the "conversational" case, where recordings are staggered, only featuring overlap at the beginnings or ends of utterances, sometimes including features such as one utterance entirely within another and regions of complete silence. Systems ability to perform in these conditions depend on the style of overlap in the training

data. However, while the "conversational" case is the most representative of many applications, it is not necessarily the most desirable condition, as it could be possible to handle varying overlap conditions with a different component, e.g. to have an overlap detection system identify regions where only two speakers are present, and only run the separation system on those regions accordingly.

For our experiments, we focus on the use of the wsj0-2mix dataset [8] as well as datasets that are designed to be similar in composition. This data consists of mixtures combined at SNRs up to 5 dB in both *min* and *max* configurations at both 8 kHz and 16 kHz sample rates.

2.4.3 Evaluation

Methods of evaluation can be categorized as either intrinsic or extrinsic. Intrinsic evaluation of speech separation is most common, involving methods and metrics designed to directly capture the quality of the speech separation. In contrast, extrinsic evaluation involves trying to capture separation performance through performance evaluation on a *different* task of a system that includes the speech separation system of interest as a component in the pipeline.

2.4.3.1 Intrinsic Evaluation

The evaluation metric in source separation that was initially used as the standard is Source-to-Distortion Ratio (SDR) [36], which measures the energy ratio (in decibels) of the target source $s_k(t)$ to the interferences, noise, and artifacts that contribute to the reconstruction error $s_k(t) - \hat{s}_k(t)$. SDR improvement (SDRi) is typically reported, reflecting the improvement in SDR from the system over the unprocessed corpus. Sometimes companion metrics are reported along with SDR, which reflect other aspects of performance compared to the relatively catch-all SDR. They are: Source-to-Interferences Ratio (SIR), Source-to-Artifacts Ratio (SAR), and Source-to-Noise Ratio (SNR) [36]. Each of these metrics is an energy ratio of the target source to some kind of error, computed based on filtering and decomposition of the residual from the estimate to the ground truth source.

In recent literature, use of SDR has been replaced with Scale-Invariant Signal-to-Distortion Ratio (SI-SDR) [31]. This became popular in part due to its use as a waveform-level training objective for end-to-end networks [15, 16], but also due to addressing downsides of the original SDR metric. This metric omits any processing of the source-estimate residual, providing a simple energy ratio of the source to the waveform-level error, with the addition of a scaling term:

$$\text{SI-SDR}(\hat{\mathbf{s}}) \coloneqq 10\log_{10} \frac{\|\mathbf{s}\|^2}{\|\mathbf{s} - \beta\hat{\mathbf{s}}\|^2}, \text{ for } \beta \text{ s.t. } \mathbf{s} \perp \mathbf{s} - \beta\hat{\mathbf{s}}, \tag{2.12}$$

where $\mathbf{s}$ is the target source waveform and $\hat{\mathbf{s}}$ is its estimate. This scaling term is chosen based on the assumption that the error is orthogonal to the source waveform. Since arbitrary linear scaling does not affect the theoretical correctness of the waveform, this solves the problem that changing the dynamic range of the estimate would increase or decrease the SDR metric without any meaningful change in the estimated signals.

Other evaluation metrics used in speech separation are Short-Time Objective Intelligibility (STOI) [37] and Perceptual Evaluation of Speech Quality (PESQ) [38]. These metrics are not designed for speech separation, however, and are more relevant to speech enhancement. As such, they are rarely used in speech separation research.

A notable downside to the standard speech separation evaluation metrics is that they require ground truth, which restricts evaluation to synthetically-generated overlapping speech instead of naturally-occurring speech. Due to the error being computed at a sample-by-sample basis, there is no possibility to collect a target-speaker signal, as even slight differences in the source-to-mic distance significantly impacts the waveform at the sample level. In addition, it also raises questions as to what the oracle speech signal should be considered to be in the case of noise and reverberation being present. Further commentary on this issue is discussed in Chapters 6 and 8.

Although rarely reported, subjective human listening tests are possible and a potential solution to evaluation on non-synthetic data, though it is unclear that subjective evaluation provides value over application-driven evaluation to an extent that justifies

the logistical costs.

2.4.3.2 Extrinsic Evaluation

The use of downstream tasks for evaluation is often primarily used to evaluate the performance of those systems specifically for applications where the downstream task is the ultimate goal. This can be important because the metrics used in intrinsic evaluation are not guaranteed to correlate with the impact of the system of a downstream task. For example, if the ultimate goal is to use speech separation as pre-processing for a speech technology designed for non-overlapping speech, the separation system that works best when coupled with the downstream system may not be the separation system with the highest SI-SDR value.

Nevertheless, there can be other reasons to use extrinsic evaluation. Extrinsic evaluation can sidestep issues relating to the direct evaluation metrics such as difficulty of computation or undesirable qualities of the metric. However, there are many downsides to extrinsic evaluation as well. Further discussion of issues specific to typical evaluation of speech separation are presented in Chapter 8.

The typical method used for downstream evaluation of speech separation is automatic speech recognition, with the standard metric being Word Error Rate (WER). This type of evaluation is not always possible, however, as it requires transcriptions of the overlapping speech. In fact, in many speech separation evaluation conditions,

the speech does not include full utterances that are even capable of transcription. And finally, even when technically possible, transcription can be a resource-intensive method of annotation.

Chapter 3

Separation of Noisy and Reverberant Speech

3.1 Overview

From the most simplistic standpoint, the addition of noise makes the task of separating two speech signals more difficult simply due to the strength of each speech signal decreasing relative to the strength of the interfering signals. At another level, the addition of noise requires the network to perform an additional separation task. Without noise, and particularly in the typical case of only two speakers, the network is essentially being asked to partition the spectro-temporal features it is given—to simply assign each segment of the signal to one of two categories of similar composition.

With noise, the network no longer can simply decide which of two categories the feature is like, but also must consider if it belongs to neither. It is also worth noting that the exact separation task in noisy speech separation is not very well defined: it is not clear if the noise in the recording must be removed from the speech or not. In addition, due to the speech separation task relying more on structure than statistical properties compared to other source separation tasks, if the noise has aspects that resemble speech, such as babble noise, it can confuse the network.

Reverberation can prove to be an even more challenging problem, particularly in a spectral Short-Time Fourier Transform (STFT) or STFT-like domain. Due to the relatively long length of typical room impulse responses compared to STFT windows [39], there is smearing in both time and frequency. This can make mask-based methods challenging, as the sparsity assumption is less valid and will make signals harder to identify and require more soft decisions.

3.2 Related Work

There have been some efforts to extend speech separation into conditions beyond the clean, near-field conditions represented in the foundational wsj0-2mix dataset [8]. While the efforts have been recent, the WHAM! [40] and LibriMix [41] datasets, which feature noise sources, have been released to aid the development of speech separation approaches for these conditions. There has been limited work beyond their

use in training of conventional systems, however.

In addition, the MixIT [42] work is of note. The method they propose aims to use unsupervised training data, allowing the training of networks on in-domain data and accordingly the more diverse conditions that would inlude real noise and reverberation.

3.2.1 Speech Enhancement

Closely related to the work of this dissertation is the topic of speech enhancement, which we use to refer to the tasks of both speech denoising and speech dereverberation. Additionally, while multichannel approaches are very popular and have been quite successful at speech enhancement, we focus on the subset of single-channel techniques, as in this work we are focused on the single-microphone case.

Single-channel speech denoising has followed a similar path to separation. Early denoising techniques include statistical methods [43] and Computational Auditory Scene Analysis (CASA) [25]. More recently, DNN-based approaches have dominated due to their high level of performance. Similar to separation, these enhancement systems have largely relied on spectral masking approaches [44], but have also included time-domain approaches as well [45, 46]. Additionally, these systems are similar to DNN-based speech separation techniques in that they typically rely on matched noisy-clean sample pairs in training, necessitating the noise to be added synthetically to clean speech.

In contrast, speech dereverberation has generally been handled somewhat differently from denoising and separation, largely due to the fact that reverberation is convolutive, in contrast to noise and speech being additive. Unsupervised approaches have been quite succesul, generally relying on the exponentially decaying property of reverberation, computing a Weiner-like filter from estimates of the late-reverberation power spectral density [47]. One of the most popular dereverberation methods used today, using the weighted prediction error (WPE) algorithm [48], is one such unsupervised method. Nevertheless, DNN-based supervised techniques involving spectral masking similar to the speech separation and speech denoising approaches have been shown to be successful as well [49, 50]. Again, these approaches rely on artificially-reverberated speech.

It is also important to note that there is reason to believe speech dereverberation is among the speech technologies that could be negatively impacted by overlapping speech. Dereverberation techniques rely on the estimation or modeling of some properties of the impulse response of the reverberated speech. Multiple sources will have different impulse responses, and so it is possible that some dereverberation systems will suffer performance degradation in situations where multiple sources with differing impulse responses are overlapping in a recording.

3.3 Formulation

The effects of noise and reverberation on the speech signals are additive and convolutive respectively. Though a noise signal can consist of any number of point sources or diffuse signals, the former of which can themselves include reverberation, we collapse all signals that are not the target speakers into the noise signal $n(t)$, of whose properties we make no specific assumptions. In contrast, we do make a distinction in the reverberation of the multiple target speech sources. Though they almost assuredly have comparable reverberation times due to primarily being a function of the room, the early reflections are subject to the local geometry which is not shared between speakers. This results in a different impulse response $a_k(t)$ of length C_k for each speaker. The formulation for the problem is as follows:

$$x(t) = \sum_{k=1}^{K} s_k(t) + n(t) \qquad \text{noise (3.1)}$$

$$x(t) = \sum_{k=1}^{K} \sum_{\tau=0}^{C_k} a_k(\tau) s_k(t-\tau) \qquad \text{reverberation (3.2)}$$

$$x(t) = \sum_{k=1}^{K} \sum_{\tau=0}^{C_k} a_k(\tau) s_k(t-\tau) + n(t) \qquad \text{combination (3.3)}$$

Here $x(t)$ is the time-domain mixture, the $s_k(t)$ are the varying source speech signals which we are trying to recover or estimate, convolved with the impulse response $a_k(t)$ in cases of reverberation, and $n(t)$ is interfering noise.

Though the ultimate and most successful outcome of a speech separation system operating under the situation of (3.3) would be to recover the original speech signals $s_k(t)$, the speech separation task in this domain is not necessarily clear, and other notions of success are possible depending on the specific task or application. Namely, recovering $\sum_{\tau=0}^{C_k} a_k(\tau)s_k(t-\tau) + n_k(t)$, i.e. the reverberant source with some kind of noise present, is a potentially successful outcome, as though the signal is both noisy and reverberant, it contains no interfering speech signals. It could then be reformulated as classical speech enhancement, assuming a clean speech signal is needed at all. Another possible solution would be somewhere in between, where there is only some residual noise or reverberation. Indeed, it is reasonable to consider potential solutions to be any signal that contains the entire target speaker signal, none of the interfering speaker signals, and no signals that were not present in the original mixture. However, this type of solution is difficult to evaluate. Additional discussion regarding the target signals and their evaluation in the presence of noise is presented in Chapters 6, 7, and 8.

The presence of noise in the mixture is the most easy to extend to from the clean separation formulation presented in Chapter 2, as it is simply an additional additive source alongside the speech. All of the formulations and techniques can still apply if the noise is treated as simply another speech source that is never evaluated. The relevant difference is in the *structure* of the noise. Speech is heavily structured in

a spectral domain, with the most relevant properties for separation being that it is sparse–with energy concentrated in very localized regions of the spectrum—and that those energy regions largely follow locally contiguous trajectories [43]. In contrast, we can make no such assumption about an arbitrary noise signal, and as such separation techniques that are designed to exploit the structure of speech may not be as successful when one of the interfering sources is noise.

The addition of reverberation complicates the formulation in a much more significant way. The presence of reverberation does *not* manifest as an additional source in the mixture, but rather more as a modification to the sources themselves. In time domain, reverberation can be approximated as a superposition of many delayed and attenuated versions of the sources. A generic closed-form formulation of the effect of reverberation in spectral domain is not generally possible; however, it generally results in smearing in both time and frequency [43]. The likely most harmful result from reverberation on speech separation is that this spectral smearing degrades both the sparsity and clear structure of speech in the spectral domain.

Chapter 4

Experimental Analysis

4.1 Overview

A critical component of working on a new problem is to establish that the problem itself exists, i.e. that conventional methods do not solve it, and to produce strong baselines against which to measure performance of new approaches. This involves well-designed experiments with proper data for the problem at hand, use of established systems with good performance on previously established benchmark tasks, and appropriate evaluation metrics for the new problem.

One of the core aspects of working on single-channel speech separation in noise and reverberation is that it is a problem that has only recently been addressed in a deep learning framework. Accordingly, standard datasets with wide-spread use do

not exist. The data used in training and evaluation must be prepared, and must be prepared with care, to ensure that any conclusions made may be done so fairly and in a way appropriate for target applications. It is also important that the data be made available, so that it is possible for future work by a wide variety of researchers to be conducted in a way that fair comparisons can be made between systems.

In addition, to survey the performance of standard state-of-the-art systems, a balance must be struck in choosing systems that can be representative of the various existing approaches that results in a fair characterization of the state-of-the-art, while under reasonable constraints of time and computational power.

4.2 Construction of Synthetic Mixtures of Real Conditions

Our first contribution involves the creation of new datasets in a similar manner to the wsj0-2mix dataset [8], but using data sourced from other corpora which represent a wider set of acoustic conditions that include noise and reverberation, with the goal of analyzing the effects of such interferences on speech separation systems. To this end, this dissertation first establishes the process of creating multi-domain datasets which allow clean/noisy and near/far-field comparisons. Since evaluation metrics and model training require ground truth single-speaker speech, we created a procedure to isolate

high-quality single-speaker speech regions from the real multi-talker corpora.

This section describes the method we used for the construction of multi-domain datasets of two-speaker mixtures from new source corpora that are effective for quantitative analysis of speech separation techniques: (1) selecting corpora consisting of differing difficulties for speech separation, (2) extracting single-speaker segments from noisy corpora, and (3) generating mixture lists which match the pre-existing wsj0-2mix dataset as closely as possible for fairer comparisons. The code and resulting mixture lists used in our experiments have been released to the research community for reproducibility and use in further studies[1,2].

4.2.1 Corpus Selection

To assess varying difficulties for speech separation, we selected the CHiME-5 [3] and Mixer 6 [51] speech corpora to complement the WSJ0 [18] corpus and its pre-existing synthetic overlap wsj0-2mix dataset [8], a standard of speech separation evaluation. This dataset has been effectively used in a number of speech separation research experiments, and so its composition was the model for our dataset generation pipeline.

The CHiME-5 corpus was chosen to serve as the most challenging, "realistic" condition. The corpus consists of dinner parties recorded with microphone arrays placed

[1] https://github.com/mmaciej2/kaldi/tree/chime5-single-speaker-generation/egs/chime5/single_speaker_generation

[2] https://github.com/mmaciej2/kaldi/tree/mixer6-single-speaker-generation/egs/mixer6/single_speaker_generation

around an apartment as well as binaural microphones worn by the speakers, allowing us to generate parallel near-field and far-field datasets with identical utterances. This condition resulted in a number of unique challenges in the audio, such as naturally occurring non-speech noises, multiple simultaneous speakers, and time-varying locations. The high amount of noise, variable loudness of speech, and time-varying room impulses response all contribute to a very challenging speech corpus.

The Mixer 6 corpus was chosen to serve as a middle ground between the WSJ0 and CHiME-5 corpora. Including interviews recorded with 14 microphones in a constructed recording room, the Mixer 6 corpus allows a similar near- and far-field comparison, but in a more controlled environment with stationary speakers, consistent channel, and relatively minimal noise.

4.2.2 Cleanup Methods

To ensure the source data is single-speaker, we used a pipeline implemented with the Kaldi Speech Recognition Toolkit [52] following two stages:

Stage 1) Run a speech activity detection (SAD) system to produce reasonable utterances. The SAD system used is a Time-Delay Neural Network-based system with statistics pooling trained as described in [53] with reverberated LibriSpeech [54] data and added noise from MUSAN [55]. The SAD output is then merged with single-speaker region labeling, which comes from the reference transcription for CHiME-5

and an energy-based analysis for Mixer 6.

Stage 2) Perform segment verification by removing utterances which are too short, have incorrect speaker labels, or are non-speech vocalizations. We used a state-of-the-art speaker identification setup with x-vectors [56] and a probabilistic linear discriminant analysis (PLDA) backend [57, 58] for these tasks. The models were trained using the VoxCeleb [59] and VoxCeleb2 [60] corpora augmented with MUSAN [55] and reverberated with the simulated room impulse responses described in [61]. We scored utterance embeddings against embeddings extracted from all speech by its speaker and rejected the utterances below a qualitatively-tuned score threshold.

4.2.3 Mixture List Generation

For consistency, we generated the mixture lists to be compatible with the MERL scripts for generating overlap[3] and with similar properties to the WSJ0 mixtures. Still, there was a lot of freedom in how to pair utterances from the base corpus (i.e. CHiME-5 or Mixer 6) to generate mixtures. As a result, we created the mixture lists algorithmically according to a set of desirable criteria selected to maximize data diversity and utilization of the source corpus:

1. avoiding mixtures of two utterances by the same speaker

[3] `http://www.merl.com/demos/deep-clustering/`

2. minimizing repeated usage of any particular utterance for multiple mixtures

3. maximizing speaker diversity among all mixtures using any particular utterance

4. pairing utterances of similar length

Further description of the algorithm is presented in Appendix A.1.

We selected two microphone conditions from each corpus for use in our experiments. For the far-field CHiME-5 condition, we selected the first channel of the first microphone array. For the near-field CHiME-5 condition, we used the left channel of the binaural microphone for the speaker corresponding to each utterance. For the far-field Mixer 6 condition, we chose channel 9, which is the microphone placed farthest from the speaker. For the near-field Mixer 6 condition, we chose channel 2, which is the lapel microphone worn by the speaker.

4.2.4 Overlap Dataset Design

In constructing the CHiME-5 and Mixer 6 mixture data, we made an attempt to match the WSJ0 mixture dataset as closely as possible. Along those lines, we name the resulting datasets ch5-2mix and mx6-2mix respexctively, to parallel the wsj0-2mix name. We constructed training, development, and test sets of equivalent size (20k, 5k, and 3k mixtures respectively). We chose mixture energy ratio levels following the same distribution as well. However, because the size of each base corpus varied, the

Table 4.1: Synthetic overlap dataset statistics. 'mean utt. usage' refers to the average number of times a single-speaker segment is used in a synthetic mixture, giving a sense of how much repeated speech is present in overlap mixtures. The "train 100k" row refers to a large dataset, discussed in Section 5.2.

source corpus	set	spk. count	mix. count	total length	mean utt. usage	mean mix. length
WSJ0	train	101	20k	30.4 h	4.6	5.5 s
	dev.	101	5k	7.7 h	2.8	5.5 s
	eval.	18	3k	4.8 h	3.4	5.8 s
Mixer 6	train	451	20k	28.3 h	1.0	5.1 s
	dev.	50	5k	6.1 h	1.0	4.4 s
	eval.	45	3k	4.1 h	1.0	4.9 s
	train 100k	453	100k	98.3 h	1.3	3.5 s
CHiME-5	train	32	20k	12.7 h	3.4	2.3 s
	dev.	8	5k	3.3 h	8.1	2.4 s

amount each speaker and utterance were used varied as well. Comparison of usage statistics are in Table 4.1.

In both the ch5-2mix and mx6-2mix datasets, we constructed both near-field and far-field conditions. When doing so, we used identical utterance pairs, as opposed to generating new mixture sets, to reduce the number of confounding factors when comparing speech separation performance between near-field and far-field conditions.

4.3 Construction of Full Synthetic Mixtures

To further aid in the development and evaluation of speech separation systems in realistic conditions, we introduced the WHAMR! dataset that adds reverberation to

WHAM!'s noise augmentation of wsj0-2mix. We generated realistic room parameters which in turn we used to generate room impulse responses that can produce reverberant audio waveforms for each source in a manner similar to the multi-channel version of wsj0-2mix introduced in [62], but with the microphone geometry constrained by the binaural recording setup used to collect the WHAM! noise corpus. The use of synthetically-added noise and reverberation for WHAMR! contrasts with the CHiME-5 and Mixer 6 mixtures in the previous sections, which are constructed using actual recordings of noisy and reverberant speech. As such, those datasets lack ground truth for clean and anechoic speech. WHAMR! provides a contrasting and complementary data paradigm; similarly to other WSJ0-based speech separation datasets, WHAMR! is constructed synthetically, with artificially-mixed speech plus noise and artificial reverberation. This synthetic construction provides the ground truth of all component speech signals with and without reverberation, which is necessary to effectively train and evaluate deep learning-based systems.

4.3.1 WHAMR! Dataset

The WHAMR! dataset[4] is an extension of the WHAM! dataset [40] as mentioned above, which itself is a noise-augmented version of the wsj0-2mix dataset [8]. The wsj0-2mix dataset consists of mixtures of utterances from the WSJ0 corpus, combined

[4]Available at: `http://wham.whisper.ai`

Table 4.2: Room impulse response parameter sampling distributions. Units for all parameters are meters with the exception of reverberation time (T_{60}) which is in seconds and angles in radians.

Room	length	$\mathcal{U}(5,10)$
	width	$\mathcal{U}(5,10)$
	height	$\mathcal{U}(3,4)$
$\mathbf{T_{60}}$	high	$\mathcal{U}(0.4,1.0)$
	med.	$\mathcal{U}(0.2,0.6)$
	low	$\mathcal{U}(0.1,0.3)$

Mic. Center	length	$\frac{\text{length}_{\text{Room}}}{2} + \mathcal{U}(-0.2,0.2)$
	width	$\frac{\text{width}_{\text{Room}}}{2} + \mathcal{U}(-0.2,0.2)$
	height	$\mathcal{U}(0.9,1.8)$
Mic. Array	sep.	noise mic. separation
	θ	$\mathcal{U}(0,2\pi)$
Sources	height	$\mathcal{U}(0.9,1.8)$
	dist.	$\mathcal{U}(0.66,2)$
	θ	$\mathcal{U}(0,2\pi)$

with random gain between 0 and 5 dB to create overlapping speech. There are four configurations: a *min* condition where the mixture is trimmed to the length of the shorter utterance and the corresponding non-trimmed *max* condition, both available at 8 kHz and 16 kHz sampling rate. The mixtures are partitioned into training, validation, and test sets of 20,000, 5,000, and 3,000 mixtures respectively. In the WHAM! dataset, each speech mixture from the wsj0-2mix corpus was assigned to a randomly sampled excerpt from noises recorded with binaural microphones in various urban environments throughout the San Francisco Bay Area, and mixed such that the louder speaker was at a randomly selected SNR between -6 and $+3$ dB relative to the noise [40].

WHAMR! extends WHAM! by introducing reverberation to the speech sources in addition to the existing noise. Room impulse responses were generated and convolved using pyroomacoustics [63] according to the random room configurations shown in

Table 4.2. Reverberation times were chosen to approximate domestic and classroom environments [39] (as we expect these to be similar to the restaurants and coffee shops where the WHAM! noise was collected), and further classified as high, medium, and low reverberation based on a qualitative assessment of the mixture's noise recording.

We created spatialized versions—*anechoic* and *reverberant*—of all components of the original WHAM! dataset, except noise, which was recorded spatialized. The anechoic sources (i.e., direct path signals) serve as targets to reverberated sources for models involving dereverberation, allowing them to be trained without needing to account for the time delay of the spatialized sources. In spatializing the audio, we generated a two-channel version of the dataset, using microphone spacing from the WHAM! noise metadata, but in this study we focus on single-channel separation and use only the left channel. The spatialized audio was rescaled to remove attenuation, such that the non-spatialized WHAM! and anechoic WHAMR! differ only by small time delays, and we found negligible performance differences when training and testing models using the two datasets. While the results for non-reverberant conditions in Section 4.4.2.4 use anechoic WHAMR!, they are directly comparable with WHAM! [40].

Since all source, noise, and reverberated components and their combinations are included in the corpus, several enhancement, separation, and joint enhancement-separation tasks are enabled for training and evaluation. For example, in separating

noisy and reverberant speech, we may want to produce either two clean, anechoic recordings or two clean, reverberant recordings, leaving dereverberation to post-processing. For this dataset we chose to define four core separation tasks:

- **clean** – anechoic clean mixture to anechoic sources
- **noisy** – anechoic noisy mixture to anechoic sources
- **reverberant** – reverberant clean mixture to anechoic sources
- **noisy and reverberant** – reverberant noisy mixture to anechoic sources

All other configurations are only considered and evaluated as sub-components to the above tasks. The evaluation metric we used was Scale-Invariant Signal-to-Distortion Ratio (SI-SDR), as described in Section 2.4.3.1. Since each condition has its own unprocessed SI-SDR (i.e. the value resulting when SI-SDR is evaluated using the input waveform), comparisons across tasks can be difficult. By restricting to the above tasks, where the *targets* are the same in all four conditions, raw SI-SDR can be thought of as a directly comparable, "objective" quality metric of the output sources across tasks. SI-SDR *improvement*, i.e. the difference in SDR between the system output and the unprocessed mixture, provides additional insight by reporting how much improvement a system has made to the signal.

4.4 Experimental Results

The primary goal of these experiments is to demonstrate a degradation in performance of speech separation systems in noise and reverberation. This involves selecting models that are representative of the state of the art, verifying their performance on established datasets, and then evaluating them in the new conditions, providing baselines and oracle results when possible.

4.4.1 Mixtures of Real Conditions

This subsection is dedicated to experimental analysis of existing speech separation techniques on the datasets introduced in Section 4.2 consisting of mixtures created using real noisy and reverberant speech sourced from the Mixer 6 and CHiME-5 corpora.

4.4.1.1 Models

The models we used for our evaluation of the mixtures of real conditions described in Section 4.2 were utterance-level Permutation Invariant Training (uPIT), Deep Clustering (DPCL), Recurrent Selective Attention Network (RSAN), and TasNet-BLSTM. More details on these models can be found in Section 2.4. To ensure our implementations were correct and give the most fair comparisons, we largely selected hyperparameters for the networks that match those of the originally published results.

The spectrograms were generated using a STFT from down-sampled 8 kHz audio with a window length of 512 and a step of 128 in the case of the uPIT and RSAN experiments, and a window length of 256 and a step of 64 in the DPCL setup. The input to the networks was the mixture magnitude spectrum. The input speech was a mixture of two speakers, and the systems always output exactly two masks (i.e., $K = 2$ in Section 2.3).

Both the uPIT and RSAN networks used in our experiments consisted of two 600-node BLSTM layers followed by a linear layer, with a sigmoid output. The uPIT network had an input dimension F of 257 with a final output of 514 for two speaker masks, while the RSAN network had an input dimension of 514 to account for the attention mask, with a final output of 257, and was run twice to recursively extract two speaker masks.

The DPCL network used in our experiments also used two 600-node BLSTM layers followed by a linear layer, with hyperbolic tangent and ℓ_2-normalization. The input dimension F was 129, and the output dimension was 5,160, corresponding to an embedding dimension of 40 (i.e., $E = 40$). The backend used in the DPCL setup to produce masks was k-means clustering with cosine distance between embedding vectors with $k = 2$ (two speakers).

The TasNet-BLSTM network uses parameters matching those from the original TasNet publication [15]. This means 500 basis filters of length 40 samples with a shift

of 40 samples. The BLSTM is 4 layers of 500 units in each direction.

4.4.1.2 Training

The STFT mask approximation networks (uPIT and RSAN) were trained using permutation-invariant training with the standard magnitude MSE loss function (2.11) using the Ideal Ratio Mask. The Deep Clustering network was trained using the standard loss function (2.8). These networks were trained for 200 epochs with an initial learning rate of 0.001 using the Adam [64] optimizer.

For the TasNet-BLSTM network, the negative SI-SDR objective was used. This network was trained for 100 epochs with an initial learning rate of 0.001 using the Adam optimizer. Performance was monitored on the 'cv' set, and the learning rate was halved if performance did not improve for three epochs. In addition, gradient clipping was applied with a maximum ℓ_2 norm of 5.

4.4.1.3 Evaluation

For evaluation, we used the three companion metrics described in Section 2.4.3.1 as implemented in the mir_eval library [65]: signal to distortion ratio (SDR), signal to interferences ratio (SIR), and signal to artifacts ratio (SAR) [36]. Our primary, and most typical speech separation metric, was SDR, while we additionally used SIR and SAR for our initial experimental comparisons. We also provide the SDR of the

Table 4.3: Comparison of experimental setup on the WSJ0 2-speaker mixture dataset.

	Method	SDRi [dB]
From the Literature	uPIT-BLSTM-ST [11]	10.0
	RSAN [14]	8.6
	DPCL [8]	5.8
	DPCL++ [9]	10.8
	TasNet-BLSTM [15]	11.1
From our Experiments	uPIT	9.3
	RSAN	9.5
	DPCL	7.7
	TasNet	14.0

unprocessed corpus for reference and for computation of SDR improvement.

4.4.1.4 Results and Discussion

We analyzed the robustness of the speech separation techniques based on our implementations of RSAN, uPIT, DPCL, and TasNet with multiple datasets, as introduced in Section 4.2. We used widely-reported SDR improvement on the wsj0-2mix [8] dataset to verify that our implementations used are within the range of state-of-the-art performance, reflected in other reports, as shown in Table 4.3.

Results of experiments containing models trained purely on in-domain data are presented in Table 4.4. Sub-tables (a), (b), and (c) report SDR, SIR, and SAR respectively, with SDR being the primary metric. In all cases larger numbers reflect better performance. Each row reports the results for a different architecture, and each column is a different dataset. Moving from left to right, the datasets are in order of

Table 4.4: Comparison of SDR, SIR, and SAR in matched-condition train and eval sets.

(a) SDR [db]

Datset		wsj0	mx6 near	ch5 near	mx6 far	ch5 far
Network	uPIT	9.4	6.9	7.1	**4.1**	**2.2**
	RSAN	9.7	6.9	**7.2**	3.5	1.8
	DPCL	7.8	3.2	2.7	−3.1	−2.9
	TasNet	**14.0**	**9.0**	6.2	3.6	0.7

(b) SIR [db]

Dataset		wsj0	mx6 near	ch5 near	mx6 far	ch5 far
Network	uPIT	14.2	10.3	10.1	**5.9**	**4.1**
	RSAN	14.5	10.5	10.5	5.6	3.8
	DPCL	16.3	9.9	8.5	2.7	3.6
	TasNet	**23.2**	**15.1**	**12.4**	5.9	2.4

(c) SAR [db]

Datset		wsj0	mx6 near	ch5 near	mx6 far	ch5 far
Network	uPIT	11.8	10.7	11.3	10.4	9.3
	RSAN	12.0	10.8	**11.2**	**10.7**	**10.4**
	DPCL	10.3	8.6	9.1	8.6	8.5
	TasNet	**14.7**	**11.0**	8.3	9.4	8.3

expected increasing difficulty. The wsj0 dataset has neither noise nor reverberation. The mx6 data has light noise, while the ch5 data has heavy noise. And, the near-field data has minimal reverberation, while the far-field data does have reverberation. Overall, performance degrades as we move from clean to more noisy conditions as well as from near- to far-field. We also see that the uPIT and RSAN networks produce similar results due to their similar separation framework, while the DPCL shows more significant degradation. This may be due to a lack of tuning the speech/noise threshold parameter, and also is not representative of the more advanced and better-performing DPCL++ [9] method reflected in Table 4.3, which includes improvements to signal reconstruction and soft masking. Interestingly, although TasNet is the best-performing model in the clean, near-field case by far, it suffers greater degradation in the adverse acoustic conditions and is not the best-performing method. This is likely due to the waveform-level training objective. It seems the magnitude target objectives are the most condition-robust. Similar trends are reflected across all three metrics, so we chose to report only the standard SDR metric for all subsequent experiments. For similar reasons, we restrict our results to the RSAN method, chosen due to being one of our best-performing methods.

Experimental results of all train–test configurations using training sets of size 20k are shown in Table 4.5. In this table, rows report the dataset used in training the model, while columns are the dataset used for evaluation. The oracle row is performance

Table 4.5: SDR with 20k-mixture train sets and varying test conditions. To emphasize the difference between near and far conditions, the numbers greater than 5.0 are highlighted, with boldface used for the best result per evaluation condition. Oracle refers to use of the Ideal Ratio Mask.

Eval		RSAN				
		wsj	mx6 near	ch5 near	mx6 far	ch5 far
Train	wsj	**9.7**	5.0	6.1	1.1	1.1
	mx6 near	7.5	**6.9**	7.0	2.2	1.0
	ch5 near	7.1	5.6	**7.2**	2.3	1.7
	mx6 far	2.4	3.0	3.5	**3.5**	0.5
	ch5 far	1.5	–0.6	1.1	–1.1	**1.8**
Oracle		14.0	13.2	13.1	9.6	10.9
Corpus		0.2	0.2	0.3	0.3	0.3

using the oracle Ideal Ratio Mask, and the corpus row is the SDR of the input mixture. SDR improvement can be computed by subtracting the corresponding "corpus" value. Interestingly, the dataset mismatch among clean and near-field conditions did not cause a serious degradation despite the noisy and speaking-style variations across the datasets. The models trained on the Wall Street Journal, near-field Mixer 6, and near-field CHiME-5 data all resulted in an SDR over 5 when evaluated in one of those conditions. However, we observed a large degradation in any combination of training and test data when including far-field conditions. Although the oracle performance computed from the ideal ratio mask, shown in Table 4.5, reflects intrinsic difficulties of far-field conditions compared to near-field conditions, the observed degradation in using speech separation was even greater.

4.4.1.5 Conclusion

We have demonstrated that there are shortcomings of supervised speech separation techniques in conditions including noise and reverberation, as well as in mismatched conditions. In some cases, the degradation is severe, providing evidence that further work must be done to analyze these conditions and develop techniques to address them.

However, one shortcoming of the use of mixtures of real conditions with noise and reverberation is that the noise and reverberation must be present in the ground truth signal. As such, to better evaluate the capability of these speech separation techniques to handle noise and reverberation, we also evaluated systems on data where the noise and reverberation were added to the mixtures synthetically, allowing for greater control over the ground truth signal used in training and evaluation, i.e. the ability to not include the noise or reverberation in the target.

4.4.2 Fully Synthetic Mixtures

This subsection is dedicated to experimental analysis of standard systems on the datasets introduced in Section 4.3 consisting of mixtures created using artifically-added noise and the application of synthetic room impulse responses. This also includes experiments analyzing different separation techniques and their ability to suppress noise and reverberation through being trained explicitly for the speech

enhancement tasks enabled by the data.

4.4.2.1 Models

For our experiments, we use four basic network configurations, all under the same spectro-temporal masking paradigm as described in Section 2.3. In enhancement, the internal masking network produces a single mask, attempting to suppress noise and/or reverberation. In separation, the masking network produces a mask for each speech signal, attempting to suppress the interfering speakers from each target speaker.

The four configurations we use are the possible combinations of two spectral feature extractors and two internal masking networks. The feature extractors we compare are a standard short-time Fourier transform (STFT) and a TasNet-style learned basis transform [15, 16], which consists of projecting sliding-window subsegments of the waveform onto a set of learned basis functions. The resulting weights can be applied to a reconstruction set of basis functions and summed together along the same sliding window to reconstruct the signal under a similar paradigm to overlap-and-add for the STFT. For internal masking, we evaluate both bi-directional long short-term memory (BLSTM) networks (the typical internals of earlier deep learning-based speech separation systems [8, 9, 11, 15, 40, 66]) and temporal convolutional networks (TCN) [67] with dilated convolutions (popular in recent state-of-the-art separation techniques [16, 17]).

For consistency with the prior WHAM! work [40], our BLSTM architecture has four BLSTM layers with 600 units in each direction followed by a fully-connected layer for each output mask. A dropout of 0.3 was applied on each BLSTM layer output except the last. The TCN architecture was chosen to match the best system reported in [16]. It consists of a 128-dimensional bottleneck, 128-dimensional skip-connection paths, and 512 channels in the convolutional blocks, with kernel size 3, 8 blocks per repeat, and 3 repeats.

The STFT features are also chosen to be consistent with [40], with a window length of 32 ms and hop size of 8 ms. The log of the magnitude spectrum is used as input to the internal masking network. The learned basis feature parameters are also chosen to be consistent with [40], with a 10 ms window and 5 ms hop, with 500 learned basis vectors. While the original BLSTM TasNet [15] used a gated convolutional encoder, in this work we use a single learned encoder and ReLU nonlinearity as in Conv-TasNet [16] for both the BLSTM and TCN masking networks with learned bases. For separation, we evaluate learned basis configurations only, as they have been shown to outperform STFT-based methods on clean data, and performed best in preliminary experiments. However, we perform full comparisons of the differing features for enhancement, for which TasNet-like systems have only rarely been evaluated [68].

4.4.2.2 Training

We train all networks using permutation invariant training [11] with the scale-invariant signal-to-distortion ratio (SI-SDR, also referred to as SI-SNR) waveform-level training objective [9, 15, 31], presented in (2.12). SI-SDR is also the evaluation metric and allows for end-to-end joint training of cascaded enhancement and separation models.

All networks are trained on 4 second segments using the Adam optimizer [64]. The learning rate is decreased by a factor of 2 if validation loss does not improve for 3 consecutive epochs. Gradient clipping is applied with a maximum ℓ_2 norm of 5. Models are trained for 100 epochs with an initial learning rate of 10^{-3}.

4.4.2.3 Evaluation

For all experiments, we report results using SI-SDR, which, it is important to note, is the same function as the training objective. Furthermore, because the input mixture SI-SDR between tasks is highly variable, we also report the SI-SDR improvement (Δ), i.e., the difference between output and input SI-SDR.

4.4.2.4 Results and Discussion

Table 4.6 shows the results of our core systems, without cascade. Each row is one of the four permutations resulting from the presence of noise or reverberation in the data. The "Input" column contains the SI-SDR value of the input mixture. The "Output" column

Table 4.6: SI-SDR [dB] results for a single separation network. Highlighted rows represent new WHAMR! conditions.

Input			Conv-TasNet		TasNet-BLSTM	
Noise	Reverb	Input	Output	Δ	Output	Δ
		0.0	12.9	12.9	**14.2**	**14.2**
✓		−4.5	7.0	11.5	**7.5**	**12.0**
	✓	−3.3	4.3	7.6	**5.6**	**8.9**
✓	✓	−6.1	2.2	8.3	**3.0**	**9.2**

Table 4.7: SI-SDRi [dB] (Δ) comparison of our implementations with the best Conv-TasNet number in [16] and the corresponding learned feature configuration of 512 bases, window length 16, window shift 8.

TasNet-BLSTM	Conv-TasNet	Conv-TasNet [16]
16.6	14.4	15.3

is the SI-SDR for the system, with the Δ column containing SI-SDR improvement, computed as the difference between the output and input. The Conv-TasNet and TasNet-BLSTM headings refer to the two architectures used. Reverberation seems to be more challenging than noise as reflected by the lower SI-SDR. In terms of both raw SI-SDR and SI-SDR improvement, the numbers in the latter two rows containing reverberation are lower than those in the first two rows without. While the noisy and clean conditions are fairly comparable in terms of SI-SDR improvement, they still differ significantly in terms of raw SI-SDR. Interestingly, we observe consistently better performance by the BLSTM model over the TCN model, which is somewhat unexpected. Indeed, although the BLSTM contains many more parameters than the TCN, this result contradicts prior results in the literature [15, 16]. A comparison of clean separation models with a smaller basis window is shown in Table 4.7. In this

Table 4.8: SI-SDR [dB] for two-speaker enhancement tasks.

Net		Denoise		Dereverb	
Feature	Processor	Output	Δ	Output	Δ
Learned	TCN	10.8	9.6	7.2	3.2
Learned	BLSTM	**11.2**	**10.1**	**8.5**	**4.4**
STFT	TCN	8.4	7.2	4.0	0.0
STFT	BLSTM	9.5	8.4	5.9	1.8
Input SI-SDR:		1.2		4.0	

plot, all three systems have the same feature parameters, but our TasNet-BLSTM system (left) outperforms our Conv-TasNet implementation (middle) as well as the number reported in the literature (right), confirming that the performance difference is not due to the window parameters.

In addition, we note that the TasNet-BLSTM numbers in the first two rows of Table 4.6 are considerably better than the corresponding numbers in the original WHAM! paper [40]. The newer network uses the same configuration, but is trained with more aggressive gradient clipping and stagnation learning rate adjustment, which supports the findings regarding training optimizer parameters reported in [16, 68].

Table 4.8 shows experimental results with enhancement networks. The rows contain the four permutations of choice of feature (learned features vs. the STFT) and DNN masking architecture (BLSTM and TCN), as well as the SI-SDR of the input waveforms as well. The columns contain the denoising and dereverberation of two-speaker mixtures evaluation sets, which we used as a proxy for all other possible enhancement conditions of this dataset. In both conditions, the system using

a BLSTM architecture along with learned features resulted in the highest numbers. Since performance trends are consistent across these two tasks, we think this is reasonable evidence to conclude that the learned feature BLSTM model (TasNet-BLSTM) is the best architecture for enhancement. While the learned basis TCN and BLSTM perform similarly, we see significant drops in performance moving from learned basis to STFT features. This suggests that the benefits of using learned features shown in speech separation are also likely present in speech denoising and dereverberation as well.

4.4.2.5 Conclusion

We have provided preliminary evidence to demonstrate that, although noise and reverberation do degrade overall performance, networks with learned basis feature representations are effective not only in separation but also in speech enhancement. We do, however, also see that noise and reverberation still pose a challenge in comparison to the clean conditions.

4.5 Conclusion

We have introduced a number of datasets to aid in the evaluation and development of speech separation systems in noisy and reverberant conditions. This includes the mx6-2mix and ch5-2mix datasets, consisting of artificial mixtures of real conditions

from the Mixer 6 and CHiME-5 speech corpora, with a variety of conditions with varying levels of noise and reverberation. In addition we have introduced the synthetic dataset WHAMR!, an extension of the WHAM! noisy speech separation dataset to include reverberation, with the goal of further promoting the advancement of speech separation technologies towards more realistic conditions.

We have used these datasets to demonstrate that there is a degradation in the performance of conventional speech separation systems in these more challenging conditions. It is also important to note that we see a greater level of degradation in the mixtures of real conditions compared to the conditions that were artificially generated through digital summing of noise and simulated impulse responses. Further analysis of this effect is presented in Chapter 6.

Chapter 5

Techniques for Improved Performance in Noise and Reverberation

5.1 Overview

In this chapter, we present some initial techniques aimed to improve performance in noisy and reverberant conditions and evaluate their effectiveness. In some sense, the addition of noise and reverberation to the speech separation task is something of a new problem. Just as it was important to demonstrate performance degradation of traditional separation techniques (Chapter 4), it is also important to explore the

"low-hanging fruit" for the new problem. Dealing with the adverse conditions noise and reverberation create for speech technology in general is not a new problem, and there has been a great deal of effort applied to addressing these conditions in other tasks like speech recognition and speaker identification. As a result, a logical first step is to try some of the general approaches that are successful in other tasks in these conditions, and document their impact on speech separation.

5.2 Augmented Training Data

5.2.1 Introduction

One of the most ubiquitous approaches used to improve speech technologies is the addition of more training data. Not only do systems trained on a larger *quantity* of data generally perform better [54], but systems trained on a wide *variety* of conditions tend to be more robust to different conditions, even if the exact evaluation conditions are not included in the training data [56]. To this end, the use of augmentations to existing training data has been shown to be successful as well [61, 69].

In this section, we explore the use of more and a wider variety of data in training systems for speech separation in noise and reverberation. This was enabled in part through the efforts presented in Section 4.2 to create a pipeline to generate new mixture datasets from existing speech corpora.

5.2.2 Method

Due to the extensive size of the Mixer 6 corpus in comparison to the WSJ0 and CHiME-5 corpora, we were able to construct additional, larger training sets for both the near-field and far-field Mixer 6 conditions using the pipeline described in Section 4.2.3. In these datasets the total size of the training data was increased five-fold to 100k (train 100k), allowing us to do a deeper analysis of how the quantity of training data affects model performance. Referring back to Table 4.1, we can see from the "train 100k" row that extending the Mixer 6 dataset to 100k mixtures is not pushing the limits of the source corpus, which still has minimal reuse of speech data compared to the smaller wsj0-2mix dataset.

We also constructed new training sets by *combining* each of the five constructed datasets (wsj0-2mix, ch5-2mix near and far, mx6-2mix near and far). Two iterations were created: In the first, the combinations were sub-sampled to maintain the size of 20k training examples, allowing an analysis of data variety without an increase in amount of data. In the second, they were fully combined, resulting in 100k examples, making it comparable in size to the 100k mixture dataset from Mixer 6 alone. These sets allowed us to analyze the potential for producing a robust system based on training on a wide variety of properly manicured data.

5.2.3 Experimental Configuration

These experiments are an extension of the work presented in Chapter 4 on mixtures of real conditions, and the bulk of the experimental design can be found there. Details of the data creation are in Section 4.2 and details on the experimental setup are in Section 4.4.1.

However, in this chapter the experiments are focused on larger amounts and wider variety of training data. As such, our focus is on the 100k mixture datasets, namely the Mixer 6 near-field and far-field setups and the combination of the five 20k mixture conditions, with the corresponding 20k-size subsampled datasetss acting as baselines. Additionally, based on the findings in Chapter 4, we restricted our experiments to only the RSAN architecture, serving as a representative system.

5.2.4 Results and Discussion

Table 5.1 contains the results of our experiments involving larger amounts of training data. Rows contain different training datasets, and the columns are the different evaluation datasets constructed from the WSJ0, Mixer 6, and CHiME-5 corpora. The Train 20k and Train 100k sections of rows contain the 20k- and 100k-mixture versions of the three datasets that could be constructed to contain 100k training samples. The "combo" row is the condition containing a mixture of the wsj0-2mix, mx6-2mix, and ch5-2mix data. Again, the oracle row contains the SDR value of the Ideal Ratio

Table 5.1: 20k-mixture and 100k-mixture train sets SDR [dB] comparison. SDR values over 5.0 are highlighted. Oracle numbers refer to the use of the Ideal Ratio Mask

Eval		RSAN				
		wsj	mx6 near	ch5 near	mx6 far	ch5 far
Train 20k	mx6 near	7.5	6.9	7.0	2.2	1.0
	mx6 far	2.4	3.0	3.5	3.5	0.5
	combo	7.5	5.5	6.2	2.8	2.2
Train 100k	mx6 near	8.0	7.5	7.5	2.7	1.6
	mx6 far	3.4	3.6	4.2	4.5	1.3
	combo	9.0	6.8	7.5	4.1	3.1
Oracle		14.0	13.2	13.1	9.6	10.9
Corpus		0.2	0.2	0.3	0.3	0.3

Mask, and the corpus row contains the SDR value of the input mixture. We see that the training conditions comprised of a combination of all corpora (combo) result in performance near that of matched training for each condition (refer to Table 4.6 for the CHiME-5 matched-training results). Increasing the amount of training data five-fold (train 100k combo) improves performance further. This result suggests that multi-condition training, which is widely used in speech processing, is still effective for deep-learning based speech separation. However, the performance in far-field conditions is quite poor, even with multi-condition training or increased quantity of training data. In other words, training on a variety of conditions helps remove deficiencies in the model to handle other conditions *relative to other systems*, but does not improve the overall deficiency in performance of separation systems on difficult conditions.

From our experiments, we can conclude that current speech separation techniques are reasonably robust across the datasets in near-field conditions. However, these experiments also reveal that both matched and multi-condition training have significant degradation in far-field conditions, a differing result from other learning-based speech processing, notably automatic speech recognition [70, 71].

5.2.5 Conclusion

The biggest takeaway from these experiments is that adding more and a wider variety of data to the training of a separation system *does* improve performance, but does not specifically or fully address the issue of noise and reverberation in particular. Though the lack of robustness can be mitigated by training models on more data from multiple conditions, there remains a significant gap from the oracle Ideal Ratio Mask performance in far-field conditions, which advocates a need for extending separation techniques to address the issues present in far-field speech mixtures.

It is worth pointing out that one thing that is lost in the shift from the use of the STFT to learned spectral transforms is the ability to compute "oracle" masks and provide such analysis. However, while the ability to compute a "performance gap" may be lost, there is also a question as to whether or not the oracle performance is a realistic, attainable target, as it involves perfect separation and partitioning of the non-speech background noise, which the waveform-level evaluation metric should be

sensitive to. A further exploration into these issues regarding the sensitivity of the standard metric to the noise present in the ground truth is presented in Chapters 6 and 8.

5.3 Cascaded Models

5.3.1 Introduction

One major value of the WHAMR! data introduced in Section 4.3, enabled by the use of fully synthetic noise and reverberation, is that various configurations of signal mixtures and targets can be created, including unique mixture/target combinations for particular tasks like separating reberberant speech, with or without dereverberating the speech or denoising both one and two-speaker mixtures. In this section, we explore breaking the overall noisy and reverberant separation tasks into sequential subtasks that can have individually-trained models that are cascaded, with each system feeding into the next, to solve the overall task. The main motivation is that jointly separating and enhancing may be too difficult for a single network to learn, and modularization may allow the networks to focus on specific tasks. Such multi-stage approaches have previously been explored for denoising plus dereverberation [72, 73], separation plus dereverberation [74], and denoising plus separation [40].

5.3.2 Method

The cascaded configurations we considered consist of an optional pre-enhancement system cascaded into a separation network cascaded into an optional post-enhancement system. We evaluated all combinations where noise is removed by either the pre-enhancement or the separator, and reverberation is removed by either pre-enhancement, post-enhancement, or the separator. Post-separation denoising was not considered, as separation-without-denoising is a somewhat ill-defined task: noise does not 'belong' to either speech signal, so it is unclear how the network should distribute the noise when not removing it.

For cascaded systems, the sub-models were trained with appropriate input and targets for each sub-task. For example, in the system consisting of denoising followed by separation then dereverberation, the networks were trained as follows: pre-enhancement is trained with noisy reverberant mixtures as input and noise-free reverberant mixtures as output; the separator with reverberant mixtures as input and single reverberant sources as output; and post-enhancement with single reverberant sources as input and single anechoic sources as output.

Due to the scale-invariant nature of the negative SI-SDR loss function, each model's outputs have no constraint to be within any particular dynamic range, and we thus observed strong degradation in performance in cascaded systems when sub-models are trained separately, due to the scaling mismatch between the output of one

model and the training data of the next. To address this problem, we scaled each output estimate $\hat{\mathbf{s}}$ of a target source $\mathbf{s}$, obtained from an input mixture $\mathbf{x}$, to make it consistent with the scaling of $\mathbf{s}$ in $\mathbf{x}$. Because $\mathbf{s}$ is unknown, we need to rely on $\hat{\mathbf{s}}$ and $\mathbf{x}$ alone. If we assume that the interfering signal $\mathbf{e} = \mathbf{x} - \mathbf{s}$ is orthogonal to $\mathbf{s}$, which is generally approximately the case (note: additional commentary on the validity of this assumption is in Section 7.2.1), and that the direction of $\hat{\mathbf{s}}$ is close to that of $\mathbf{s}$, then a reasonable choice for the rescaling factor $\beta(\hat{\mathbf{s}}|\mathbf{x})$ is that obtained by ensuring that $\beta(\hat{\mathbf{s}}|\mathbf{x})\hat{\mathbf{s}}$ is orthogonal to the residual $\hat{\mathbf{e}} = \mathbf{x} - \beta(\hat{\mathbf{s}}|\mathbf{x})\hat{\mathbf{s}}$. This results in a scaling factor

$$\beta(\hat{\mathbf{s}}|\mathbf{x}) = \frac{\langle \mathbf{x}, \hat{\mathbf{s}} \rangle}{\|\hat{\mathbf{s}}\|^2}. \tag{5.1}$$

As the estimate $\hat{\mathbf{s}}$ improves (i.e., $\hat{\mathbf{s}}$ and $\mathbf{s}$ become more colinear), the scaling factor improves as well.

Finally, when the best-performing system of a WHAMR! task is a cascaded model, we also evaluated the system with additional end-to-end tuning. Since all component systems are waveform-to-waveform, we could tune the entire system by performing additional training through all cascaded sub-models directly. End-to-end joint training of sub-models has been shown to be successful in joint training of automatic speech recognition with enhancement and separation [75–78].

5.3.3 Experimental Configuration

These experiments are an extension of the work presented in Chapter 4 on fully-synthetic mixtures, and the bulk of details relating to the data and experimental setup can be found in Sections 4.3 and 4.4.2 respectively. However, there are some differences relevant to the evaluation of the cascaded model method.

One difference is that based on the results presented in the previous chapter in Table 4.8, we restricted ourselves to the best-performing models, namely the ones using learned TasNet-style features with BLSTMs. Furthermore, we tuned cascaded systems by additional training of the entire end-to-end system; specifically we trained the models for 25 epochs with a learning rate of 10^{-4}, compared to 100 epochs with an initial learning rate of 10^{-4} for the primary training.

We also extended our evaluation conditions to the 16 kHz conditions and *max* data subset. However, as the SI-SDR loss is undefined for silent sources, training models on the *max* data subset is cumbersome, as the 4 s segments randomly sampled during training occasionally fall within regions where only one speaker is talking, leading to undefined loss for the other speaker. Thus, for the 16 kHz *max* condition, we trained on 16 kHz *min*.

Table 5.2: Comparison of cascaded models. A dash indicates speech separation without denoising/dereverberation, while × indicates no enhancement sub-model was used. Results are sorted by increasing performance. The highlighted rows indicate the non-cascaded single-model baseline. Δ indicates SI-SDR improvement.

(a) noisy condition

System		SI-SDR	
Pre-Enh. Removes	Separate Speech while Removing	Output	Δ
×	noise	7.5	12.0
noise	–	**8.1**	**12.6**
Input SI-SDR:		–4.5	

(b) reverberant condition

System			SI-SDR	
Pre-Enh. Removes	Separate Speech while Removing	Post-Enh. Removes	Output	Δ
×	rev.	×	5.6	8.9
rev.	–	×	6.4	9.7
×	–	rev.	**6.6**	**9.9**
Input SI-SDR:			–3.3	

(c) noisy and reverberant condition

System			SI-SDR	
Pre-Enh. Removes	Separate speech while removing	Post-Enh. Removes	Output	Δ
×	noise, rev.	×	3.0	9.2
noise	rev.	×	3.5	9.7
noise, rev.	–	×	3.6	9.7
rev.	noise	×	3.7	9.8
×	noise	rev.	3.7	9.8
noise	–	rev.	**4.0**	**10.1**
Input SI-SDR:			–6.1	

5.3.4 Results and Discussion

Table 5.2 shows the results of the cascaded model experiments. Each sub-table corresponds to one of the conditions including noise, reverberation, or both. Each row corresponds to one of the potential configurations of cascaded models, with an optional pre-separation enhancement model that can denoise or dereverberate and an optional post-separation enhancement model that can dereverberate, with the remaining, with the enhancement being handled jointly with the separation module in cases where there is no separate module. The systems have been ordered in increasing performance, with the non-cascaded baseline (i.e. no separate enhancement module) being highlighted. We see that in general, moving the speech enhancement (i.e., denoising and/or dereverberation) tasks to a separate model from separation seems to help performance. From Tables 5.2(b) and (c), reverberation appears to be particularly difficult for the separation network to remove. We also see that removing reverberation post-separation is slightly better than pre-separation. As two sources will not have the same room impulse response, the dual-source (pre-enhancement) dereverberation network would have to appropriately compensate for two reverberation patterns, while the single-source dereverberation (post-enhancement) network handles only one. The separator network likely has a harder time separating the still-reverberant speech, but this effect appears to be smaller than the difference in single- and double-source dereverberation.

Table 5.3: SI-SDR [dB] comparison of best models with and without additional training. Dashes indicate the best system was not cascaded and thus was not subject to tuning. Δ indicates SI-SDR improvement.

Input			Best System w/o Tuning		Tuned	
Noise	Reverb	Input	Output	Δ	Output	Δ
		0.0	14.2	14.2	–	–
✓		–4.5	8.1	12.6	8.3	12.9
	✓	–3.3	6.6	9.9	7.0	10.3
✓	✓	–6.1	4.0	10.1	4.7	10.8

While the cascaded systems do have 2 or 3 times as many parameters as the non-cascaded system, this does not seem to be the sole source of performance improvement, as single models with increased numbers of BLSTM layers provided little performance gain over the results in Table 4.6. Furthermore, training equivalent cascaded systems from scratch without individual pre-training of the pre-enhancement, separation, and post-enhancement stages provided noticeably less performance improvement over the single network results from Table 4.6 than the reported cascaded systems in Table 5.2.

Table 5.3 shows the results of tuning the cascaded systems with additional end-to-end training. The central column of results are simply reproductions of the best-performing systems from Table 5.2, with the column on the right showing the results after those cascaded systems have undergone additional end-to-end training. Tuning the systems helps, although the performance gains are minor. The noisy and reverberant system, which contains three sub-models in contrast to the others with two, shows the greatest improvement. This suggests training helps with improving the coupling

Table 5.4: SI-SDR [dB] evaluation of 16 kHz conditions using the best model configuration trained on the 16 kHz *min* subset. Δ indicates SI-SDR improvement.

Input		16 kHz Min			16 kHz Max		
Noise	Reverb	Input	Output	Δ	Input	Output	Δ
		0.0	12.9	12.9	0.0	12.7	12.7
✓		−4.6	7.8	12.4	−5.8	7.5	13.3
	✓	−3.3	5.6	8.9	−3.4	5.4	8.8
✓	✓	−6.2	3.7	9.9	−7.2	3.5	10.7

Table 5.5: SI-SDR [dB] evaluation of the best 16 kHz model on Mixer 6 and CHiME-5 data. All data is the 16 kHz Min condition. Δ indicates SI-SDR improvement.

Dataset	Input	Output	Δ
WHAMR!	−6.2	3.7	9.9
mx6-2mix near	0.0	1.7	1.7
ch5-2mix near	0.0	2.2	2.2
mx6-2mix far	0.0	−8.2	−8.2
ch5-2mix far	0.0	−8.7	−8.7

of the connected models.

Table 5.4 shows the results of our 16 kHz systems. As mentioned earlier, we trained on 16 kHz *min* and evaluated on both the *min* and *max* conditions. Although the performance on 16 kHz data is worse than in the 8 kHz systems, there does not appear to be any significant breakdown in performance. Similarly, performance in the *max* condition is only slightly worse than the *min* condition. Although the SI-SDR improvement in the noisy case is better in *max* than *min*, this is likely due to differences in amount of speech and does not reflect any significant difference in performance.

In addition, we evaluated the best 16 kHz models on the Mixer 6 and CHiME-5

datasets introduced in Section 4.2, with results shown in Table 5.5. The performance in these new conditions is extremely poor. This may be attributed to the mismatch in train and test conditions, though this is largely unavoidable due to the fundamental differences in how these datasets are constructed. However, the severely negative SI-SDRi values are not likely representative of the qualitative system output. As presented, this approach does not seem appropriate for solving the conditions represented in the mx6-2mix and ch5-2mix datasets. However, the differences in data construction may be impacting evaluation as well, as is hinted at through the SI-SDR and SI-SDRi values being identical. The noise and reverberation present in the ground truth may be contributing to evaluation issues. This is further explored in Chapter 6.

5.3.5 Conclusion

We have also demonstrated the value in using cascaded models combining pre-trained separation and enhancement modules, and of further jointly fine-tuning them, establishing strong baseline results for the WHAMR! dataset.

However, rather than improve performance in real conditions, we have further exposed differences between the datasets created using real conditions and fully-synthetic conditions. This suggests that further analysis is important into what is responsible for these differences.

5.4 Conclusion

We have demonstrated some simple techniques to improve performance of speech separation systems in conditions with noise and reverberation. The cascaded systems partially closed the gap in performance between clean and noisy/reverberant speech separation systems, with a difference of only a couple dB in terms of SI-SDR improvement.

However, these works have raised a number of questions. One big consideration is why the cascaded systems failed to provide great improvements to the mixtures of real noisy and reverberant data. In addition, the fully-synthetic data of WHAMR! makes apparent some complications of the use of absolute SI-SDR as a metric compared to SI-SDR improvement. For example, in Table 5.4, the comparison between the clean and noisy/reverberant conditions between SI-SDR and SI-SDRi shows that the quality of speech output is vastly different despite having similar improvement, suggesting a significant portion of the gains are from speech enhancement, not speech separation.

Chapter 6

Analysis and Discussion of Differences Between Mixtures of Real and Synthetic Noisy Speech

6.1 Introduction

The core problem this chapter aims to address is the perceived gap in performance between separation systems that have been trained using artificially noisy training mixtures and systems that have been trained using artificial mixtures of real noisy speech. There has been a general lack of success in training effective speech separation models in which the training data is constructed using speech corpora with noise

already present in the speech signals. This prevents the use of in-domain data for training of effective speech separation systems for use in noisy environments.

There are two core objectives of this chapter:

1. To demonstrate the negative effects of using artificial mixtures constructed using real noisy speech in both the training and evaluation of speech separation systems

2. To provide an explanation as to why this data paradigm has such a negative impact on conventional single-channel speech separation

6.2 Theoretical Formulation

6.2.1 Noisy Separation Data Paradigms

The extension of the speech separation formulation to noisy speech separation is trivial in terms of input mixture, but becomes ambiguous in target. In the case where noise is present, we simply add an additional waveform $n(t)$ to the mixture:

$$x(t) = \sum_{k=1}^{K} s_k(t) + n(t). \tag{6.1}$$

It is important to note that we typically assume each signal to be independent (as we do in this work), meaning $n(t)$ cannot include the reverberation of the speech

signals, and use a separate formulation for cases with reverberation, as described in Section 3.3.

The ambiguity of noisy speech separation arises in what we consider the target output for a system in this situation. It might be natural to assume we should expect a system to simply again produce estimates of the individual speech signals $s_k(t)$, effectively removing all interfering signals. However, this implicitly requires removing noise from speech, which may be better suited to a speech enhancement system designed for denoising. Thus a target of $s_k(t) + n(t)$ may be more appropriate in terms of allowing the network to not use its modeling ability on denoising, leaving it for post-processing, something shown to be successful with the cascaded systems presented in Section 5.3. Or we may not even care what happens with the noise, allowing the output to consist of any part of $n(t)$, as long as it contains none of the other speakers.

Presently, the majority of state-of-the-art single-channel speech separation techniques rely on training objectives which encourage the estimates $\hat{s}_k(t)$ produced by the network to become closer to the ground truth speech signals $s_k(t)$ through a direct function of those signals. As a result, it is necessary to have access to the ground truth $s_k(t)$ signals during training. A consequence of this is that the networks must be trained on data with "synthetic" mixtures, meaning that the mixture waveform $x(t)$ is created by digitally summing recordings of single-speaker speech signals rather than

using recordings of naturally-occurring overlap.

Issues arise with this data paradigm when we move to the noisy speech separation domain. To still have access to the ground truth $s_k(t)$ signals, the noise must be digitally added to the mixture as well. This results in what we refer to as the "clean oracle" data paradigm for training noisy speech separation systems:

$$x(t) = \sum_{k=1}^{K} s_k^{\text{clean}}(t) + n(t) \qquad \text{clean oracle (6.2)}$$

$$\text{known signals: } \{x(t), s_1^{\text{clean}}(t), \ldots, s_K^{\text{clean}}(t), n(t)\}.$$

The majority of research conducted on noisy speech separation use this paradigm, notably works using the WHAM! [40], WHAMR! [20], and LibriMix [41] corpora.

However, there are downsides to using the clean oracle data paradigm. It requires that the speech recordings used for the data must be noise-free in the first place. This necessarily prevents in-domain training and also largely disallows the usage of any data that has been recorded outside of a recording studio, which greatly restricts the amount of training data available.

This leads to the "noisy oracle" data paradigm, in which mixtures are created using digitally-summed recordings of naturally noisy speech, in which the potential training data sources is extended to more environments and in-domain training in

terms of the speech-noise environment. However, it leads to a different formulation:

$$x(t) = \sum_{k=1}^{K} s_k^{\text{noisy}}(t) = \sum_{k=1}^{K} \left[s_k^{\text{clean}}(t) + n_k(t) \right] \quad \text{noisy oracle (6.3)}$$

$$\text{known signals: } \{x(t), s_1^{\text{noisy}}(t), \ldots, s_K^{\text{noisy}}(t)\}.$$

In this case we notably do not have access to the speech or noise signals directly, only their combination.

It is worth noting that the noisy oracle formulation has similarities with both the foundational separation formulation (2.1) and the clean oracle formulation (6.2). It resembles the former in that the mixture is a simple sum of speech signals from the same class, i.e. the $s_k^{\text{noisy}}(t)$, but it resembles the latter if we consider the sum of all $n_k(t)$ as a separate, singular noise source:

$$n_\Sigma(t) = \sum_{k=1}^{K} n_k(t). \tag{6.4}$$

In this sense, we can consider the noisy oracle data paradigm to be a variant of the regular noisy separation problem, with a type of poor annotation. Again, our situation aligns with the clean oracle formulation (6.1):

$$x(t) = \sum_{k=1}^{K} s_k^{\text{clean}}(t) + n_\Sigma(t). \tag{6.5}$$

Additionally, the impoverished targets $s_k^{\text{noisy}}(t)$ are within the bounds of an acceptable solution to the problem—they consist of the target source $s_k^{\text{clean}}(t)$ along with a subset $n_k(t)$ of the total noise signal $n_\Sigma(t)$. But, a key detail is that we have no knowledge of what part of the ground truth signal $s_k^{\text{noisy}}(t)$ is the speech that we care about, $s_k^{\text{clean}}(t)$, and what part is the undesirable-yet-acceptable noise $n_k(t)$ that could be reduced by a denoisingsystem later.

6.2.2 Noisy Oracle Paradigm Problems

Despite relative success in noisy speech separation studies using the clean oracle data paradigm, there has been less success in studies using the noisy oracle data paradigm [19] and a general lack of success of single-channel source separation in the CHiME challenges [3, 79] which restrict the use of training data, disallowing any corpora with clean speech recordings. This suggests there may be issues inherent to the noisy oracle data paradigm that must be accounted for in ways beyond simply using the same strategies as those used in clean speech separation or noisy speech separation with the "clean oracle" data paradigm.

An investigation into the exact task being asked of the system when training with the noisy oracle paradigm suggests why this may cause issues. When training a network to separate a set of noisy sources $s_k^{\text{noisy}}(t)$, we are in essence asking the network to be able to discriminate each $s_k^{\text{clean}}(t)$ and $n_k(t)$ from the remaining

$\{s_i^{\text{clean}}(t), n_i(t) \mid i \neq k\}$. In addition, the $s_k^{\text{clean}}(t)$ and $n_k(t)$ must be paired together appropriately to match the ground truth $s_k^{\text{noisy}}(t)$. There are thus two relevant issues: source separability and permutation pairing.

As each component signal is independent of the others, the discrimination can be assessed according to three categories: speech/speech separation, speech/noise separation, and noise/noise separation. The first two categories (comprising clean speech separation and speech enhancement respectively) are well-studied with widely-published baseline performance. They rely on the fact that speech and noise have separate statistical properties and additionally that speech has structure in spectral domains, with energy concentrated in very limited frequencies at any given time and following somewhat predictable trajectories, allowing for separation of speech even between voices with similar statistical properties [43]. However, noise/noise separation is most analogous to universal sound separation. This task is typically restricted to certain classes of sounds with differing statistical properties or sounds with temporal localization. Even in a work directed at separation of universal sounds presented by [80], they notably exclude ambient/environmental noises, which are not temporally localized.

In the noisy oracle separation paradigm, the noise sources are likely to be environmental, and if coming from a single corpus, may be from the same environment and thus have very similar statistical properties. This gives reason to believe that the

noise/noise separation subtask of the noisy oracle paradigm may be disproportionately difficult compared to the other two, and may be causing accordingly disproportionate harm in the training of separation networks. This is particularly undesirable, since separating each $n_k(t)$ from the total noise $n_\Sigma(t)$ is not even part of the task formulation presented in Section 3.3.

The other issue with the noisy oracle data paradigm is the issue of permutation pairing. Even if the network were capable of perfectly isolating each $s_k^{\text{clean}}(t)$ and $n_k(t)$, to perfectly produce each $s_k^{\text{noisy}}(t)$ it must find the correct permutation of $s_k^{\text{clean}}(t)$ and $n_k(t)$ that pair appropriately to match the available ground truth. If the speech and noise signals are uncorrelated according to the formulation, there should be no solution to this, and the system is being asked to solve an unsolvable problem, likely contributing to poor system performance.

We do note that the system presented by [42] shows success in addressing a nearly identical issue: training a network to produce subsets of the available ground truth and using permutation-invariant training to train the network with properly-paired sources without requiring the network to learn the pairing. While this approach could address the permutation pairing issue, it requires the sources themselves to be separable, something potentially untrue in the noisy oracle separation problem. Our initial efforts to apply their approach to this problem were not successful.

6.3 Demonstration of Problem

We conducted a set of experiments focused on demonstrating the issues relating to the noisy oracle data paradigm and how they can affect the training and evaluation of systems. All experiments use a TasNet-BLSTM [15] network trained and evaluated with SI-SDR [31] using a dataset we created [23] using the WHAM! [40] corpus, which we feel is reasonably representative of the techniques presently used in the research community, as the majority of state-of-the-art techniques are based on spectral maskingusing a TasNet-style learned basis.

6.3.1 Dataset Design

The data used for our experiments were new synthetic mixtures created using the WHAM! [40] data. While using synthetic data is not ideal, it is necessary for analyzing the variations in ground truth signals while controlling for other factors. We took the wsj0-2mix [8] mixtures consisting of clean speech from the WSJ0 dataset [18] and assigned each mixture two noise sources from the WHAM! noises, one for each source. The resulting samples can be configured for training or evaluation in a number of ways. First of all, the noises are scaled to be at a given signal-to-noise ratio (SNR) relative to their source, allowing simulations of the various SNRs present in real recordings. Secondly, the samples can be configured to mix the sources with their noise to produce two noisy samples, or they can be configured to produce two clean

Table 6.1: SI-SDR improvement [dB] comparison across networks trained on the speech-speech separation task, the speech-noise separation task, and the noise-noise separation task. The speech signals come from the wsj0-2mix corpus [8] and the noise signals are ambient recordings from restaurants, bars, and similar environments released in the WHAM! corpus [40].

Speech-Speech Separation SI-SDRi:	15.3
Speech-Noise Separation SI-SDRi:	13.4
Noise-Noise Separation SI-SDRi:	0.4

sources along with the mixture of noise, resulting in the "noisy oracle" and "clean oracle" formulations accordingly. Given the unique ability to be configured in this "noisy oracle" manner, we refer to this dataset as no-2mix.

For our experiments, we used the 16 kHz sample rate and *min* configuration of the data. We evaluated datasets created with SNRs ranging from 25 dB to -5 dB in decrements of 5 dB, as well as clean speech. We also evaluated a 'pure noise' configuration with both speech signals removed, and an 'enhancement' condition where the second speech and noise signals are removed and the first speech and noise signals are treated as the sources.

6.3.2 Separability of Noise

First, we demonstrate that separating environmental noise from noise is a considerably more difficult task than separating speech from noise or even speech from speech. In our experiments, we used the 5 SNR data configuration, removing the speech or noise and changing the targets according to the task. First, we remove the noise signals

and train the system to produce the $s_k^{\text{clean}}(t)$ from the clean speech mixture $\sum s_k^{\text{clean}}(t)$. Then, separately for each k, we train the system to produce $s_k^{\text{clean}}(t)$ and $n_k(t)$ from $s_k^{\text{clean}}(t) + n_k(t)$. And finally, we remove the speech and train the system to produce the $n_k(t)$ from $\sum n_k(t)$. The results of these experiments are shown in Table 6.1.

As we can see, the SDR improvement for noise is close to 0 compared to the high numbers in the speech-involved cases. This supports our claim that separating ambient noise is very difficult. Interestingly enough, the speech-noise separation task performs slightly worse than the speech-speech separation task, which is counter to the expectation that enhancement is easier than separation. This may be due to the fact that the speech-noise separation network is required to produce estimates of both the speech and noise, as opposed to purely speech in a true enhancement task. Noise may be difficult to estimate, and in addition my not share similar internal representations within the network in the same way multiple speech sources might.

6.3.3 Issues with Training on Data with Noisy Ground Truth

Our next sets of experiments deal with demonstrating the difference in performance between training models under the noisy oracle and clean oracle data paradigms. For these experiments we utilized the fact that our data allows training models on identical mixtures but with different ground truth training targets. We trained and evaluated models in a variety of combinations of SNR, evaluating with SI-SDR of the clean

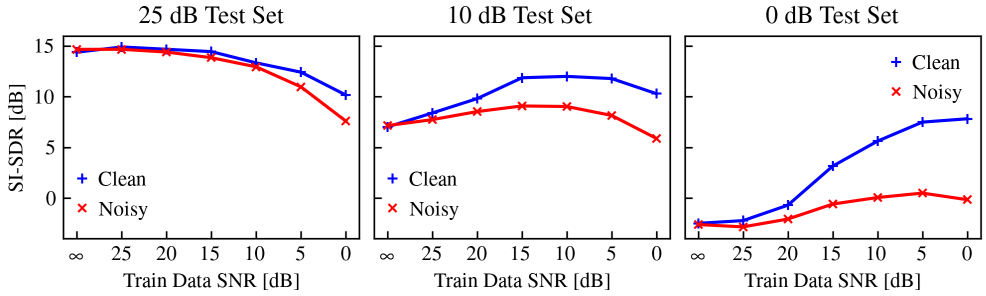


Figure 6.1: Evaluation of models trained with the SI-SDR objective with varying training data SNR using a 'clean oracle' data paradigm test set, measuring the quality of the output speech. The blue line represents models trained using data configured according to the 'clean oracle' data paradigm (equation (6.2)) while the red line represents models trained using the same mixtures with ground truth configured to the 'noisy oracle' paradigm (equation (6.3)).

speech targets. As the clean speech targets are shared across all setups, this allows a sort of "objective" quality of the output speech by different systems. While this does mean we are implicitly evaluating networks on their denoising capabilities, we consider this acceptable as the speech is the primary signal we are interested in.

Results for these experiments are shown in Figure 6.1. Each of the three plots contain the models evaluated over three conditions, with an increasing amount of noise from left to right. The horizontal axis is the amount of noise was present in the training data, with increasing noise from left to right. The difference between the two curves is that the blue line with plus markers reflect the systems trained under the 'clean oracle' paradigm and the red line with cross markers reflect the systems trained under the 'noisy oracle' paradigm. In these sets of plots, the evaluation was done using the 'clean oracle' ground truth.

There are a number of interesting trends shown by these experiments. The first is that in high-SNR evaluation sets, performance is relatively level across training data SNR. In other words, the introduction of noise into the training data has a relatively small impact on separation performance. In contrast, in low-SNR evaluation sets, there is a great gulf in performance between systems trained on high and low SNR data, demonstrating the need for the models to have encountered noisy speech in training.

However, it is also here that the differences in training data paradigm come into play. The gains in performance from training on noisy data are only present in the clean oracle paradigm. As more noise is added to the training data, the models trained with the noisy oracle paradigm show little improvement, leading to a large gap in performance on the noisy condition across both training data paradigms. This suggests that in desired applications where the task involves noise, simply adding recordings of in-domain noisy speech to the training mixtures will likely not significantly improve performance in that condition with current techniques.

6.3.4 Issues with Evaluating on Data with Noisy Ground Truth

Our final set of experiments investigating the effects of the noisy oracle data paradigm are centered on exploring the effects on performance evaluation. These experiments

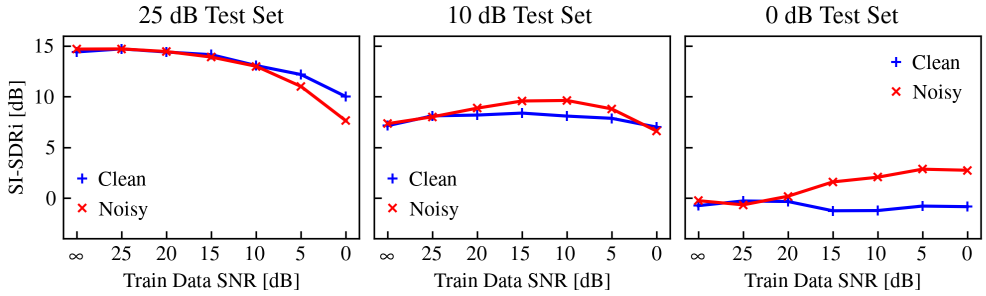


Figure 6.2: Evaluation of models trained with the SI-SDR objective with varying training data SNR using a 'noisy oracle' data paradigm test set, measuring output speech according to noisy signals. The blue line represents models trained using data configured according to the 'clean oracle' data paradigm (equation (6.2)) while the red line represents models trained using the same mixtures with ground truth configured to the 'noisy oracle' paradigm (equation (6.3)).

are similar to those in the previous subsection, except that we report SI-SDR improvement using the noisy oracle ground truth. These experiments use the exact same networks as the previous section, evaluated on the same mixtures. The only difference is the test set ground truth they are evaluated with. Here we must use SI-SDR improvement as the targets include noise and are thus not consistent across all datasets.

The results of these experiments are shown in Figure 6.2. These plots represent the same systems and data as Figure 6.1 with the sole exception that the SI-SDR values were computed using the 'noisy oracle' version of the ground truth signals rather than the 'clean oracle' versions.

The significant takeaway here is that having improper ground truth is not simply reflected by a shift in results, but an entire change of message. We know from the clean

oracle evaluation that the models trained with data using the clean oracle paradigm perform significantly better than those trained using the noisy oracle paradigm; however, this trend is not reflected while evaluating with the noisy oracle evaluation set, where the noisy oracle trained models fare better than the clean oracle ones. This suggests that not only is the noisy oracle data paradigm harmful for training models, but even has an impact in the evaluation of models, leading to even incorrect *ranking* among systems for the desired task.

6.4 Conclusion

We have demonstrated that there are significant consequences resulting from the use of two different types of ground truth data for noisy speech separation. A theoretical analysis suggests ways in which this could impact system training and evaluation. In addition, we have constructed data to experimentally validate our hypotheses and provided evidence that the ground truth paradigm can impact the performance of systems trained using that data as well as the evaluation of systems.

Chapter 7

Training Speech Separation Systems on Mixtures of Real Noisy Speech

7.1 Introduction

In this chapter we develop a technique aimed at closing the gap in performance between systems trained on mixtures of real noisy speech and systems trained on noisy mixtures where the noise has been synthetically added, based on the analysis presented in Chapter 6. Our approach is based on attempting to avoid the downsides created by the implicit requirement of noise separation resulting from the application of the SI-SDR waveform-level objective on noisy targets. As a result, we aim to develop a function that can minimize the contributions of any residual noise to the

training loss while still penalizing the network for failing to separate speech.

7.2 Proposed Solution

Our core approach to improving performance of systems trained on the noisy oracle data paradigm is to rely on the fact that ambient noise is harder for networks to separate than speech is. Though formulated as separation problem, the core noisy oracle task—producing estimates of the unknowable $s_k^{\text{clean}}(t)$ from a mixture of $s_k^{\text{noisy}}(t)$—is essentially a semi-supervised speech enhancement and separation task. The goal is to remove the noise from the speech and separate the speech without any actual annotation of what speech or noise signals look like, only their combination. We are thus restricted to using the information we do have access to—in this approach the fact that the implicit noise separation subtask in the traditional noisy oracle formulation is considerably more difficult than the speech separation subtask, as demonstrated in Section 6.3.2.

The approach used to utilize this information is a modified objective function, which introduces a kind of "no attempt" case to the function to complement the simple measure of output correctness in a traditional objective function. The motivation is to create three sets of categories that components of the mixture will fall into according to the type of data and network's capabilities:

1. correctly separating a component of the signal should make the most positive contribution to the objective function;

2. not attempting to separate a component of the signal should make an intermediate contribution to the objective function;

3. incorrectly separating a component of the signal should make the most negative contribution to the objective function.

Through this breakdown, the network will be encouraged to implicitly partition the network into speech and noise when optimizing this objective function. The network is largely incapable of separating noise, so it is better to make no attempt at separation than make an attempt and (inevitably) get it wrong. However, speech is generally separable so the network is incentivized to attempt to separate the speech.

This approach is implemented through an additional output to the separation network for a noise estimate $\hat{n}_\Sigma(t)$, serving as the "no attempt" category. We then apply a discount in the objective function on the component of separation errors that have been identified *a priori* by the network as noise.

7.2.1 Theoretical Approach

The primary technical consideration of the approach is how to isolate and interact with sub-components of waveforms, which we never have direct access to. For example, we

want our objective function for the estimate $\hat{s}_k(t)$ of a particular source to be invariant to any residual $s_k^{\text{noisy}}(t) - \hat{s}_k(t)$ that is contained within $n_k(t)$, provided that the $n_k(t)$ has also been correctly estimated within $\hat{n}_\Sigma(t)$. However, we never have access to $n_k(t)$ directly or even an estimate $\hat{n}_k(t)$ of it—we only know that it belongs to our ground truth $s_k^{\text{noisy}}(t)$ and unknown estimate target $n_\Sigma(t)$.

The core assumption we rely on is that audio signals can be considered to be zero-mean random processes. As a consequence, as long as two sources are independent, when treating the signals as vectors, their dot product will be approximately zero and they are thus approximately orthogonal. As such, we will shift notation from functional form $x(t)$, $s_k(t)$, $n_k(t)$, etc. to vector form $\mathbf{x}$, $\mathbf{s}_k$, $\mathbf{n}_k$, etc. $\in \mathbb{R}^T$. More precisely, for a given mixture $\mathbf{x}$, we can define the set $\mathcal{P}_\mathbf{x}$ of independent components

$$\mathcal{P}_\mathbf{x} \coloneqq \{\mathbf{s}_1^{\text{clean}}, \mathbf{n}_1, \ldots, \mathbf{s}_K^{\text{clean}}, \mathbf{n}_K\}, \tag{7.1}$$

which according to the orthogonality assumption have the property

$$\langle \mathbf{a}, \mathbf{b} \rangle = 0 \quad \forall \mathbf{a}, \mathbf{b} \in \mathcal{P}_\mathbf{x},\ \mathbf{a} \neq \mathbf{b}. \tag{7.2}$$

It is worth noting that the above property and result is why this approach does not apply to reverberation, as the reverberations of a given source cannot be considered to be independent from that source.

Table 7.1: Statistics of angle between vector representation of mixture components, measured as deviations from the expected 90° angle that would result from true orthogonality.

	Speech-Speech			Speech-Noise			Noise-Noise		
Split	Mean	Std.	Abs. Max	Mean	Std.	Abs. Max	Mean	Std.	Abs. Max
tr	0.0°	0.6°	5.1°	0.0°	0.4°	5.4°	0.0°	1.7°	49.2°
cv	−0.0°	0.6°	4.4°	0.0°	0.5°	7.0°	0.0°	1.0°	8.2°
tt	0.0°	0.5°	2.4°	0.0°	0.4°	2.6°	0.0°	1.4°	12.5°
All	0.0°	0.6°	5.1°	0.0°	0.4°	7.0°	0.0°	1.6°	49.2°

In addition, we verified how true the orthogonality assumptions are within the data used for our experiments, shown in Table 7.1. This table shows statistics of deviations from 90° within the dataset between sources within a mixture. The rows are each subset of the data, as well as the full dataset. For the speech-speech, speech-noise, and noise-noise comparisons we report both the mean and standard deviation of the deviation of the angle from 90°, as well as the magnitude of the largest deviation. While the noise-noise comparisons have some concerning very large outliers, on average the property is generally true.

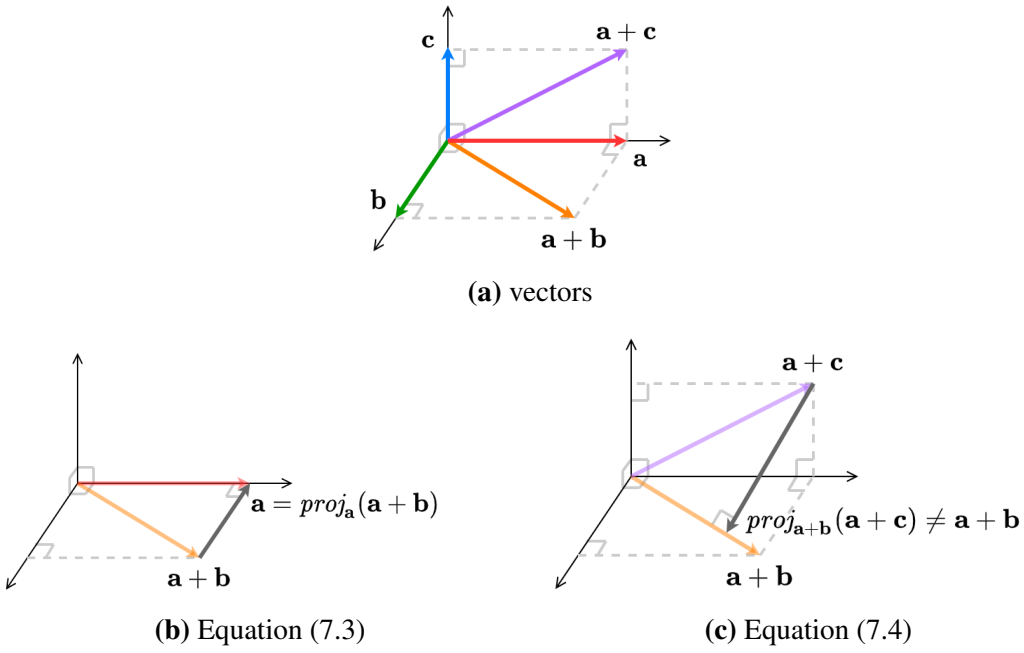
The property above leads to the following projection results for any $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c} \in \mathcal{P}_{\mathbf{x}}$:

$$proj_{\mathbf{a}}(\mathbf{a}+\mathbf{b}) = \mathbf{a} \tag{7.3}$$

$$proj_{\mathbf{a}+\mathbf{b}}(\mathbf{a}+\mathbf{c}) = \frac{\|\mathbf{a}\|^2}{\|\mathbf{a}+\mathbf{b}\|^2}(\mathbf{a}+\mathbf{b}). \tag{7.4}$$

Geometric interpretations of these equations are shown in Figure 7.1.

Equation 7.3 is of particular interest to our solution, as the traditional error of our



(a) vectors

(b) Equation (7.3)

(c) Equation (7.4)

Figure 7.1: Demonstration of vector projections.

target signal lies within the noise signal, i.e.:

$$\mathbf{s}_k^{\text{noisy}} - \mathbf{s}_k^{\text{clean}} = \mathbf{n}_k. \tag{7.5}$$

Thus in this ideal case, projecting the noise mixture onto the error recovers that error:

$$\begin{aligned} proj_{(\mathbf{s}_k^{\text{noisy}} - \mathbf{s}_k^{\text{clean}})} \mathbf{n}_\Sigma &= proj_{\mathbf{n}_k} \mathbf{n}_\Sigma \\ &= \mathbf{n}_k \\ &= \mathbf{s}_k^{\text{noisy}} - \mathbf{s}_k^{\text{clean}}. \end{aligned} \tag{7.6}$$

This leads us to the formulation of the core of our training objective, an error term that is reduced by errors deemed acceptable:

$$(\mathbf{s}_k^{\text{noisy}} - \hat{\mathbf{s}}_k) - proj_{(\mathbf{s}_k^{\text{noisy}} - \hat{\mathbf{s}}_k)} \hat{\mathbf{n}}_\Sigma. \tag{7.7}$$

This equation represents the separation error $\mathbf{s}_k^{\text{noisy}} - \hat{\mathbf{s}}_k$ discounted by the amount of error that is within the estimate $\hat{\mathbf{n}}_\Sigma$ of the noise. In other words, if the estimate fails to separate the noise, it does not contribute to the overall error as long as it has been identified within the noise estimate.

7.2.2 Objective Function

We present two iterations of our proposed objective function, called Estimated Source-to-Separation Error Ratio (ESSER). Similar to SDR and SNR objectives, it is based on an energy ratio in decibels of the target signal (the noise-free 'estimated source') to the error term (the divergence from the ground truth caused by the parts of the signal that we are trying to separate).

7.2.2.1 ESSER Objective

The original ESSER [23] formulation is as follows:

$$\text{ESSER}_{\lambda}(\hat{\mathbf{s}}_k, \hat{\mathbf{n}}_{\Sigma}) := 10\log_{10} \frac{\|\hat{\mathbf{s}}_k\|^2}{\|(\mathbf{s}_k^{\text{noisy}} - \hat{\mathbf{s}}_k) - \lambda * proj_{(\mathbf{s}_k^{\text{noisy}} - \hat{\mathbf{s}}_k)} \hat{\mathbf{n}}_{\Sigma} + proj_{\hat{\mathbf{s}}_k} \hat{\mathbf{n}}_{\Sigma}\|^2}. \quad (7.8)$$

The core motivations behind this objective function is to encourage the network to partition the signal $\mathbf{x}$ into two subsignals—one that is separated and scored accordingly ($\mathbf{s}_k^{\text{separated}}$) and one that is not separated and only partially contributes to the overall error ($\mathbf{s}_k^{\text{non-sep}}$).

If the signal is not included in the non-separated category $\hat{\mathbf{n}}_{\Sigma}$, the projection discount term falls out and the objective resembles the standard SI-SDR [31] objective:

$$\approx 10\log_{10} \frac{\|\hat{\mathbf{s}}_k\|^2}{\|(\mathbf{s}_k^{\text{separated}} - \hat{\mathbf{s}}_k)\|^2}. \quad (7.9)$$

In contrast, if the signal is included in the non-separated category $\hat{\mathbf{n}}_{\Sigma}$, the projection is equivalent to the error, and is weighted down according to the parameter λ:

$$\approx 10\log_{10} \frac{\|\cdot\|^2}{\|\mathbf{s}_k^{\text{non-sep.}} - \lambda * proj_{\mathbf{s}_k^{\text{non-sep.}}} \hat{\mathbf{n}}_{\Sigma}\|^2} = 10\log_{10} \frac{\|\cdot\|^2}{\|(1-\lambda)\mathbf{s}_k^{\text{non-sep.}}\|^2}. \quad (7.10)$$

And finally, the $proj_{\hat{\mathbf{s}}_k} \hat{\mathbf{n}}_{\Sigma}$ term in the denominator is to introduce a penalty for components of the mixture appearing in both the source and noise estimates, violating the

desired "partition" property.

The overall hope is that for speech, the denominator term of (7.9) can get close to zero (close enough to achieve a SNR of roughly 15 dB for the system reported in Table 6.1 of this paper). This is better than the "flat discount" of $(1-\lambda)$ given by (7.10). In contrast, for inseparable noise, the denominator term of (7.9) is unlikely to have any canceling out of the source from the estimate $\hat{\mathbf{s}}_k$, resulting in the flat discount of (7.10) being a better option. As a result, during training, the network will learn to put inseparable parts of the signal in the noise estimate and learn to separate the remaining part of the signal.

7.2.2.2 ESSER Issues

The core issues of the ESSER objective stem from the general instability of training a network using this function with respect to the parameter λ. As noted in [23], tuning of parameters related to the loss can be very difficult, as we are tuning the parameters for performance on a task that we do not have any ground truth for—while our goal is performance akin to the clean oracle paradigm, our validation sets will also be restricted to the noisy oracle paradigm. As such, it is particularly undesirable to have a parameter that is difficult to tune and varies from dataset to dataset. Even in [23], it is noted that the heuristic used to select parameters did not work in all cases. In Section 7.4.2 we present reults showing that ESSER performance suffers significantly

with improper parameters as well.

The ESSER objective can be thought of as having entire classes of solutions rather than a single optimum. For example, in the case of $\lambda = 1.0$, there are infinitely many solutions that attain an infinite ESSER value—every partitioning of the mixture where the separated portion of the signal has been separated perfectly. This includes both extremes of perfect separation and putting the entire signal into the non-separated category. As λ is decreased, this class of solutions does not all have infinite value; only perfect separation attains an infinite value, otherwise decreasing as more and more of the signal is put into the non-separated category. The ideal solution to our problem, of course, is one somewhere between the two extremes, as we want speech in the separated category and noise in the non-separated category.

The problem is that in practice, the gradients are pulling the network to both extremes and λ must be tuned to balance those gradients. The network can very easily fall into a hole on one of the two extremes, with little chance of getting out. The parts of the loss represented by (7.9) result in gradients pulling the entire $\mathbf{s}_k^{\text{noisy}}$ into the $\hat{\mathbf{s}}_k$, and the parts of the loss represented by equation (7.10) pull the entire mixture into the $\hat{\mathbf{n}}_\Sigma$ estimate. It is thus very difficult to tune a single parameter λ to balance the strength of those gradients such that the former are stronger for speech and the latter are stronger for noise.

Additionally, we empirically found that the noise estimate generally contained

energy in spectral regions dominated by speech, suggesting that the approach used to ensure the network produced a strict partition of the signal was insufficient for that goal.

7.2.2.3 ESSER2 Objective

The second iteration of the objective function that we introduce, called ESSER2, is an extension of the principles behind the original ESSER objective to produce an objective that is more robust in training and can more reliably produce a network that performs better than the baseline on the task while training on data using the noisy oracle paradigm.

The primary component of the ESSER2 objective function is a discounted source-to-error ratio (DSER) of the estimated source, defined as follows:

$$\text{DSER}_{\lambda}(\mathbf{e}_k) \coloneqq 10\log_{10}\frac{\|\hat{\mathbf{s}}_k\|^2}{\|\mathbf{e}_k - \lambda * proj_{\mathbf{e}_k}\hat{\mathbf{n}}_{\Sigma}\|^2}. \tag{7.11}$$

This is a function of an estimated source ($\hat{\mathbf{s}}$), an error ($\mathbf{e}$) associated with that source estimate (which is simply ($\mathbf{s}^{\text{noisy}} - \hat{\mathbf{s}}$) in the typical ESSER/SI-SDR case), and an estimate ($\hat{\mathbf{n}}$) of non-separated components that is used to discount the error according to a parameter λ. The motivation for this equation is covered in Section 7.2.2.1.

DSER differs from the ESSER equation in that it is missing the partition constraint,

which is addressed in ESSER2 in a different manner. The partition constraint is handled through the averaging of two DSER terms using different error terms:

$$\frac{1}{2}\left[\text{DSER}_{\lambda}(\mathbf{s}_k^{\text{noisy}} - \hat{\mathbf{s}}_k) + \text{DSER}_{\lambda}((\hat{\mathbf{s}}_k + \hat{\mathbf{n}}_{\Sigma}) - \mathbf{s}_k^{\text{noisy}})\right]. \tag{7.12}$$

The two averaged DSER terms differ only in what they use as error. The first, more traditional term, is $\mathbf{s}_k^{\text{noisy}} - \hat{\mathbf{s}}_k$ and is the "left out" error—the parts of the ground truth signal we did not include in our separation estimate. In the ideal case, this resulting error will be $\mathbf{n}_k$, i.e. the part of the ground truth source that we do not want the network to separate and as a result should be discounted. The second error term, $(\hat{\mathbf{s}}_k + \hat{\mathbf{n}}_{\Sigma}) - \mathbf{s}_k^{\text{noisy}}$, is complementary and represents the "total coverage" error. In this case, we seek to fully account for the ground truth $\mathbf{s}_k^{\text{noisy}}$ by adding in the noise to our estimate, but overshoot due to not having an isolated estimate of $\mathbf{n}_k$. And, the overshoot contributing to the error discourages anything included in $\hat{\mathbf{s}}_k$ or $\hat{\mathbf{n}}_{\Sigma}$ beyond the minimum of what is required to reconstruct $\mathbf{s}_k^{\text{noisy}}$.

In the ideal solution, both error terms result in a perfect target/non-target partition

of the noise mixture:

$$\mathbf{s}_k^{\text{noisy}} - \mathbf{s}_k^{\text{clean}} = \mathbf{n}_k; \tag{7.13}$$

$$(\mathbf{s}_k^{\text{clean}} + \mathbf{n}_\Sigma) - \mathbf{s}_k^{\text{noisy}} = \sum_{j \neq k} \mathbf{n}_j; \tag{7.14}$$

$$\left[\mathbf{s}_k^{\text{noisy}} - \mathbf{s}_k^{\text{clean}}\right] + \left[(\mathbf{s}_k^{\text{clean}} + \mathbf{n}_\Sigma) - \mathbf{s}_k^{\text{noisy}}\right] = \mathbf{n}_\Sigma. \tag{7.15}$$

The other addition included in ESSER2 is the inclusion of a regularizer term to keep the network from falling into one of the two extremes of categorizing the mixture as entirely speech or entirely noise. For this, we add in a weighted mean squared error term between the signal-to-noise ratio (SNR) of the estimates and the *a priori* estimated training data SNR:

$$\text{MSE}_{\text{SNR}_{\text{data}}}(\hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_K, \hat{\mathbf{n}}_\Sigma) \coloneqq (\min(\text{SNR}_{\text{data}}, 20) - \text{SNR}_{\text{est}})^2. \tag{7.16}$$

In this equation, the dataset SNR is clamped at 20 dB, serving as a maximum clean SNR, above which it is not worth trying to achieve. The SNR of the estimates is computed as follows:

$$\text{SNR}_{\text{est}} \coloneqq 10 \log_{10} \frac{\sum_{k=1}^{K} \|\hat{\mathbf{s}}_k\|^2}{\|\hat{\mathbf{n}}_\Sigma\|^2}. \tag{7.17}$$

It must be computed over all estimated sources at once, as we do not have access to estimates of the individual $\mathbf{n}_k$.

This brings us to our final ESSER2 formulation, parametrized by λ_m, λ_r, and SNR_{data}:

$$\text{ESSER2}(\hat{\mathbf{s}}_k, \hat{\mathbf{n}}_\Sigma) \coloneqq \frac{1}{2}\left[\text{DSER}_{\lambda_m}(\mathbf{s}_k^{\text{noisy}} - \hat{\mathbf{s}}_k) + \text{DSER}_{\lambda_m}((\hat{\mathbf{s}}_k + \hat{\mathbf{n}}_\Sigma) - \mathbf{s}_k^{\text{noisy}})\right]$$

$$+ \lambda_r * \text{MSE}_{\text{SNR}}(\hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_K, \hat{\mathbf{n}}_\Sigma). \tag{7.18}$$

This function is evaluated on a per-source basis, using the definitions of DSER and MSE defined in (7.11) and (7.16), and is parametrized by three hyperparameters: λ_m, the parameter controlling the mitigation of error from the non-separated part of the signal; λ_r, the parameter weighing the regularizer term; and SNR_{data}, an estimate of the oracle $\mathbf{s}_k^{\text{clean}}$-to-$\mathbf{n}_k$ SNR of the training data. While this does add two more hyperparameters than the initial ESSER objective, the trade-off is that they are much more stable and data-invariant—with the exception of the dataset SNR, which benefits from corresponding to a real, estimate-able quantity, and in practice has shown signs of being very forgiving in terms of tolerance as well.

7.2.2.4 Considerations of Scaling

While not inherently a part of the ESSER objectives, the issue of signal scaling is a necessary consideration, as the ESSER objectives rely on relative signal magnitudes, and TasNet-based separation methods (representing the majority of state-of-the-art systems, including that of this work) and others produce waveforms that do not adhere to any particular signal scale. This problem is not trivially solved, so we present the approach we used.

We use an approach and assumptions comparable to those used in SI-SDR [31], namely projection operations combined with the assumption that the error is orthogonal to the source, similar to our approach to the core problem that was presented in Section 7.2.1. In a typical separation paradigm, we have a ground truth signal $\mathbf{s}^{\text{clean}}$ and an arbitrarily-scaled estimate of that signal $\tilde{\mathbf{s}} = \alpha(\mathbf{s}^{\text{clean}} + \mathbf{e})$ that consists of a scaled version of the target signal plus some orthogonal noise signal. Exploiting the projection operation formulated in (7.3), we can scale $\tilde{\mathbf{s}}$ by β such that the projection

of $\beta\tilde{\mathbf{s}}$ onto $\mathbf{s}^{\text{clean}}$ is itself $\mathbf{s}^{\text{clean}}$:

$$proj_{\mathbf{s}^{\text{clean}}}\beta\tilde{\mathbf{s}} = \mathbf{s}^{\text{clean}} \tag{7.19}$$

$$proj_{\mathbf{s}^{\text{clean}}}\beta\alpha(\mathbf{s}^{\text{clean}} + \mathbf{e}) = \mathbf{s}^{\text{clean}}$$

$$\beta\alpha\mathbf{s}^{\text{clean}} = \mathbf{s}^{\text{clean}}$$

$$\beta\alpha = 1. \tag{7.20}$$

Thus with this method, the resulting value of β cancels out the coefficient α on the pre-scaled $\tilde{\mathbf{s}}$ and we achieve optimal scaling.

This approach does not work, however, in the noisy oracle data paradigm, as the comparable projection would be that of (7.4), not (7.3), as there is an additional orthogonal component in the ground truth. While the output estimate can be formulated in the same manner, $\tilde{\mathbf{s}} = \alpha(\mathbf{s} + \mathbf{e})$, the ground truth $\mathbf{s}^{\text{noisy}} = \mathbf{s}^{\text{clean}} + \mathbf{n}$ now has the

non-target noise included. The comparable strategy does not work:

$$proj_{\mathbf{s}^{\text{noisy}}} \beta \tilde{\mathbf{s}} = \mathbf{s}^{\text{noisy}} \tag{7.21}$$

$$proj_{\mathbf{s}^{\text{clean}}+\mathbf{n}} \beta\alpha(\mathbf{s}^{\text{clean}} + \mathbf{e}) = \mathbf{s}^{\text{clean}} + \mathbf{n}$$

$$\frac{\beta\alpha\|\mathbf{s}^{\text{clean}}\|^2}{\|\mathbf{s}^{\text{clean}} + \mathbf{n}\|^2}(\mathbf{s}^{\text{clean}} + \mathbf{n}) = \mathbf{s}^{\text{clean}} + \mathbf{n}$$

$$\beta\alpha = \frac{\|\mathbf{s}^{\text{clean}} + \mathbf{n}\|^2}{\|\mathbf{s}^{\text{clean}}\|^2}, \tag{7.22}$$

and thus the signal is over-scaled. We can instead take the alternate strategy of scaling $\tilde{s}$ such that the projection of the ground truth onto the scaled estimate is the scaled estimate:

$$proj_{\beta\tilde{\mathbf{s}}} \mathbf{s}^{\text{noisy}} = \beta\tilde{\mathbf{s}} \tag{7.23}$$

$$proj_{\beta\alpha(\mathbf{s}^{\text{clean}}+\mathbf{e})}(\mathbf{s}^{\text{clean}} + \mathbf{n}) = \beta\alpha(\mathbf{s}^{\text{clean}} + \mathbf{e})$$

$$\frac{\beta\alpha\|\mathbf{s}^{\text{clean}}\|^2}{\beta^2\alpha^2\|\mathbf{s}^{\text{clean}} + \mathbf{e}\|^2}\beta\alpha(\mathbf{s}^{\text{clean}} + \mathbf{e}) = \beta\alpha(\mathbf{s}^{\text{clean}} + \mathbf{e})$$

$$\frac{\|\mathbf{s}^{\text{clean}}\|^2}{\|\mathbf{s}^{\text{clean}} + \mathbf{e}\|^2} = \beta\alpha. \tag{7.24}$$

In this case, the signal is under-scaled.

Though neither solution is optimal, we use the strategy reflected in (7.23), and in

our work the estimates $\hat{\mathbf{s}}_k$ and $\hat{\mathbf{n}}_\Sigma$ have been scaled using this strategy. In the case of overscaling reflected in (7.22), the strategy has the upside of being invariant to the reconstruction error. However, the scaling is never correct, and cannot be compensated for without knowledge of the relative magnitudes of $\mathbf{s}^{\text{clean}}$ and $\mathbf{n}$. In contrast, the underscale case reflected in (7.24) will converge to optimal as the error approaches zero, though varying with respect to the signal error.

7.3 Experimental Configuration

7.3.1 Data

The data used in our work is the dataset described in Section 6.3.1, consisting of the wsj0-2mix [8] with added noises from the WHAM! [40] corpus, where each mixture consists of two sources and two noises, and can be configured in either the clean oracle or noisy oracle paradigm. In addition, the source-to-noise SNR can be configured as a parameter. For all of our experiments, we used the 16 kHz sample rate and 'min' wsj0-2mix configuration. We evaluated data with SNR configurations ranging from 25 dB to -5 dB in increments of 5 dB as well as noiseless 'clean' condition.

7.3.2 Models

For all of our experiments we used a standard TasNet-BLSTM [15] with four Bi-directional Long Short-Term Memory (BLSTM) layers with 600 units in each direction. For the analysis and synthesis bases, we used 500 filters of length 5 ms with a shift of 2.5 ms, with ReLU and LayerNorm applied to the analysis features.

We feel this setup is representative of state-of-the-art methods, as most modern architectures are based on a TasNet design [42, 81, 82], with nearly all exceptions using an SDR training objective [32].

7.3.3 Training

All networks were trained with either negative SI-SDR [31], negative ESSER, or negative ESSER2 loss using an utterance-level permutation invariant manner [11]. Models were trained using batches of 4-second random segments from each sample. We used the Adam algorithm [64] using an initial learning rate of 0.001, decreasing the learning rate by a factor of two if the validation loss fails to reach a new minimum for three consecutive epochs. In addition, we applied gradient clipping using a maximum ℓ_2 norm of 5.

7.3.4 Evaluation

For all experiments we evaluated the results using SI-SDR. In some cases we report the raw SI-SDR value and in some cases we report SI-SDR improvement (SI-SDRi).

The typical values reported in speech separation are SI-SDR improvement, which is the difference between the SI-SDR value from the system output and the SI-SDR value where the input mixture is evaluated as the estimate. In some sense this is a measure of how much better the estimate is, or how much the network has accomplished, which is a natural metric to report as a result.

In some cases we report the raw SI-SDR value instead. One reason this is not typically reported is that it depends on the reference waveform, meaning that it is not directly comparable across evaluation sets. This issue is addressed by the fact that across many configurations of our data, the sources are consistent—meaning that regardless of the dataset SNR or if it is trained with noisy oracle or clean oracle paradigm, as long as the evaluation set is the clean oracle paradigm, the reference target is identical and all values can be directly compared. The metric gives us a sense of the objective quality of the system output, i.e. how it sounds. This can also help decouple the effect caused by the fact that evaluating SI-SDRi on noisy speech will include improvements made through both denoising and separation.

An additional evaluation metric we use is Perceptual Evaluation of Speech Quality (PESQ) [38]. This is another full-reference metric which needs the ground truth

Table 7.2: Performance comparison across training objectives and ground truth paradigms with identical mixtures. The SI-SDR system trained with noisy oracle ground truth sources serves as a performance floor, while the clean oracle ground truth source SI-SDR system serves as a performance ceiling.

Datset	SI-SDR		ESSER	ESSER2	ESSER2
SNR [dB]	Noisy	Clean	Noisy	Noisy	Noise
∞	15.3	15.0	15.3	15.0	–
15.0	11.9	13.5	11.9	11.1	−5.5
10.0	9.0	12.0	9.0	**9.2**	**5.1**
5.0	5.0	10.4	5.7	**5.7**	**2.4**
0.0	−0.1	7.8	0.8	**1.4**	−0.2
−5.0	−9.0	3.5	−9.3	−11.0	−3.8
	Separation SI-SDR [dB]				Noise Estimate SI-SDRi [dB]

Parameters	
λ_m:	0.1
λ_r:	0.1
SNR_{data}:	oracle

waveform. PESQ was designed to model human listening tests of voice quality, reporting a Mean Opinion Score (MOS) ranging from 1 to 5, with 1 being the lowest and 5 being the best. As such, it is primarily used for speech enhancement tasks, but can be reasonably used in speech separation tasks, particularly in this case where the presence of noise is of interest. As such, we report PESQ to further validate our findings.

7.4 Results and Discussion

7.4.1 System Performance on Core Task

The results of our primary experiments are reflected in Table 7.2. In these experiments, we evaluate models trained and tested on the same dataset SNR, computing the raw SI-SDR value using the clean oracle sources. The SI-SDR columns represent the baseline systems, being trained with the regular SI-SDR objective, but with both training data paradigms. The 'noisy' column represents the baseline case of training a network normally, while the 'clean' column represents a performance ceiling, i.e. the best we could reasonably expect a network to do, as that network is given perfect signal information. The ESSER2 'noisy' column shows the results of our ESSER2 system. In addition, we report the SI-SDRi improvement of the noise estimate, which gives us an idea of how close our noise estimate $\hat{\mathbf{n}}_{\Sigma}$ is to the true noise signal $\mathbf{n}_{\Sigma}$. For these experiments we used a value of 0.1 for λ_m and λ_r, and using the oracle dataset SNR for the SNR_{data} parameter.

The system shows modest gains over the baseline in the 10, 5, and 0 dB conditions. It is unsurprising that the system would fail to make improvements in very low noise conditions, as this method is focused on addressing degradation due to the presence of noise. In addition, the system fails completely, performing worse than the baseline, in the -5 dB condition. While disappointing, it is worth noting that even the performance

Table 7.3: Performance comparison similar to Table 7.2 with results in PESQ

Datset	SI-SDR		ESSER	ESSER2
SNR [dB]	Noisy	Clean	Noisy	Noisy
∞	3.3	3.2	3.2	3.2
15.0	2.7	3.0	2.7	2.7
10.0	2.5	2.8	2.5	2.4
5.0	2.1	2.6	2.2	1.1
0.0	1.7	2.3	1.7	**1.8**
−5.0	1.3	1.8	1.3	**1.5**

ceiling is performing very poorly in this condition. It is also worth noting that this is the first condition in which the noise contains more energy than the speech, which may be causing issues with the objective function.

The results of the ESSER2-trained system are comparable to those of the original ESSER system [23]. Though peak performance has not improved compared to ESSER, the ESSER2 objective does provide multiple benefits over the original objective function. As we will demonstrate in the following section, the ESSER loss function is exceptionally fragile with respect to its λ parameter, while the ESSER2 function is quite robust with respect to its parameters. In addition, as shown in Figure 7.2, the system outputs more closely resemble the target signals in the spectral domain.

The results of these systems evaluated with PESQ are shown in Table 7.3. The overall trends are fairly consistent to the SI-SDR results, with a few exceptions. One notable outlier is that the performance of the ESSER2 system in the 5 dB noise condition appears to have failed, with a MOS of only 0.1 higher than the minimum, lower than the traditional SI-SDR–trained system baseline. This type of result would

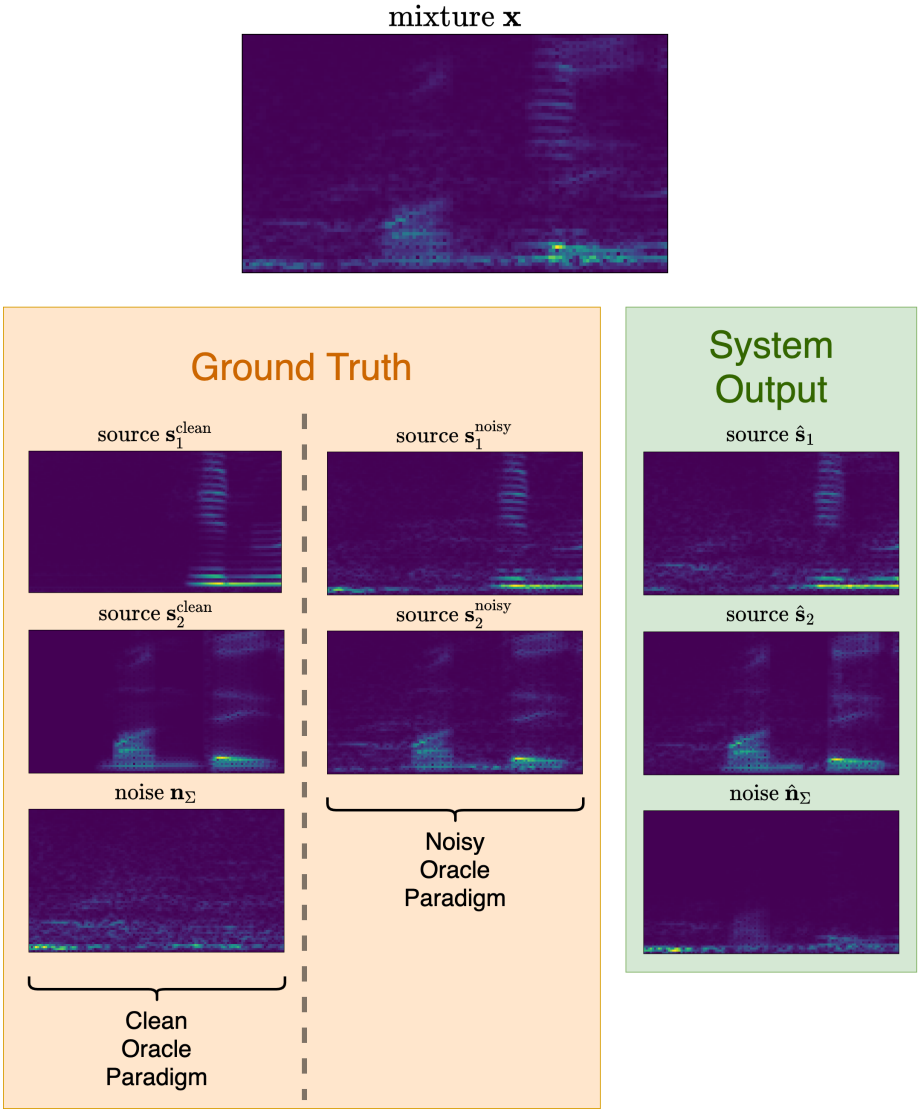


Figure 7.2: Sample section of magnitude spectra from the 5 dB evaluation set comparing ESSER2 system output to the oracle signals. The box on the left shows an example of what the ground truth signals look like across the two data paradigms. The box on the right is an example of real system output on this mixture. This system has been trained on data with ground truth following the noisy oracle paradigm but is trying to produce outputs consistent with the clean oracle paradigm. We draw particular attention to the bottom left corner of the plots for $\mathbf{n}_\Sigma$, $\mathbf{s}_1^{\text{noisy}}$, and $\hat{\mathbf{n}}_\Sigma$, where a relatively high-energy portion of noise is present. The system has successfully identified this as noise, despite being trained on data where this type of signal was merely included in a source signal similar to $\mathbf{s}_1^{\text{noisy}}$, without ever being explicitly annotated as noise.

typically indicate a model that failed to train, yet performance measured in SI-SDR showed improvement. Another interesting result is that in the -5 dB test condition the ESSER2 system shows some improvement. While the baseline systems still are performing quite poorly, the PESQ results do not show the same system failures that SI-SDR does.

7.4.2 Analysis of Parameter Robustness

We performed a number of experiments analyzing the robustness of the ESSER and ESSER2 objectives with respect to their hyperparameters. This is a particularly sensitive issue with tasks such as this, where the desired output of our system cannot be directly compared to the ground truth that we have access to. As noted in Section 6.3.4, evaluation of systems using data under the noisy oracle paradigm is not appropriate for estimating performance of a system for the true task we care about. This can be a serious issue for hyperparameter tuning, as parameters cannot be appropriately tuned using a held-out set from the training data. This problem is handled in the original ESSER function through a proxy function used on the validation set, but it fails to produce optimal results in multiple reported conditions.

Unfortunately, as the ESSER and ESSER2 systems use different hyperparameters, we cannot directly compare their sensitivity to any given parameter's value. Nevertheless, it is worth evaluating how stable the network training is with respect to the

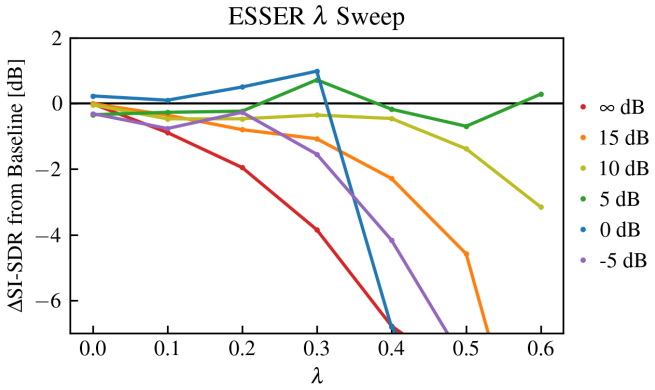


Figure 7.3: Plot of improvement over baseline as a function of λ parameter for original ESSER loss across multiple conditions. We show that there is no value that is consistently good, and incorrect values can perform significantly worse than the baseline.

parameters, particularly since the improvements of the ESSER2 function are primarily for decreased sensitivity to parameter values.

One reason this is so critical, as discussed by [23], is due to the inability to tune parameters according to a held-out set. One of the fundamental aspects of this work is to train and evaluate in conditions where we inherently do not have appropriate ground truth. As shown in Section 6.3.4, the issues in ground truth are relevant to evaluation as well as training, so using a held-out set would be inappropriate, as it would suffer from the same issues faced in evaluation. The work by [23] uses a heuristic to tune the λ parameter, noting that the heuristic failed in some cases. In this work, we instead worry less about tuning optimal performance of the system, and instead focus on demonstrating the relative invariance of performance with respect to parameter values, removing the need to develop techniques to tune parameters for new conditions.

Figure 7.3 shows the results of a parameter sweep over systems trained using the

ESSER objective. This plot shows the improvement over baseline as a function of λ, with each line representing a different condition. The horizontal axis contains the sweep over λ and the vertical axis contains the improvement over the baseline, with a horizontal black line at the baseline value. Each curve is a different noise condition. These systems were trained with negative ESSER loss, training on noisy oracle data and evaluated on the matched-SNR test set with clean oracle for evaluation. To an extent, it is difficult to make out any trend in performance, which is one point. The fact that there is no consistent trend, along with the fact that performance can degrade catastrophically with poorly-chosen parameters, means that training a system on unknown data would be unlikely to be successful.

However, there is some evidence of λ acting appropriately. The systems perform at approximately baseline performance when $\lambda = 0$, which is equivalent to turning off the error discount. Increasing λ increases the noise discount, encouraging the network to put more and more of the signal in the noise estimate. For mixtures with lower SNRs, this can improve performance, but for all networks, at some point the discount becomes too strong and the performance falls apart completely, likely due to not attempting to separate enough of the signal.

Figure 7.4 shows a sweep of the comparable parameter in ESSER2, λ_m, on the 5 and 0 dB test sets. The performance here is considerably more stable, with performance only starting to decrease at $\lambda_m = 0.4$ and $\lambda_m = 0.5$, and needing to be

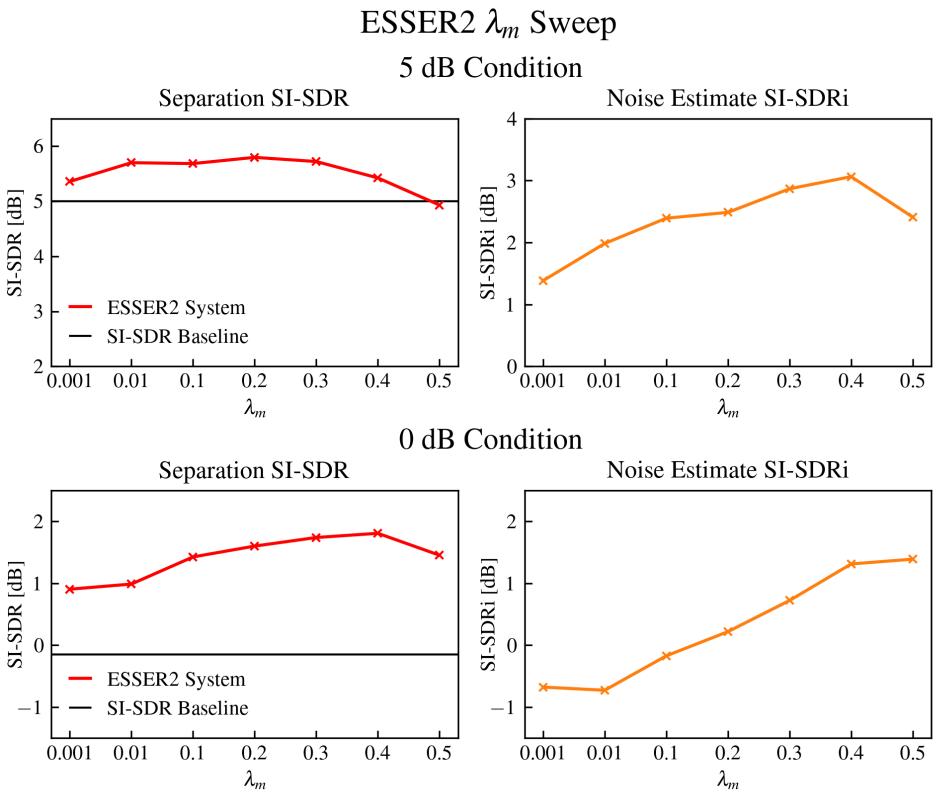


Figure 7.4: Comparison of performance as a function of λ_m with ESSER2 loss, with other parameters set to values reported in Table 7.2. We can see that only in extreme values do we start to see breakdown of performance. Note that the x-axis scale is not linear.

decreased to $\lambda_m = 0.001$ to show a decrease in performance at the low end. We believe that networks trained with ESSER2 loss are likely stable with respect to λ_m as long as it is below the breakdown point reflected in the experiments with ESSER loss. Not attempting separation is a considerably easier task than separation, which we believe leads to a very powerful optimization local optimum. As a result, the greatest risks come from providing too much of a discount.

The challenge of this overall approach is that the network must be encouraged to take advantage of the existence of the discounted noise estimate category without letting it swallow up the entire signal. In ESSER, adjusting λ to a point where the network does something beyond regular separation but not falling into a degenerate point is very difficult. However, in ESSER2, the regularizer term is what is used to guarantee that the network does something besides pure separation, and the λ_m term is only used to inform how the network partitions the signal. It can still fall into the degenerate optimum if λ_m is too large, but otherwise will be fine.

Figure 7.5 shows the results of a comparable set of experiments, but varying the regularizer term weight λ_r. Unsurprisingly, performance degrades for very small values of λ_r, as it is not influential enough to affect the objective function. Otherwise, the performance is robust with respect to this parameter, as long as it is within an appropriate dynamic range.

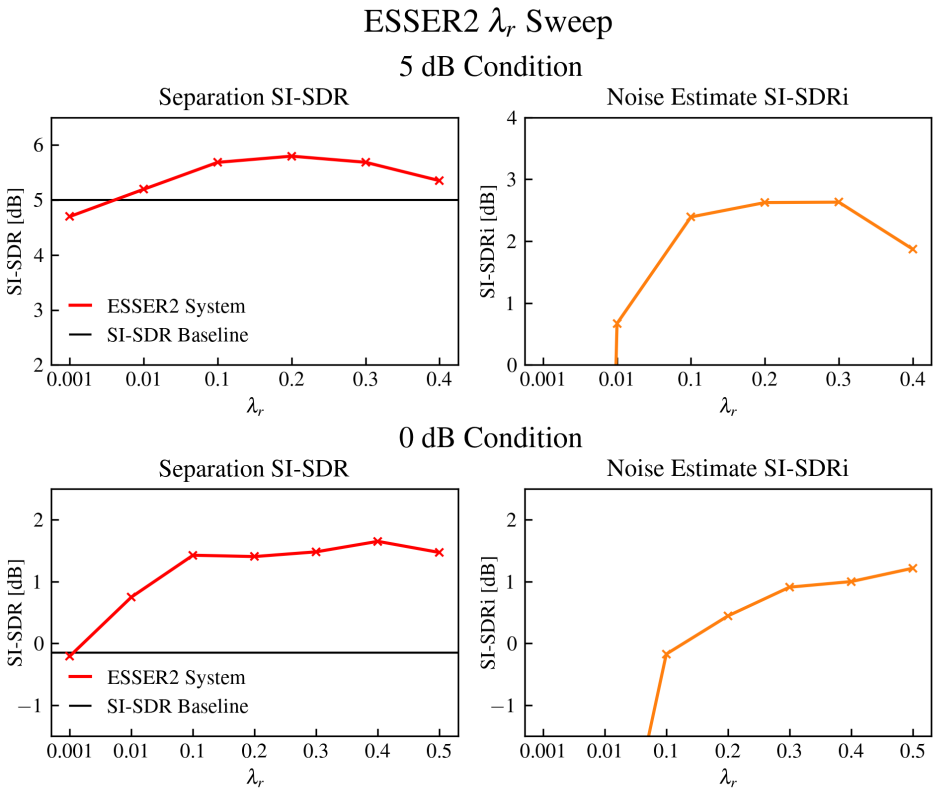


Figure 7.5: Comparison of performance as a function of λ_r with ESSER2 loss, with other parameters set to values reported in Table 7.2. We can see that performance is fairly stable as a function of this parameter, which is typical for a regularizer. Note that the x-axis scale is not linear.

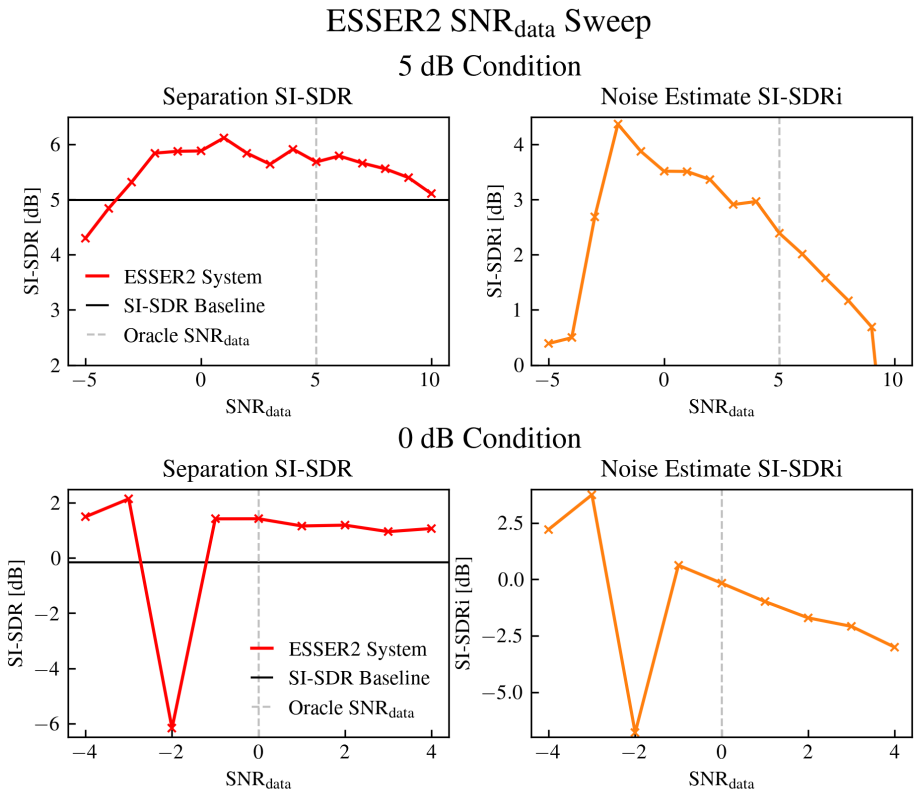


Figure 7.6: Comparison of model performance on the 5 and 0 dB datasets as a function of the SNR_{data} parameter in ESSER2 loss, with other parameters set to values reported in Table 7.2. We can see that performance is very stable as a function of this parameter, maintaining high values over a range of 10+ dB. Interestingly, the system seems to perform better at SNR values lower than oracle, suggesting perhaps the true SNR value is not best performance-wise.

Finally, Figure 7.6 shows the results of our experiments regarding network robustness with respect to the SNR_{data} parameter. This is of particular interest as the value would vary from dataset to dataset, and even could be a challenge on data where the SNR is not as consistent from file to file as it is with our synthetic dataset. The dataset SNR is something that could be estimated ahead of time, but an estimate will always have some level of error, and we would want our systems to be robust to errors in SNR estimation.

Fortunately, the range of SNR_{data} values which achieve comparable performance are over a range of 10+ dB, which gives a very large margin of error. What is particularly interesting, however, is that this buffer is not centered around the oracle SNR value, and the oracle value does not produce the best-performing system, with lower SNR values achieving better performance. We also note that the network trained in 0 dB condition with a parameter value of -2 dB failed to train. This does unfortunately indicate there may still be instabilities in training.

7.4.3 Issues and Future Work

One of the largest challenges facing this task is the issue of properly scaling the signals. As noted in Section 7.2.2.4, taking an approach to scaling that is comparable to the one used in SI-SDR cannot accordingly produce optimal scaling. The fact that these methods rely on the relative magnitude of the signals, despite being variable—even

depending on system output—is likely harming system performance. In fact, we explored use of a reconstruction loss, where all estimates should sum as close as possible to the mixture, with no success, something we attribute to likely being due to scaling issues.

Another line of future work would be further exploiting information we could gain about signals beyond perfect speech/non-speech annotation. For example, we greatly improved robustness of the system by simply giving it additional information about the signals: the SNR of the input data. The more information that we can provide for the system, the more likely it is that we can build a better training regime for this problem. We would like to explore a framework such as Generative Adversarial Networks, along the lines of [83], that would allow the incorporation of signal priors or some other expected signal information to aid the network in training. Additionally, many datasets involving noisy, far-field speech are recorded in parallel with close-talking microphones with cleaner speech signals [3, 51, 84]. It may be possible to use these parallel recordings to better inform what parts of the signal are more likely to be speech or noise. This type of information may be beneficial in improving performance while training on noisy data without perfect signal information.

Finally, we would like to extend this work to conditions that include reverberated speech. It is likely the case that reverberation, not just noise, is a significant reason as to why systems trained on real-world often perform very poorly. As a result, a

necessary next step to this line of research is to include reverberation as well.

7.5 Conclusion

We have proposed an approach for dealing with the challenges of training a speech separation system using data involving mixtures of speech with noise already present, addressing the problems raised in Chapter 6. The gains of the proposed ESSER2 loss are minor and do not solve the problem, but show promise that something can be done to handle these challenges.

Chapter 8

Speaker Recognition as Extrinsic Evaluation of Speech Separation

8.1 Introduction

Training and evaluation of speech separation systems in noise and reverberation pose a number of challenges. One of the major challenges is that speech separation evaluation is generally done by computing the divergence of the estimated waveform from the ground truth waveform, as discussed in Section 2.4.3. The evaluation metrics are accordingly sensitive to all components of the waveform, penalizing performance for reasons other than the true separation errors, i.e. failing to produce the desired speaker's speech or failing to remove the other speakers. And, beyond this, these

metrics cannot be used to evaluate *naturally* overlapped speech at all, as the ground truth separation waveforms are not available in any capacity.

As a result, we have conducted an investigation demonstrating the viability downstream speaker recognition as a method for extrinsic evaluation of the quality of speech separation. Speaker recognition is the class of tasks aiming to identify the person who was speaking in a recording. If these systems are designed to only work on non-overlapping speech, we could expect that their performance would degrade as more and more of another speaker is present in a waveform, or—reframed in the context of speech separation—as the quality of separation is reduced. Speaker recognition has the benefit of being very lightweight with respect to annotation: rather than requiring an entire ground truth waveform, requiring only knowledge of the speakers identities. And, state-of-the-art speaker recognition systems are relatively invariant to noise and reverberation [56].

An additional benefit of this extrinsic metric is that it provides some evidence of the value in speech separation as pre-processing for speaker verification tasks. Many applications impacted by overlap in conversational settings are either speaker recognition tasks or can be related to them, such as speaker diarization.

8.2 Discussion of Separation Evaluation

The methods of evaluation for system performance on a given task fall into two categories: intrinsic or direct evaluation and extrinsic evaluation through down-stream task performance. In direct evaluation, some fidelity metric is used to evaluate how closely system output matches the desired output. In evaluation through down-stream tasks, the system output is used as preprocessing for another system used on a different task, which is then evaluated.

8.2.1 Direct Separation Evaluation

8.2.1.1 Commonly-Used Metrics

As discussed in Section 2.4.3, the primary metric used in speech separation evaluation is Scale-Invariant Signal-to-Distortion Ratio (SI-SDR) [31]. This metric directly computes the error between the estimated source waveform and the ground truth waveform. Other metrics that are directly computed from the waveforms, but are less frequently used, are Signal-to-Distortion Ratio (SDR) and its companion metrics Signal-to-Interferences Ratio (SIR) and Signal-to-Artifacts Ratio (SAR) [36], as well as Short-Time Inteligibility (STOI) [37] and Perceptual Evaluation of Speech Quality (PESQ) [38] are used as well.

8.2.1.2 Challenges of Direct Evaluation

One of the biggest challenges in direct evaluation of speech separation quality is that the level of ground truth annotation required is simply not available in a natural mixture. By contrast, in a task such as automatic speech recognition (ASR), the annotation required for evaluation is simply the corresponding text. Systems can easily be trained and evaluated in noisy and reveberant conditions, as the labels themselves are unaffected by the audio condition. Speech separation evaluation however struggles in these sorts of conditions, as it is not possible to recover the "clean" speech signal from the interfering signals. And, regardless of whether the ground truth signal is clean or not, the performance metric will include the non-separation errors of failing to remove or produce noise in the estimates accordingly. There is also no way to evaluate separation performance on real, naturally-occurring speech mixtures—recovering the ground truth waveforms is a superhuman task that itself is the problem speech separation aims to solve.

Finally, as speech separation is a task largely desired as pre-processing for downstream speech technologies that are not designed for overlapping speech, a potential shortcoming of direct evaluation is that the evaluation metrics are not guaranteed to correlate with the impact of the separation system on the performance of the downstream task.

8.2.2 Separation Evaluation Through Downstream Tasks

By considering downstream tasks, there are considerably more options for performance evaluation, which additionally can be valuable for cases where the downstream task is the ultimate goal. However, a significant downside is that this approach is heavily dependent on the task, technique, and downstream model used. And, in some cases, downstream deep neural network (DNN)-based speech systems have been shown to have degraded performance on audio that has been produced or enhanced by a DNN [85, 86]. The most commonly desired downstream applications are speech recognition, speaker identification, and embedding-clustering–based diarization. Since the embeddings used in diarization systems typically come from speaker identification systems, we only consider the first two applications.

8.2.2.1 Separation Evaluation Through Speech Recognition

Though speech recognition of overlapping speech has largely been approached through end-to-end systems [87, 88], there has been some precedent of evaluating separated speech with speech recognition [89–91]. However, using ASR for downstream evaluation of speech separation does have downsides.

The biggest downside is that the data must contain complete utterances, which disallows the *min* condition defined in the wsj0-2mix dataset [18] and the use of any corpora that do not contain the constraint that the single-speaker waveforms contain

full utterances [19]. In cases where both speech signals are full utterance, this almost assuredly leads to a condition that does not consist of 100% overlap, which is its own challenge in speech separation. And in cases of natural overlap, only a small portion of the utterances contain overlap. And finally, transcription is one of the more intensive and costly annotation procedures, particularly in comparison to speaker identity.

8.2.2.2 Separation Evaluation Through Speaker Verification

The biggest appeal of speaker identification-based evaluation is that speaker identity is very easy annotation to attain compared to the transcript required for ASR and ground truth waveform for direct evaluation. In addition, the annotation labels are time-invariant, applying to any duration of speech, avoiding many of the issues of ASR regarding segmentation and level of overlap. This makes speaker ID performance a valuable extrinsic metric for separation evaluation.

There are, however, several design choices in such evaluations. The primary decision is whether to perform speaker identification (speaker identity classification) or speaker verification (acceptance/rejection of the presence of a speaker from an enrollment recording), and how to extend those tasks to multi-speaker mixtures. For sake of simplification, in this work we focus on a verification task, where a trial is *target* if the enrollment speaker is present in the mixture and *non-target* if none of the speakers in that mixture are the enrollment speaker.

8.3 Experimental Configuration

8.3.1 Data

For our experiments, we used the wsj0-2mix dataset [8] based on WSJ0 [18], the WHAMR! dataset introduced in Section 4.3 that extends wsj0-2mix to noisy and reverberant conditions, the "noisy oracle" (no-2mix) dataset described in Section 6.3.1, as well as the mx6-2mix and ch5-2mix datasets described in Section 4.2 based on the Mixer 6 [51] and CHiME-5 [3] corpora respectively. In all cases, the 16 kHz versions and *min* conditions are used. The wsj0-2mix corpus was chosen due to its ubiquitous use in nearly every deep learning-based speech separation study. WHAMR! was chosen due to the similarity of its conditions to many real-world applications. The no-2mix dataset was chosen to evaluate speaker verification as a solution to the evaluation challenges raised in its work. And finally, the mx6-2mix and ch5-2mix datasets were chosen for the purpose of evaluation in realistic environments, in contrast to the other datasets which contain fully-synthetic mixtures, with noise and reverberation being added to clean speech recordings after the fact.

For creating test sets for speaker verification evaluation, we developed an algorithm to generate self-contained trials from a given separation dataset. In other words, we wanted to be able to create a speaker verification test set that required no additional data beyond the separation dataset. As a result, the enrollment utterances used for

each mixture are taken from the ground truth non-overlapping source speech from a different mixture. A benefit of this approach is that a speaker verification evaluation set could be generated from any speech separation dataset that has speaker label information. However, due to the reuse of utterances that inherently results from this approach, care must be made in maximizing the diversity of pairwise comparisons. For example, it would be undesirable for the verification set to compare the same two utterances twice, or even two by the same pair of speaker, if avoidable. Our algorithm thus is designed to maximize the diversity of speaker comparisons as well as the usage of utterances for enrollment. Further details of this algorithm are presented in Appendix A.2.

To generate each speaker verification evaluation condition, we generated 2 target trials and 2 non-target trials for each mixture—the target trials consist of one speaker match for each of the two speakers in the mixture, and the non-target trials simply use a speaker that is different from both present in the mixture. It is worth noting that the trials we generated are not necessarily gender-balanced, instead approximately matching the gender balance of the source corpora. This is not ideal, but we felt the best option was to compromise on gender balance and focus on matching the conditions between the separation and speaker verification evaluation setups for a given dataset. In cases where separation trials correspond across datasets, the same trials were reused (wsj0-2mix, WHAMR!, and no-2mix use the same source recordings, and

both mx6-2mix and ch5-2mix have the same mixtures across multiple microphones).

8.3.2 Models and Training

All of our separation networks are TasNet-BLSTM [15] networks with 600 units in each direction, trained with negative SI-SDR [31] loss, which we feel is reasonably representative of standard speech separation techniques. For the analysis and synthesis bases, we used 500 filters with length 5 ms and shift 2.5 ms. To train a range of separation models for our experiments with varying levels of performance, we increased the filter size and stride, as we feel there is evidence in the research community that demonstrates that these parameters correlate strongly with separation performance [81].

Models were trained for 100 epochs using 4 second segments using the Adam [64] optimizer with an initial learning rate of 0.001. The learning rate is decreased by a factor of two if the validation loss does not improve for three consecutive epochs. In addition, gradient clipping is performed with a maximum ℓ_2 norm of 5. All networks were trained with negative SI-SDR loss in an utterance-level permutation-invariant manner [11].

The system we used for speaker identification was x-vector speaker embeddings [56] with a Probabilistic Linear Descriminant Analysis (PLDA) [57, 58] backend for producing scores between utterances. We used models trained for the Speakers

in the Wild evaluation [92] as described in [93]. The models are trained on VoxCeleb 1 [59] and VoxCeleb 2 [60], augmented with noise, music, babble, and reverberation. While this system was not trained on any overlapping or multi-speaker speech, we felt that this was a reasonable system to use as our speaker verification backend due to its strong performance record, its design for noisy and reverberant environments comparable to desired speech separation application environments, and the fact that it was designed for an application in which multi-speaker environments were a key aspect of the evaluation. In addition, as the primary goal of this work is to evaluate speech separation quality, we do not necessarily want to maximize the invariance of the speaker identification technique to overlapping speech. And, as this system is reasonably current, it still lends credibility to the value of speech separation as pre-processing for speaker identification.

Additionally, we evaluated models using two separate PLDA-based backends. The first was the original speaker verification system's backend, trained with the VoxCeleb data. We also evaluated a re-trained backend, that was trained on in-domain data for the test condition, i.e. speech that had been enhanced by the separation network. This was a combination of separated and oracle sources from the 'cv' sets that were consistent with the test separated sources and enrollment sources. While this significantly reduces the amount of data the PLDA is exposed to, it allows for better-matched recording conditions, and more importantly will allow the PLDA to

compensate to some degree for artifacts introduced by the separation DNN, which the system would otherwise have never been exposed to.

8.3.3 Evaluation

The intrinsic metric we use to evaluate the standard speech separation performance is Scale-Invariant Signal-to-Distortion Ratio (SI-SDR), defined in (2.12). This measures the ratio of signal power to error power, using a scaled version of the estimated source such that the error is orthogonal to the signal.

For the speaker verification experiments, we use the most common metric, Equal Error Rate (EER). In typical speaker verification systems, a pairwise score is computed for each trial. The threshold used for acceptance/rejection of a trial can be tuned per application to favor either false acceptances and false rejections. EER reports the error rate at the operating point where the percentage of false acceptances and false rejections are equal.

In our mixture-based speech separation experiments, we used a straightforward approach to handling trials with multiple speakers. In cases where no separation was performed, we simply scored the enrollment utterance against the mixture itself. In cases where we were evaluating separated mixtures, we scored the enrollment utterance against both separated waveforms, using the closest score as the ultimate score for the presence of the enrollment speaker in the test mixture.

Table 8.1: Documentation of performance across multiple conditions. EER [%] represents an error rate where smaller is better, while SI-SDRi [dB] measures signal improvement where larger is better. The Mix and Oracle columns provide an expected performance floor and ceiling, evaluating unprocessed mixtures and ground truth separation respectively, while Sys. columns report the performance of a TasNet separation system. The PLDA column is the system where the PLDA has been retrained with in-domain data including separated output.

Dataset	**Mix** EER	**Oracle** EER	**Sys.** EER	**PLDA** EER	**Sys.** SDRi
wsj0-2mix [8]	13.7	2.4	4.7	3.4	14.5
no-2mix [23]	20.9	2.6	15.7	6.8	12.9
no-2mix w/ noisy target	22.4	7.4	13.1	8.4	2.8
WHAMR! [20] rev.	16.6	2.6	11.4	6.1	9.4
WHAMR! rev. w/ rev. target	17.8	4.6	9.3	6.9	9.9
WHAMR! noise & rev.	20.8	2.6	19.5	9.5	9.3
mx6-2mix [19] near	19.0	5.8	12.9	13.9	9.2
mx6-2mix far	23.7	11.2	21.6	19.7	2.3
ch5-2mix [19] near	33.8	35.4	36.4	35.4	6.9
ch5-2mix far	37.4	31.9	37.5	36.9	0.4

For our baseline experiments, we provide a sense of performance floor and ceiling for a given dataset and separation task by performing speaker verification evaluation on both the input mixture and the oracle ground truth separated signal, which exist due to the synthetic nature of mixtures required for standard speech separation training and scoring.

8.4 Results and Discussion

8.4.1 Survey of Conditions

Our first set of experiments was to simply evaluate and document performance across a wide variety of condition, the results of which are shown in Table 8.1. The purpose of these experiments were to document the potential gains for speaker verification through separation of overlapping speech, as well as show how close to the expected speaker verification performance ceiling a speech separation system is able to attain given its separation performance. The "mix" column shows verification performance using the unprocessed mixture, and the "oracle" column shows verification performance using the ground truth separated sources, representing the floor and ceiling respectively. The "system" column shows the results of the separation system output, and the "PLDA" column shows the results of those same systems where the PLDA has been trained using audio separated by that system. Finally, the rightmost column shows the SI-SDRi performance for those systems, for reference.

The results show that in general, there is a significant difference in performance of speaker verification between evaluating mixtures and evaluating the oracle separated speech, giving evidence that there is great potential for improvements to speaker recognition systems through separation pre-processing. One notable exception is the CHiME-5 conditions, where the lack of improvement is likely due to the conditions

being exceptionally challenging. We also see from WHAMR! and no-2mix that although the speaker verification system was trained to be invariant to noise and reverberation, it tends to perform better on clean speech than noisy or reverberant speech.

In terms of the separation system performance, overall separation does generally improve performance over the baseline, with system EERs being lower than the EERs using the mixtures. One of the most interesting trends is in cases where the ground truth does or does not include the noise or reverberation (i.e., do we require the separation network to enhance the signal or not). In this case, intrinsic separation performance is better when the network is asked to perform enhancement (drastically so in the no-2mix dataset), but speaker verification performance tends to be better when the network is not enhanced. Further discussion of this is in Section 8.4.3.

The systems using a retrained PLDA generally improves performance of the system when using fully-synthetic data. However, it shows minimal impact in the mixtures using real data, i.e. the data created using Mixer 6 and CHiME-5. One theory is that this is a result of the ground truth data including noise, as explored in Chapter 6. However, the fact that the PLDA improves performance of the noisy target no-2mix case serves as evidence that this is not the case.

Another theory is that the realistic recordings are more similar to the VoxCeleb data the original PLDA was trained on, so retraining does little to help. However, this

discounts the theory that the PLDA could be compensating for the separation system's artifacts, which seems unlikely given its significant improvements in the synthetic datasets.

An additional theory is that the shorter utterance lengths of the realistic data (see Table 4.1) means that the xvectors are too low quality to train an effective PLDA. The relatively high error rates, even in the oracle results, support this theory. However, the mx6-2mix mixtures are not much shorter than the wsj0-2mix/no-2mix/WHAMR! mixtures on average, and the near-field mx6-2mix dataset has better oracle performance than the noisy target no-2mix dataset, despite the no-2mix condition benefiting from the PLDA.

8.4.2 System Comparison of SI-SDR to EER

Our next set of experiments were to investigate the relationship between SI-SDR and EER, with results shown in Figure 8.1. For these experiments, we varied performance of our separation system by changing the window size and shift for the TasNet bases. For each plot, the horizontal axis is the separation performance measured with SI-SDRi and the vertical axis is the verification performance measured with EER. The horizontal dotted and dashed lines are the verification performance of mixture and oracle separated audio. The top two plots are the wsj0-2mix and mx6-2mix near-field datasets, and the bottom two are those same systems with the PLDA trained on

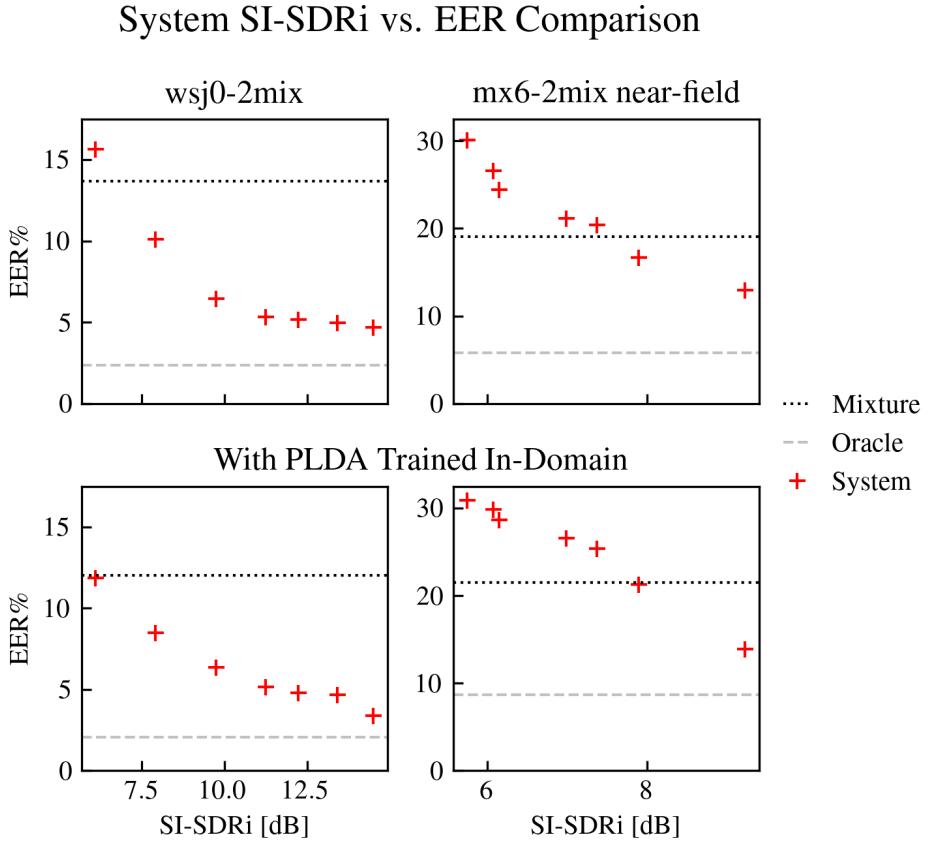


Figure 8.1: Comparison between SI-SDRi and EER on wsj0-2mix and the near-field Mixer 6 condition over a variety of TasNet models with different performance attained with variable sliding window size and shift.

in-domain system-separated audio.

The overall relationship among data points we collected is strictly monotonic, which is encouraging for the use of EER as a proxy metric for SI-SDR. We do, however, see the EER appear to level off at around double the oracle EER. This could suggest a number of things: that better system verification performance would require very significant separation gains and that pushing separation performance higher has strongly diminishing returns; that there is some inherent limitation to this model and that the SI-SDR–EER curve is system-dependent; or some alternative explanation. It is also worth noting that systems below a certain SI-SDRi around 7 dB performed worse on the speaker verification task than using the unprocessed mixture—even though those systems showed separation improvement over the mixture, the verification system performed worse using that separated output than using the mixture. This effect in wsj0-2mix, which contains no non-speech signals, supports the claim that the separation system is creating artifacts that the verification system is not robust to.

8.4.3 Noisy Ground Truth Results

Our final set of experiments were conducted to evaluate the sensitivity of the metrics to non-speech signals in the ground truth. The direct separation evaluation metrics are computed directly from the ground truth waveform and accordingly can penalize errors from parts of the signal that are not speech. This is an extension of the results

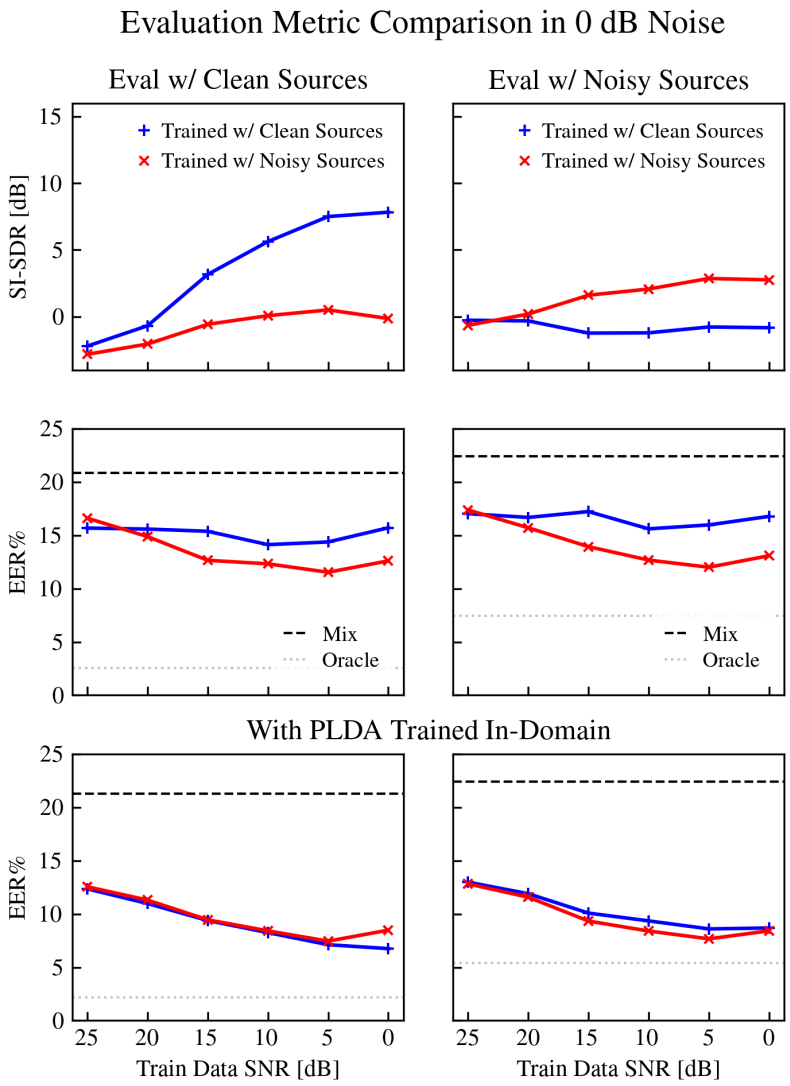


Figure 8.2: Comparison between SI-SDRi and EER on a 0 dB no-2mix condition in both clean and noisy ground truth configurations. Note that larger numbers are better for SI-SDR while smaller are better for EER.

analyzed in Section 6.3. Figure 8.2 shows the results of our experiments comparing the sensitivity of SI-SDR and EER to noise in the ground truth. The SI-SDR figures are reproduced from Chapter 6 for ease of comparison.

In all plots, both curves are the same two separation systems—the difference is that while they are trained using the same noisy mixtures, one is trained with clean sources and the other with noisy sources. Similarly, the left plots are evaluated according to clean source ground truth and the right noisy. The top plots report SI-SDR, the middle EER, and the bottom is EER using retrained, in-domain PLDA models.

A promising result is that the speaker verification results are consistent across both ground truth test conditions, and do show improvement in performance over the speaker verification baseline. In contrast, the direct separation evaluation differs greatly. And, not only does the performance differ, but the relative ranking of the two systems is not consistent.

An unfortunate result is that the only result showing successful separation comes from the model trained with clean targets and evaluated with clean targets—which is not the better-performing system for speaker verification. A qualitative assessment of sample audio suggests that the clean-trained separation model produces high-quality separated and denoised audio, while the noisy-trained model produces separated-but-noisy speech, which suggests that perhaps the extra processing of the denoising of the clean-trained model may result in a greater amount of harmful DNN artifacts.

The third row represents experiments designed to address this theory, using PLDA models trained using a combination of separated and enrollment utterances from the 'cv' set, with hopes that the PLDA could compensate for the effect of separation artifacts. Interestingly, it not only improves performance but almost completely closes the gap between both systems. The speaker ID systems seem to be relatively invariant to the ground truth condition a separation network is trained on. This is encouraging for the use of separation as pre-processing, but adds further evidence for concerns about the use of SI-SDR as a metric in noisy conditions: the first row suggests SI-SDR is very strongly affected by the amount of noise in the signals.

8.5 Conclusion

We have demonstrated the utility of speaker verification as a downstream evaluation of speech separation system performance, both through evidence of a monotonic relationship with SI-SDR and also through evidence of stronger invariance to non-speech signals present in the evaluated waveforms. Additionally, we have provided evidence that speech separation can improve performance of systems used in speaker recognition tasks.

Chapter 9

Conclusion

In this dissertation, we have contributed to the establishment of single-channel speech separation in the presence of noise in reverberation. This includes constructing and releasing a number of datasets suitable for training and evaluation, establishing the performance of state-of-the-art separation systems in those conditions, theoretical analysis of the impact of the interfering signals on the separation systems, as well as development of techniques aimed at addressing the performance degradation. In the following sections we will summarize the contributions of this thesis as well as the open problems that stem from this work.

9.1 Contributions

9.1.1 Dataset Creation

One of the primary contributions of this work was the development and release of a number of datasets suitable for training and evaluating the performance of single-channel speech separation systems in conditions where noise and reverberation are present. This work was presented in Chapter 4. It included both mixtures of naturally-occurring noise and reverberation, resulting in the mx6-2mix and ch5-2mix datasets [19], as well as mixtures where noise and reverberation have been added synthetically, the WHAMR! dataset [20]. The varying type of data creation allowed for a nuanced analysis of the impact that different types of ground truth signal can have on conventional speech separation systems.

9.1.2 Data Paradigm Analysis

Another contribution of this work is an analysis of different paradigms in creating data suitable for training and evaluating conventional speech separation systems that include noise. When noise is included in a speech mixture, the ground truth information can either include noise in the separated speech or it can be omitted from the ground truth sources.

Chapter 6 includes a demonstration of the significant impact this difference can

make not only in the training of systems, but also in evaluating them as well. This chapter also includes an analysis as to why this effect happens.

Additionally, Chapter 8 includes an exploration and demonstration of the use of speaker verification as an extrinsic metric to improve evaluation of speech separation systems, in part to address the sensitivity of the conventional metric to the non-speech signals in the mixture.

9.1.3 Improved Techniques

The final significant contribution of this work is the development of new techniques to address the problems articulated in this dissertation and improve performance of the systems subject to those problems.

Chapter 5 presents approaches for improving performance of separation systems when noise and reverberation is present—both an exploration of training data augmentation, as well as breaking down the task into denoising, dereverberating, and separation subtasks that are trained accordingly.

Chapter 7 presents a new training objective aimed at lessening the effects of noise being present in the ground truth waveforms while training separation systems with data involving noisy conditions. This approach aims to remove the impact on the waveform-level training objective of errors resulting from failure to separate noise signals.

9.2 Future Work

9.2.1 Metric Exploration

One of the primary contributions of this dissertation was a demonstration of shortcomings with the standard SI-SDR metric regarding its sensitivity to non-speech signals in the ground truth. Additionally, the stability that the speaker verification evaluation had with respect to these signals further suggests that SI-SDR has significant downsides. These results call for a deeper analysis of the metrics used and what those metrics are measuring. These issues are also exacerbated by the use of SI-SDR as both the training criterion and evaluation metric.

9.2.2 Alternative Training Objectives

We have demonstrated that the standard training objective used in speech separation systems shows strong sensitivity to variations in the ground truth, even when the speech signals themselves are consistent. As the speaker verification evaluation metric has shown to be relatively invariant to these issues, a promising line of work would be to explore alternative training objectives that do not require waveform-level reconstruction of the ground truth sources. However, one of the main challenges of this type of approach would be that, as the task itself requires the production of waveforms, there must be some mechanism to ensure the output is reasonable audio.

Appendix A

Dataset Creation Algorithms

A.1 Single Speakers to Mixtures

Algorithm Single Speaker Utterance List to Mixtures List

while num_mixes $<$ target_mixes **do**
  $\mathcal{S}_1 = \{utt : \text{usage\_count}(utt) = \min \text{usage\_count}(\cdot)\}$
  $u_1 = \arg\max_{u \in \mathcal{S}_1} \text{length}(u)$
  $i \leftarrow 0$
  while u_1 not yet matched **do**
    $\mathcal{S}_2 = \{utt : \text{usage\_count}(utt) = \min \text{usage\_count}(\cdot) + i\}$
    $\mathcal{S}_3 = \{utt : \text{spk}(utt) \notin \{spk : u_1 \text{ previously paired}\}\}$
    if $\mathcal{S}_2 \cap \mathcal{S}_3 \neq \emptyset$ **then**
      $u_2 = \arg\min_{u_2 \in \mathcal{S}_2 \cap \mathcal{S}_3} \mid \text{len}(u_1) - \text{len}(u_2) \mid$
      pair u_1 and u_2, update data structures
    else
      if $\mathcal{S}_2 = \emptyset$ **then**
        $\{spk : u_1 \text{ previously paired}\} \leftarrow \emptyset$
        $i \leftarrow 0$
      else
        $i \leftarrow i + 1$
      end if
    end if
  end while
end while

A.2 Mixtures to Speaker Verification

The algorithm for generating speaker verification trials from a list of mixtures is based on a particular queue structure. The motivation is that we want to minimize the amount of times any particular waveform is used. However, as the elements are selected according to a set of constraints specific to particular usages, it is not always possible to use a less-used waveform for every usage. So, to generate a queue for a

given set of elements, we initiate an infinite queue with the set of elements repeating indefinitely. To access an element from a queue, the queue is given a set of constraints and returns the first element that meets those constraints and removes it from the queue. This effectively returns the least-used element that matches the constraints. While this approach is greedy to some extent and is not necessarily optimal, we believe it is reasonable to solve this problem.

The queues used are as follows:

- spk_to_utts_queue: queue of utterances that exist for a particular speaker, parametrized by utterances to not be used
- utt_to_mix_queue: queue of mixtures that use a particular utterance (since we source enrollment from ground truth sources, a particular utterance can vary in length from mixture to mixture)
- utt_queue: queue of all utterances, parametrized by speaker to exclude

We also use the following structure worth explaining:

- spk_to_paired_spks: mapping from speaker to speakers that have already been used in non-target trials, for avoiding reuse of speaker pairs

Algorithm Mixtures List to Speaker Verification Trials

```
for mix in dataset do
    for utt in mix do
        spk ← utt_to_spk(utt)
        for num_target_trials per speaker per mix do
            enroll_utt ← spk_to_utts_queue[spk](utt)
            enroll_mix ← utt_to_mix_queue(enroll_utt)
            append "mix enroll_mix target" to trials
        end for
    end for
    for num_nontarget_trials per mixture do
        unusable_spks ← ∪[spk_to_paired_spks(utt_to_spk(utt)) for utt in mix]
        if len(unusable_spks) = num_spks then
            for spk in mix do
                clear spk_to_paired_spks(spk) if full
            end for
        end if
        enroll_utt ← utt_queue(unusable_spks)
        enroll_mix ← utt_to_mix_queue(enroll_utt)
        append "mix enroll_mix nontarget" to trials
    end for
end for
```

References

[1] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, *et al.*, "The ICSI meeting corpus," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, 2003, pp. 364–367.

[2] S. Bengio and H. Bourlard, *Machine learning for multimodal interaction*. Springer, 2005.

[3] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The fifth CHiME speech separation and recognition challenge: Dataset, task and baselines," in *Proc. ISCA Interspeech*, 2018.

[4] N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman, *First DIHARD challenge evaluation plan*, 2018.

[5] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe, and S. Khudanpur, "Diarization

is hard: Some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge," in *Proc. ISCA Interspeech*, 2018, pp. 2808–2812.

[6] M. Diez, F. Landini, L. Burget, J. Rohdin, A. Silnova, K. Žmolíková, O. Novotný, K. Veselý, O. Glembek, O. Plchot, L. Mošner, and P. Matějka, "BUT system for DIHARD speech diarization challenge 2018," in *Proc. ISCA Interspeech*, 2018, pp. 2798–2802.

[7] L. Sun, J. Du, C. Jiang, X. Zhang, S. He, B. Yin, and C.-H. Lee, "Speaker diarization with enhancing speech for the first DIHARD challenge," in *Proc. ISCA Interspeech*, 2018, pp. 2793–2797.

[8] J. R. Hershey, Z. Chen, and J. Le Roux, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 31–35.

[9] Y. Isik, J. Le Roux, Z. Chen, S. Watanabe, and J. R. Hershey, "Single-channel multi-speaker separation using deep clustering," in *Proc. ISCA Interspeech*, 2016, pp. 545–549.

[10] D. Yu, M. Kolbæk, Z. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 241–245.

[11] M. Kolbæk, D. Yu, Z.-H. Tan, and J. Jensen, "Multi-talker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 25, no. 10, pp. 1901–1913, 2017.

[12] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 246–250.

[13] Y. Luo, Z. Chen, and N. Mesgarani, "Speaker-independent speech separation with deep attractor network," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 4, pp. 787–796, 2018.

[14] K. Kinoshita, L. Drude, M. Delcroix, and T. Nakatani, "Listening to each speaker one by one with recurrent selective hearing networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5064–5068.

[15] Y. Luo and N. Mesgarani, "TasNet: Time-domain audio separation network for real-time, single-channel speech separation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 696–700.

[16] Y. Luo and N. Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[17] Z. Shi, H. Lin, L. Liu, R. Liu, S. Hayakawa, and J. Han, "FurcaX: End-to-end monaural speech separation based on deep gated (de)convolutional neural networks with adversarial example training," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6985–6989.

[18] J. Garofolo, D. Graff, D. Paul, and D. Pallett, "CSR-I (WSJ0) complete LDC93S6A," 1993, Web Download. Philadelphia: Linguistic Data Consortium.

[19] M. Maciejewski, G. Sell, Y. Fujita, L. P. Garcia-Perera, S. Watanabe, and S. Khudanpur, "Analysis of robustness of deep single-channel speech separation using corpora constructed from multiple domains," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Oct. 2019, pp. 165–169.

[20] M. Maciejewski, G. Wichern, E. McQuinn, and J. Le Roux, "WHAMR!: Noisy and reverberant single-channel speech separation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 696–700.

[21] M. Pariente, S. Cornell, J. Cosentino, S. Sivasankaran, E. Tzinis, J. Heitkaemper, M. Olvera, F.-R. Stöter, M. Hu, J. M. Martín-Doñas, D. Ditter, A. Frank, A. Deleforge, and E. Vincent, "Asteroid: The PyTorch-based audio source separation toolkit for researchers," in *Proc. ISCA Interspeech*, 2020.

[22] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proc. ISCA Interspeech*, 2018, pp. 2207–2211.

[23] M. Maciejewski, J. Shi, S. Watanabe, and S. Khudanpur, "Training noisy single-channel speech separation with noisy oracle sources: A large gap and a small step," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5774–5778.

[24] M. Maciejewski, S. Watanabe, and S. Khudanpur, "Speaker verification-based evaluation of single-channel speech separation," in *Proc. ISCA Interspeech*, 2021.

[25] D. Wang and G. J. Brown, *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Wiley-IEEE press, 2006.

[26] S. Makino, T.-W. Lee, and H. Sawada, *Blind speech separation*. Springer, 2007, vol. 615.

[27] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 1–12, 2007.

[28] D. Yu, X. Chang, and Y. Qian, "Recognizing multi-talker speech with permutation invariant training," in *Proc. ISCA Interspeech*, 2017, pp. 2456–2460.

[29] Y. Qian, X. Chang, and D. Yu, "Single-channel multi-talker speech recognition with permutation invariant training," *Speech Communication*, vol. 104, pp. 1–11, 2018.

[30] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with permutation-free objectives," in *Proc. ISCA Interspeech*, 2019, pp. 4300–4304.

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

[32] N. Zeghidour and D. Grangier, *Wavesplit: End-to-end speech separation by speaker clustering*, 2020. arXiv: 2002.08933 [eess.AS].

[33] Z.-Q. Wang, K. Tan, and D. Wang, "Deep learning based phase reconstruction for speaker separation: A trigonometric perspective," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 71–75.

[34] J. Le Roux, G. Wichern, S. Watanabe, A. Sarroff, and J. R. Hershey, "Phasebook and friends: Leveraging discrete representations for source separation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 370–382, 2019.

[35] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015.

[36] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[37] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010, pp. 4214–4217.

[38] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2, 2001, 749–752 vol.2.

[39] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, "A consolidated perspective on multimicrophone speech enhancement and source separation," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 25, no. 4, pp. 692–730, Apr. 2017.

[40] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. Le Roux, "WHAM!: Extending speech separation to noisy environments," in *Proc. ISCA Interspeech*, Sep. 2019.

[41] J. Cosentino, M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, *Librimix: An open-source dataset for generalizable speech separation*, 2020. arXiv: 2005.11262 [eess.AS].

[42] S. Wisdom, E. Tzinis, H. Erdogan, R. J. Weiss, K. Wilson, and J. R. Hershey, *Unsupervised sound separation using mixtures of mixtures*, 2020. arXiv: 2006.12701 [eess.AS].

[43] E. Vincent, T. Virtanen, and S. Gannot, *Audio Source Separation and Speech Enhancement*, 1st. Wiley Publishing, 2018, ISBN: 9781119279891.

[44] K. Tan and D. Wang, "A convolutional recurrent neural network for real-time speech enhancement," in *Proc. ISCA Interspeech*, 2018, pp. 3229–3233.

[45] A. Pandey and D. Wang, "A new framework for CNN-based speech enhancement in the time domain," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 27, no. 7, pp. 1179–1188, 2019.

[46] S. Pascual, A. Bonafonte, and J. Serrà, "SEGAN: Speech enhancement generative adversarial network," in *Proc. ISCA Interspeech*, 2017, pp. 3642–3646.

[47] S. Braun, B. Schwartz, S. Gannot, and E. A. P. Habets, "Late reverberation PSD estimation for single-channel dereverberation using relative convolutive transfer functions," in *Proc. IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2016, pp. 1–5.

[48] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. Juang, "Speech dereverberation based on variance-normalized delayed linear prediction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1717–1731, 2010.

[49] K. Han, Y. Wang, D. Wang, W. S. Woods, I. Merks, and T. Zhang, "Learning spectral mapping for speech dereverberation and denoising," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 23, no. 6, pp. 982–992, 2015.

[50] Z.-Q. Wang and D. Wang, "Deep learning based target cancellation for speech dereverberation," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 28, pp. 941–950, 2020.

[51] L. Brandschain, D. Graff, and K. Walker, *Mixer-6 Speech LDC2013S03*. Philadelphia: Linguistic Data Consortium, 2013.

[52] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannermann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The Kaldi speech recognition toolkit," in *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011.

[53] P. Ghahremani, V. Manohar, D. Povey, and S. Khudanpur, "Acoustic modelling from the signal domain using CNNs.," in *Proc. ISCA Interspeech*, 2016, pp. 3434–3438.

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

[55] D. Snyder, G. Chen, and D. Povey, *MUSAN: A music, speech, and noise corpus*, 2015. arXiv: 1510.08484 [cs.SD].

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

[57] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *IEEE International Conference on Computer Vision*, 2007, pp. 1–8.

[58] S. Ioffe, "Probabilistic linear discriminant analysis," in *ECCV*, A. Leonardis, H. Bischof, and A. Pinz, Eds., Springer-Verlag, 2006, pp. 531–542.

[59] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in *Proc. ISCA Interspeech*, 2017.

[60] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. ISCA Interspeech*, 2018.

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

[62] Z.-Q. Wang, J. Le Roux, and J. R. Hershey, "Multi-channel deep clustering: Discriminative spectral and spatial embeddings for speaker-independent speech separation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018.

[63] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A Python package for audio room simulation and array processing algorithms," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 351–355.

[64] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[65] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "mir_eval: A transparent implementation of common MIR metrics," in *Proc. of the 15th International Conference on Music Information Retrieval (ICMIR)*, 2014.

[66] Z.-Q. Wang, J. Le Roux, and J. R. Hershey, "Alternative objective functions for deep clustering," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018.

[67] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 1003–1012.

[68] Y. Luo and N. Mesgarani, "Real-time single-channel dereverberation and separation with time-domain audio separation network.," in *Proc. ISCA Interspeech*, 2018, pp. 342–346.

[69] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. ISCA Interspeech*, 2019, pp. 2613–2617.

[70] K. Kinoshita, M. Delcroix, S. Gannot, E. A. Habets, R. Haeb-Umbach, W. Kellermann, V. Leutnant, R. Maas, T. Nakatani, B. Raj, *et al.*, "A summary of the reverb challenge: State-of-the-art and remaining challenges in reverberant speech processing research," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, p. 7, 2016.

[71] B. Li, T. N. Sainath, A. Narayanan, J. Caroselli, M. Bacchiani, A. Misra, I. Shafran, H. Sak, G. Pundak, K. K. Chin, *et al.*, "Acoustic modeling for Google Home," in *Proc. ISCA Interspeech*, 2017, pp. 399–403.

[72] K. Han, Y. Wang, D. Wang, W. S. Woods, I. Merks, and T. Zhang, "Learning spectral mapping for speech dereverberation and denoising," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 23, no. 6, pp. 982–992, 2015.

[73] Y. Zhao, Z.-Q. Wang, and D. Wang, "Two-stage deep learning for noisy-reverberant speech enhancement," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 27, no. 1, pp. 53–62, 2018.

[74] M. Delfarah and D. Wang, "Deep learning for talker-dependent reverberant speaker separation: An empirical study," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 27, no. 11, pp. 1839–1848, 2019.

[75] T. Ochiai, S. Watanabe, T. Hori, J. R. Hershey, and X. Xiao, "Unified architecture for multichannel end-to-end speech recognition with neural beamforming,"

IEEE Journal of Selected Topics in Signal Processing, vol. 11, no. 8, pp. 1274–1288, Dec. 2017.

[76] S. Settle, J. Le Roux, T. Hori, S. Watanabe, and J. R. Hershey, "End-to-end multi-speaker speech recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018.

[77] H. Seki, T. Hori, S. Watanabe, J. Le Roux, and J. R. Hershey, "A purely end-to-end system for multi-speaker speech recognition," in *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, Jul. 2018.

[78] X. Chang, Y. Qian, K. Yu, and S. Watanabe, "End-to-end monaural multi-speaker ASR system without pretraining," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 6256–6260.

[79] S. Watanabe, M. Mandel, J. Barker, E. Vincent, A. Arora, X. Chang, S. Khudanpur, V. Manohar, D. Povey, D. Raj, D. Snyder, A. S. Subramanian, J. Trmal, B. B. Yair, C. Boeddeker, Z. Ni, Y. Fujita, S. Horiguchi, N. Kanda, T. Yoshioka, and N. Ryant, *CHiME-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings*, 2020. arXiv: 2004.09249 [cs.SD].

[80] I. Kavalerov, S. Wisdom, H. Erdogan, B. Patton, K. Wilson, J. Le Roux, and J. R. Hershey, "Universal sound separation," in *Proc. IEEE Workshop on*

Applications of Signal Processing to Audio and Acoustics (WASPAA), 2019, pp. 175–179.

[81] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path RNN: Efficient long sequence modeling for time-domain single-channel speech separation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 46–50.

[82] J. Shi, J. Xu, Y. Fujita, S. Watanabe, and B. Xu, "Speaker-conditional chain model for speech separation and extraction," in *Proc. ISCA Interspeech*, 2020, pp. 2707–2711.

[83] V. Narayanaswamy, J. J. Thiagarajan, R. Anirudh, and A. Spanias, "Unsupervised audio source separation using generative priors," in *Proc. ISCA Interspeech*, 2020.

[84] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, *et al.*, "The AMI meeting corpus: A pre-announcement," in *International workshop on machine learning for multimodal interaction*, Springer, 2005, pp. 28–39.

[85] S.-J. Chen, A. S. Subramanian, H. Xu, and S. Watanabe, "Building state-of-the-art distant speech recognition using the CHiME-4 challenge with a setup of speech enhancement baseline," in *Proc. ISCA Interspeech*, 2018, pp. 1571–1575.

[86] Z. Meng, J. Li, Y. Gong, and B.-H. Juang, "Adversarial feature-mapping for speech enhancement," in *Proc. ISCA Interspeech*, 2018, pp. 3259–3263.

[87] X. Chang, W. Zhang, Y. Qian, J. L. Roux, and S. Watanabe, "End-to-end multi-speaker speech recognition with transformer," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6134–6138.

[88] A. S. Subramanian, C. Weng, S. Watanabe, M. Yu, and D. Yu, "Deep learning based multi-source localization with source splitting and its effectiveness in multi-talker speech recognition," *arXiv*, 2021. arXiv: 2102.07955 [eess.AS].

[89] T. Menne, I. Sklyar, R. Schlüter, and H. Ney, "Analysis of deep clustering as preprocessing for automatic speech recognition of sparsely overlapping speech," in *Proc. ISCA Interspeech*, 2019, pp. 2638–2642.

[90] T. von Neumann, C. Boeddeker, L. Drude, K. Kinoshita, M. Delcroix, T. Nakatani, and R. Haeb-Umbach, "Multi-talker ASR for an unknown number of sources: Joint training of source counting, separation and ASR," in *Proc. ISCA Interspeech*, 2020, pp. 3097–3101.

[91] N. Kanda, C. Boeddeker, J. Heitkaemper, Y. Fujita, S. Horiguchi, K. Nagamatsu, and R. Haeb-Umbach, "Guided source separation meets a strong ASR backend: Hitachi/Paderborn University joint investigation for dinner party ASR," in *Proc. ISCA Interspeech*, 2019, pp. 1248–1252.

[92] M. McLaren, L. Ferrer, and D. Castán Lavilla, “The 2016 speakers in the wild speaker recognition evaluation,” in *Proc. ISCA Interspeech*, 2016, pp. 823–827.

[93] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, “Speaker recognition for multi-speaker conversations using x-vectors,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5796–5800.

Vita

Matthew Maciejewski received his BS in Electrical & Computer Engineering from Carnegie Mellon University in 2014. He began a PhD in the Electrical & Computer Engineering department in the Center for Language and Speech Processing later that year. Sanjeev Khudanpur and Shinji Watanabe co-advised his research, which focused on conversational speech technology in far-field conditions, primarily directed at single-channel speech separation. In July 2021, Matthew joined the Acoustic Event Detection research group as a research scientist at Amazon Alexa AI in Cambridge, Massachusetts.