SINGLE CHANNEL MULTI-TALKER SPEECH SEPARATION WITH DEEP LEARNING

CHENGLIN XU

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2020

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

16 Jan 2020 ..... Date

 ..... CHENGLIN XU

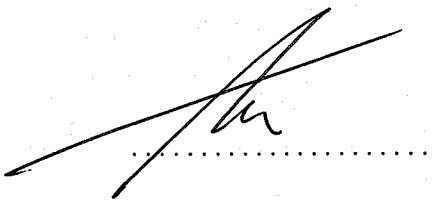
Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

16 Jan 2020

.....

Date



.....

A/Prof ENG SIONG CHNG

Authorship Attribution Statement

This thesis contains material from six papers published in the following peer-reviewed journal / from papers accepted at conferences in which I am listed as an author.

Chapter 3 is published as C. Xu, W. Rao, X. Xiao, E. S. Chng, and H. Li, Single channel speech separation with constrained utterance level permutation invariant training using grid LSTM, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 6-10.
and C. Xu, W. Rao, E. S. Chng, and H. Li, A shifted delta coefficient objective for monaural speech separation using multi-task learning, in *INTERSPEECH*, 2018, pp. 3479-3483.

The contributions of the co-authors are as follows:

- A/Prof Chng, Prof Li and I initialized the project direction of speech separation.
- Dr Xiao provided invaluable suggestions for the first work.
- Dr Rao, A/Prof Chng and Prof Li provided invaluable suggestions.
- I came up the ideas, designed the experiments, implemented codes, did the experiments and prepared the manuscript.
- The manuscript was revised together with Dr Rao, A/Prof Chng and Prof Li.

Chapter 4 and Chapter 5 are published as C. Xu, W. Rao, E. S. Chng, and H. Li, Optimization of speaker extraction neural network with magnitude and temporal spectrum approximation loss, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6990-6994.
and C. Xu, W. Rao, E. S. Chng, and H. Li, Time-domain speaker extraction network, in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 327-334.
and C. Xu, W. Rao, E. S. Chng, and H. Li, SpEx: multi-scale time domain speaker extraction network, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020.

The contributions of the co-authors are as follows:

- I initialized the speaker extraction idea, Dr Rao, A/Prof Chng and Prof Li provided invaluable suggestions.
- Dr Rao provided invaluable comments on how to capture the speaker characteristics as voiceprint using speaker embedding.

- I designed the experiments, implemented codes, did the experiments and prepared the manuscript.
- The manuscript was revised together with Dr Rao, A/Prof Chng and Prof Li.

Chapter 6 is published as W. Rao, C. Xu, E. S. Chng, and H. Li, Target speaker extraction for multi-talker speaker verification, in *INTERSPEECH*, 2019, pp. 1273-1277.

The contributions of the co-authors are as follows:

- I proposed the novel idea in solving the problem of speaker verification in multi-talker environment.
- I did the work of two speaker extraction methods to extract an output stream of single speaker from the overlapped multi-talker speech.
- I did speaker verification experiments and prepared the manuscript together with Dr Rao.
- Dr Rao, A/Prof Chng, Prof Li and I revised the manuscript together.

16 Jan 2020 ..... Date

Xu ..... CHENGLIN XU

To my dear wife, son and parents!

Abstract

The objective of speech separation is to divide a mixture signal, i.e., multiple speakers with background noise, into a set of individual streams, where each stream only contains a single speaker's voice. The study of speech separation is important as the performance of speech applications degrades dramatically in the presence of background noise, especially, interference speakers. Many real world speech applications are thus greatly limited. Towards this end, this thesis focuses on the techniques to improve the performance and practicality of speech separation, and the usage of speech separation technology to solve multi-talker speaker verification.

Speech always shows its characteristics of temporal continuity. However, the temporal continuity of the separated speech is broken by a frame leakage problem that someone's voice segment is wrongly separated into another speaker's output stream. To this end, this thesis first proposes a temporal objective function to optimize the neural network and a grid long short-term memory (LSTM) to learn spectro-temporal features. With these explicit temporal information as supervision and features, the temporal continuity, that is broken by windowing effect between frames, is bridged. The speech separation network is optimized through a multi-task learning framework with a subtask to predict an attribute (silence, single, and overlapped) for each time-frequency (TF) bin of the mixture magnitude. Experiments show that the proposed method significantly outperforms the corresponding baseline in both objective and subjective evaluations.

In general, speech separation methods require knowing or estimating the number of speakers in the mixture in advance. However, the number of speakers couldn't always be known in real world applications. Moreover, speech separation may suffer from what is called global permutation ambiguity problem, where the separated voice for the same speaker may not stick to the same output stream across long pauses or utterances. These two problems greatly limited speech separation in realistic environment. To this end, the second and third contributions of this thesis focus on a frequency-domain and a time-domain speaker extraction solutions, that

are special cases of speech separation, to address the aforementioned problems. The idea is to mimic human's ability of selective auditory attention by only extracting the target speaker's voice given a reference speech of that speaker. The time-domain speaker extraction method further avoids the inherent phase estimation problem in the frequency-domain method during the signal reconstruction stage. Experiments show that the proposed method significantly outperforms a variety of baseline approaches in different evaluation environments. The experiments also confirm that the proposed method is more flexible and practicable than other traditional speech separation methods.

The performance of speaker verification degrades significantly when the test speech is corrupted by interference speakers. Speaker diarization also fails to segregate speakers in presence of an overlapped multi-talker speech. To the best of my knowledge, the forth contribution of this thesis is the first solution that addresses the overlapped multi-talker speaker verification by a tandem system. The proposed speaker extraction methods are exploited as the front-end processing of a traditional speaker verification system, that is called as SE-SV. Experimental results show that SE-SV significantly improves the performance of speaker verification with overlapped multi-talker speech and outperforms oracle speaker diarization.

Acknowledgements

First and foremost, I would like to express my deepest appreciation to my supervisor, A/Prof Eng Siong Chng, and co-supervisor, Prof Haizhou Li (now in NUS), for their valuable discussions and suggestions. As through their solid knowledge, and insightful views, I have been inspired to come up novel ideas and conduct this research step by step. Their enthusiasm in research will continue to impact my future life.

I would like to thank Dr. Xiong Xiao (now in Microsoft), firstly for introducing me to the topic of speech separation, and secondly for his valuable suggestions and help on my PhD study. I also would like to thank Dr. Wei Rao for the great collaboration.

I would like to thank my teammates, Dr. Haihua Xu, Dr. Xiaohai Tian, Dr. Pham Van Tung, Dr. Chong Tze Yuang, Lim Zhi Hao, Ho Thi Nga, Zhiping Zeng, Nana Hou, Vu Thi Ly, Kyaw Zin Tun, Jibin Wu, for their supports in various ways.

I also would like to thank my wife, son, and parents for their love, encouragement, and support all the time.

Last but not least, I would like to thank the examiners for their voluntary work and suggestions.

List of Author's Publications

Journal

- **Chenglin Xu**, Wei Rao, Eng Siong Chng, and Haizhou Li. "SpEx: Multi-Scale Time Domain Speaker Extraction Network", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1370-1384, 2020.

- Jibin Wu, **Chenglin Xu**, Daquan Zhou, Haizhou Li, and Kay Chen Tan. "Progressive Tandem Learning for Pattern Recognition with Deep Spiking Neural Networks", submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Conference

- Xiang Hao, **Chenglin Xu**, Nana Hou, Lei Xie, Eng Siong Chng, and Haizhou Li, "Time-domain Neural Network Approach for Speech Bandwidth Extension", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 866-870.

- **Chenglin Xu**, Wei Rao, Eng Siong Chng, and Haizhou Li. "Time-domain Speaker Extraction Network", in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 327-334.

- Wei Rao, **Chenglin Xu**, Eng Siong Chng, and Haizhou Li, "Target Speaker Extraction for Multi-Talker Speaker Verification", in *INTERSPEECH*, 2019, pp. 1273-1277.

- **Chenglin Xu**, Wei Rao, Eng Siong Chng, and Haizhou Li, "Optimization of Speaker Extraction Neural Network with Magnitude and Temporal Spectrum

Approximation Loss", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6990-6994.

- Nana Hou, **Chenglin Xu**, Eng Siong Chng, and Haizhou Li, "Domain Adversarial Training for Speech Enhancement", in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2019, pp. 667-672.

- **Chenglin Xu**, Wei Rao, Eng Siong Chng, and Haizhou Li, "A Shifted Delta Coefficient Objective for Monaural Speech Separation using Multi-task Learning", in *INTERSPEECH*, 2018, pp. 3479-3483.

- **Chenglin Xu**, Wei Rao, Xiong Xiao, Eng Siong Chng, and Haizhou Li, "Single Channel Speech Separation with Constrained Utterance Level Permutation Invariant Training Using Grid LSTM", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 6-10, Calgary, Canada.

- **Chenglin Xu**, Xiong Xiao, Sining Sun, Wei Rao, Eng Siong Chng, and Haizhou Li, "Weighted Spatial Covariance Matrix Estimation for MUSIC based TDOA Estimation of Speech Sources", in *INTERSPEECH*, 2017, pp. 1894-1898, Stockholm, Sweden.

- Xiong Xiao, **Chenglin Xu**, Zhaofeng Zhang, Shengkui Zhao, Sining Sun, Shinji Watanabe, Longbiao Wang, Lei Xie, Douglas L Jones, Eng Siong Chng, and Haizhou Li, "A study of learning based beamforming methods for speech recognition", in *CHiME Workshop*, 2016, pp. 26-31.

Contents

Abstract xi

Acknowledgements xiii

List of Author's Publications xv

List of Figures xxi

List of Tables xxv

Symbols and Acronyms xxxi

1 Introduction 1
- 1.1 Background . . . 2
- 1.2 Motivations . . . 3
- 1.3 Contributions . . . 5
- 1.4 Thesis Outline . . . 7

2 Overview of Speech Separation 9
- 2.1 Background . . . 9
  - 2.1.1 Definition of Speech Separation . . . 9
  - 2.1.2 Corpus . . . 11
  - 2.1.3 Evaluation Metrics . . . 14
- 2.2 Speech Separation Technologies Before Deep Learning Era . . . 15
  - 2.2.1 Auditory Model . . . 16
  - 2.2.2 Decomposition Model . . . 17
  - 2.2.3 Generative Model . . . 19
- 2.3 Deep Learning based Speech Separation Technologies . . . 21
  - 2.3.1 Conventional Regression Techniques . . . 22
  - 2.3.2 Conventional Time-Frequency Masking Techniques . . . 24
  - 2.3.3 Deep Clustering . . . 26
  - 2.3.4 Deep Attractor Network . . . 28
  - 2.3.5 Permutation Invariant Training . . . 29
  - 2.3.6 Iterative Phase Reconstruction of Speech Separation . . . 32

2.3.7 Time-domain Audio Separation Network . . . 34
2.3.8 Speaker Beam for Target Speaker Recognition . . . 36
2.4 Summary . . . 38

3 Multi-task Learning of Neural Networks for Temporal Continuity in Speech Separation 39
3.1 Recapping of Time-frequency Masking based Speech Separation . . . 40
3.2 Temporal Objective Function and Grid LSTM Spectro-Temporal Feature . . . 41
3.2.1 Temporal Objective Functions . . . 41
3.2.2 Grid LSTM Spectro-Temporal Feature . . . 43
3.3 Multi-Task Learning Framework . . . 45
3.4 Experimental Setup . . . 47
3.4.1 Dataset and Network Configuration . . . 47
3.4.2 Evaluation Metrics . . . 48
3.5 Experimental Results . . . 49
3.5.1 Effect of Temporal Objective Functions . . . 49
3.5.2 Effect of Grid LSTM Spectro-Temporal Feature . . . 52
3.5.3 Effect of Multi-task Learning . . . 53
3.5.4 Same vs. Different Gender . . . 55
3.5.5 Evaluation of Frame Leakage . . . 55
3.5.6 Comparisons with Other Methods . . . 57
3.5.7 Subjective Evaluation . . . 58
3.5.8 Discussions . . . 59
3.6 Conclusion . . . 60

4 Top-down Selective Auditory Attention with Speaker Extraction 61
4.1 Motivation . . . 61
4.2 Relation to Speech Separation . . . 63
4.3 Frequency-domain Speaker Extraction . . . 65
4.3.1 A General Frequency-domain Framework . . . 65
4.3.2 Proposed Speaker Extraction with Magnitude and Temporal Spectra Approximation Loss . . . 66
4.4 Multi-scale Time-domain Speaker Extraction . . . 69
4.4.1 SpEx Architecture . . . 71
4.4.2 Multi-scale Encoding and Decoding . . . 76
4.4.3 Multi-task Learning . . . 77
4.4.4 Relationship with TasNet . . . 78
4.5 Summary . . . 79

5 Evaluation and Analysis of Speaker Extraction 81
5.1 Database . . . 82
5.1.1 WSJ0-2mix-extr and WSJ0-3mix-extr . . . 82
5.1.2 WSJ0-2mix, WHAM! and WHAMR! . . . 84

5.2 Frequency-domain Speaker Extraction Setup . . . 84
5.3 Multi-scale Time-domain Speaker Extraction Setup . . . 86
5.3.1 Speaker Encoder . . . 87
5.3.2 Speaker Extraction Pipeline . . . 88
5.4 Evaluation of Speaker Extraction on WSJ0-2mix-extr Database . . . 89
5.4.1 Frequency-domain Speaker Extraction Result . . . 89
5.4.2 Multi-scale Time-domain Speaker Extraction Result . . . 91
5.5 Evaluation of Speaker Extraction on WSJ0-3mix-extr Database . . . 101
5.6 Evaluation of Speaker Extraction on WSJ0-2mix Database . . . 102
5.7 Evaluation of Speaker Extraction on WHAM! Database . . . 104
5.8 Evaluation of Speaker Extraction on WHAMR! Database . . . 105
5.9 Conclusions . . . 107

6 Multi-talker Speaker Verification with Speaker Extraction 109
6.1 Problem of Multi-talker Speaker Verification . . . 110
6.2 Multi-Talker Speaker Verification with Speaker Extraction . . . 110
6.2.1 SBF-MTSAL . . . 112
6.2.2 SBF-MTSAL-Concat . . . 113
6.2.3 SpEx . . . 114
6.3 Experimental Setup . . . 116
6.3.1 Speaker Extraction Database . . . 116
6.3.2 Speaker Verification Database . . . 117
6.3.3 Speaker Extraction Setup . . . 118
6.3.4 Speaker Verification Setup . . . 118
6.4 Experimental Results . . . 120
6.4.1 SBF-MTSAL and SBF-MTSAL-Concat . . . 120
6.4.2 SpEx . . . 122
6.4.3 Speaker Extraction vs. Oracle Speaker Diarization . . . 122
6.5 Conclusions . . . 123

7 Conclusions and Future Work 125
7.1 Conclusions . . . 125
7.1.1 Multi-task Learning of Neural Networks for Temporal Continuity in Speech Separation . . . 125
7.1.2 Top-down Selective Auditory Attention with Speaker Extraction . . . 126
7.1.3 Multi-talker Speaker Verification with Speaker Extraction . . . 128
7.2 Future Work . . . 128

Bibliography 131

List of Figures

2.1 The NMF framework of training and inference procedure. Two speakers' mixture is taken as an example. 18
2.2 The regression framework of deep learning based speaker dependent systems. Two speakers' mixture is taken as an example. 23
2.3 The system architecture of DANet method for training. During run-time testing, the upper dotted box is not necessary. The systems takes input mixture and outputs the separation. 29
2.4 The framework of PIT technique using DNN or CNN for training. In run-time process, the upper dotted box is not necessary. The systems takes input mixture and outputs the output1 and output2 for the case of two speakers' mixture. 30
2.5 The architecture of uPIT technique with BLSTM for training in two speaker's mixture. In run-time process, the upper dotted box is not necessary. The systems takes input mixture and outputs the output1 and output2. 31
2.6 The general framework of TasNet technique. The separator could be a deep LSTM or a stacked temporal convolutional network (TCN) blocks. The TCN block is a dilated depth-wise separable convolution. The systems takes mixture signal in and outputs two separated source signals for the case of two speakers' mixture. 35
2.7 The framework of the SBF-IBM method to extract the target speaker's voice from a multi-talker speech given the speaker information encoded by the speaker encoder network. ""Sub 1" to "Sub N" are the number of sub-layers that have same network configuration. α_1 to α_N are the adaptation weights that are associated with the target speaker. 37

3.1 The computation of the loss with objective functions of speech features that minimizing the permutation error. 43
3.2 The training schema of the proposed multi-task learning framework for monaural speech separation. At run-time, the upper dotted box and the subtask in the upper right dotted box are not necessary. The system takes the input mixture and separates it into output1 and output2, where two-speaker mixture is taken as an example. . . 46

3.3 The GNSDR (dB) with different order L of the delta coefficient (See Eq. 3.1) objective function on the development set. 'All gender' is the overall result of the development set that has both different gender and same gender speakers in the mixtures. 'Different gender' is evaluated on the subset of the development set that only has different gender speakers in the mixtures (female and male mixture). Similarly, 'Same gender' is evaluated on the other subset of the development set that only contains same gender speakers in the mixtures (female and female mixture, male and male mixture). . . 50
3.4 The GNSDR (dB) with different shift I and block K with the SDC objective function on the development set. 51
3.5 The GNSDR (dB) of tuning the weight λ in our SDC-G-MTL multi-task learning system using 1 Grid LSTM layer and 3 BLSTM layers evaluated under three conditions on the development set. All gender: all data are used to calculate GNSDR. Different gender: GNSDR is computed on the data that the gender of the input mixture is different. Similarly, same gender: GNSDR is computed on the data that the gender of the input mixture is same. 54
3.6 Spectra of the mixture, two target speech of a female-female mixed example ('050a050i_2.1935_421c020b_-2.1935') from the test set are shown in (a), (b) and (c). The spectra of two separated output streams by the uPIT-BLSTM baseline and the proposed SDC-G-MTL method are shown in (d) and (e), (f) and (g). The speech frames of target speaker are marked in red in the upper panel of (h) and (i). The frame assignments of the uPIT-BLSTM baseline are shown in the middle panel, while those of the SDC-G-MTL method are in the lower panel. 56
3.7 The A/B preference test result of the reconstructed speech waveform between the proposed SDC-G-MTL method and the uPIT-BLSTM baseline. We conducted t-test using a significance level of $p < 0.05$ which is depicted with the error bars. 59

4.1 Emulating humans' ability of selective auditory attention with speaker extraction network, where a reference speech of target speaker is used to direct the top-down voluntary focus. 63
4.2 The block diagram of a general frequency-domain speaker extraction network, which consists of a speaker encoder and a speaker extractor. 66
4.3 The framework of SBF-MTSAL-Concat for speaker extraction with magnitude and temporal spectra approximation loss. $|\overline{S}|$ is the clean magnitude of the target speaker with a phase difference, which is equal to $|S| \otimes cos(\theta_y - \theta_s)$, as defined in Eq. 4.6. $f_d(\cdot)$ and $f_a(\cdot)$ are delta and acceleration computation function. 67

4.4 The block diagram of a general speaker extraction network, that consists of a speaker encode (in green), a speech encoder (in cyan), a speaker extractor (in purple), and a speech decoder (in cyan). The network components in Figure 4.4 and 4.5 share the same color codes for ease of cross reference. The speaker encoder simulates a top-down voluntary focus of cognitive process with the target speaker as the attention task. 70

4.5 The block diagram of the proposed SpEx network, that consists of a speaker encoder (in green), a speech encoder (in cyan), a speaker extractor (in purple), and a speech decoder (in cyan). The network components in Figure 4.4 and 4.5 share the same color codes for ease of cross reference. Ⓡ is an operator that concatenates the speaker vector repeatedly to the intermediate representations of mixture speech along the channel dimension. Ⓧ refers to the element-wise multiplication. The "conv" and "deconv" are 1-D convolutional and de-convolutional operations. "relu" and "sigmoid" are the rectified linear unit (ReLU) and sigmoid functions. The structure of the "tcn" block is similar to Conv-TasNet as shown in Figure 4.6. The extracted signal s_1 is chosen as the ultimate output of the system at run-time inference. 72

4.6 The structure of the "tcn" block is temporal convolutional network used in Figure 4.5. ⊕ denotes the residual connection. The "d-conv" is depth-wise convolution which forms a depth-wise separable convolution together with the last "1x1 conv". "prelu" is the parametric rectified linear unit (PReLU). "g-norm' is the mean and variance on both dimensions of time frames and channels scaled by the trainable bias and gain parameters. 76

5.1 The log magnitude spectra of a female-female mixture, its extracted speech for target speaker by the four baselines, the proposed SpEx network, and the clean speech from target speaker. 98

5.2 The A/B preference test result of the extracted target speaker's voice between the proposed SpEx method and the best SBF-MTSAL-Concat baseline. We conducted t-test using a significance level of $p < 0.05$ which is depicted with the error bars. 100

6.1 The flow chart of overlapped multi-talker speaker verification system. It consists of a speaker extraction module and a traditional speaker verification module. "$y(t)$" represents the overlapped multi-talker speech. "$x(t)$" represents the enrollment speech, which is also used as the reference speech in speaker extraction. "$\hat{s}(t)$" represents the extracted target speaker's speech from $y(t)$. 111

6.2 The architecture of the SBF-MTSAL method. "Sub" indicates the sub-layer. α represents the adaptation weights associated with the target speaker. "N" is the number of sub-layers, which is also equal to the dimension of the adaptation weights. $|\hat{S}|$ and $|S|$ are the extracted magnitude and the target clean magnitude, respectively. $f_d(\cdot)$ and $f_a(\cdot)$ are the delta and acceleration computation functions. During the inference, the calculation of the delta and acceleration is not necessary. 112
6.3 The overlapped percentages of the two-speaker mixture in the training set, development set, and test set. 117

List of Tables

2.1 Configuration of room impulse response parameters. 13

3.1 A comparison of GNSDR, SIR and SAR (dB) with 95% confidence intervals over different objective functions calculated on magnitude, delta, acceleration and SDC on WSJ0-2mix test sets using 3 BLSTM layers same as the network configuration in the uPIT-BLSTM baseline. The order L is set to 2 in delta, acceleration and SDC calculation. The shift I and block K are 2 and 4 when computing SDC. w_d, w_a and w_{sdc} are tuned to be 4.5, 10.0 and 5.0. 51

3.2 GNSDR, SIR and SAR (dB) with 95% confidence intervals in a comparative study with and without a grid LSTM on WSJ0-2mix test set. The SDC method refers to the network with 3 BLSTM layers, that is the same as in uPIT-BLSTM, with a SDC-based objective function (J_{sdc}). The SDC-G method further inserts one grid LSTM layer between the input and the BLSTM layers. * indicates our re-implementation 53

3.3 GNSDR, SIR and SAR (dB) with 95% confidence intervals in a comparative study of with or without multi-task learning on WSJ0-2mix test set. The weight λ is set to 0.2 for multi-task learning systems. * indicates our re-implementation. 54

3.4 GNSDR, SIR and SAR (dB) in a comparative study of same and different gender combinations on WSJ0-2mix test sets (Def Assign.). 'Diff.' is evaluated on the subset of the test set that only has different gender speakers in the mixtures (female and male mixture). Similarly, 'Same' is evaluated on the other subset of the test set that only contains same gender speakers in the mixtures (female and female mixture, male and male mixture). The weight λ is set to 0.2 for multi-task learning systems. * indicates our re-implementation. 55

3.5 FLER (%) in a comparative study of the proposed techniques and the baseline. The pitch continuity is evaluated to show the separation continuity. "VUV (%)" and "CORR" represent the voiced and unvoiced error rate and correlation calculated on the pitches between the separated and clean speech. The results are obtained on the WSJ0-2mix test set. * indicates our re-implementation. . . . 57

3.6 GNSDR (dB) in a comparative study among the competitive methods on the WSJ0-2mix dataset with optimal frame level assignment (re-alignment with reference speeches) and default assignment (actual assignment) on closed (CC) and open (OC) conditions. † is a two-stage model by stacking. ‡ is with curriculum learning. * indicates our re-implementation. IRM and IPSM are the upper-bound performance by reconstructing the signal using IRM or IPSM with the phase of the mixture signal. 58

5.1 SDR (dB), SI-SDR(dB) and PESQ in a comparative study of the frequency-domain speaker extraction systems under open condition. "Mixture" refers to original input mixture with zero effort. "SBF-MSAL" replaces the mask approximation loss in the SBF-IBM baseline with a magnitude spectrum approximation loss, as definded in Eq. 4.5. "SBF-MTSAL" further adds the temporal constraint to form a magnitude and temporal spectrum approximation loss, as defined in Eq. 4.6. "SBF-MTSAL-Concat" is the novel concatenation framework with magnitude and temporal spectrum approximation loss instead of the adaptation structure. "#Paras" means the number of parameters of the model. 89

5.2 SDR (dB) and PESQ in a comparative study of the same gender and different gender mixture under open condition. 91

5.3 SDR (dB), SI-SDR(dB) and PESQ in a comparative study between frequency-domain and time-domain under open condition. L_1 is the filter length of the convolution in the speech encoder for single scale in this experiment. N, O, P, Q, B, R are the parameters of the extractor defined in Section 4.4.1.3. In the frequency-domain implementation, we use the phase spectrum from the original mixture speech to reconstruct the speech signal. "#Paras" indicates the total number of parameters in the network. i-vector is used as feature representation of reference speaker. 92

5.4 SDR (dB), SI-SDR (dB) and PESQ in a comparative study between single-scale and multi-scale under open condition. L_1, L_2 and L_3 are the various filter lengths of convolutions in the speech encoder. N (256), O (256), P (512), Q (3), B (8), R (4) are the parameters of the extractor defined in Section 4.4.1.3. α and β are the weights defined in the multi-scale SI-SDR loss J_1 in Eq. 4.10. "#Paras" indicates the total number of parameters in the network. s_w=(1-α-β)s_1+αs_2+βs_3 denotes the weighted summation of the reconstructed signal. The number of parameters during evaluation is less than that of training when only picking s_1 as the reconstructed signal. i-vector is used as feature representation of reference speaker. 93

5.5 SDR (dB), SI-SDR(dB) and PESQ in a comparative study between i-vector and speaker embedding as feature representations of reference speaker under open condition. L_1 (20), L_2 (80) and L_3 (160) are the various filter lengths of convolutions in the speech encoder. N (256), O (256), P (512), Q (3), B (8), R (4) are the parameters of the extractor defined in Section 4.4.1.3. α and β are the weights defined in the multi-scale SI-SDR loss J_1 in Eq. 4.10. γ is the weight of multi-task learning defined in Eq. 4.13. "MTL" indicates whether the multi-task learning is applied. "#Paras" indicates the total number of parameters in the network. s_w=(1-α-β)s_1+αs_2+βs_3 denotes the weighted summation of the reconstructed signal. The number of parameters during evaluation is less than training when only picking s_1 as the output. 95

5.6 SDR (dB), SI-SDR(dB) and PESQ of extracted speech for the proposed SpEx network and other 4 competitive frequency-domain systems under open condition. "Mixture" refers to original input mixture with zero effort. "#Paras" means the number of parameters of the model. 96

5.7 SDR (dB) and PESQ in a comparative study of different and same gender mixture under open condition. 97

5.8 SDR (dB) of extracted speech when we evaluate the same SpEx system on varying duration of reference speech of target speaker at [0, 1)dB, [1, 3)dB, [3, 5]dB. 99

5.9 SDR (dB), SI-SDR(dB) and PESQ in a comparative study of different duration of the reference speech. "Random" indicates that the duration of the reference speech is random. 101

5.10 SDR (dB), SI-SDR(dB) and PESQ in a comparative study of different number of speakers in the mixed speech on WSJ0-2mix-extr and WSJ0-3mix-extr datasets. The duration of the reference speech is random during training. "#speakers" indicates the number of speakers in the mixture. "Dur." indicates the duration of the reference speech. 102

5.11 SDRi (dB), SI-SDR(dB) and PESQ in a comparative study on the WSJ0-2mix dataset under the open condition. "Ref: Avg. 7.3s" means the duration of the reference speech in the test set is random with an average duration of 7.3s. "Ref:60s" indicates the duration of the reference speech is fixed as 60s in the test set. The duration of reference speech in SpEx is always random during training. "#Paras" refers to the number of parameters of the model. ‡ indicates the latest Conv-TasNet with an additional skip-connection in each TCN block. For speech separation (SS) task, we report the results evaluated on the oracle-selected streams. For speaker extraction (SE) task, we report the results evaluated on the SpEx-extracted stream. The SI-SDR on the original mixture speech is 0dB, thus the SI-SDR is same as the value of SI-SDR improvement (SI-SDRi) that are reported in some works. 103

5.12 SDRi (dB), SI-SDRi(dB) and PESQ in a comparative study on the WHAM! dataset under the open condition, where the mixture is corrupted with additive noise. "Ref: Avg. 7.3s" means the duration of the reference speech in the test set is random with an average duration of 7.3s. "Ref:60s" indicates the duration of the reference speech is fixed as 60s in the test set. The duration of reference speech in SpEx is always random during training. For speech separation (SS) task, we report the results evaluated on the oracle-selected streams. For speaker extraction (SE) task, we report the results evaluated on the SpEx-extracted stream. The SDR and SI-SDR of the origianl mixture speech are −4.23dB and −4.5dB. 104
5.13 SDRi (dB), SI-SDRi(dB) and PESQ in a comparative study on the WHAMR! dataset under the open condition, where the mixture is corrupted with reverberation. "Ref: Avg. 7.3s" means the duration of the reference speech in the test set is random with an average duration of 7.3s. "Ref:60s" indicates the duration of the reference speech is fixed as 60s in the test set. The duration of reference speech in SpEx is always random during training. For speech separation (SS) task, we report the results evaluated on the oracle-selected streams. For speaker extraction (SE) task, we report the results evaluated on the SpEx-extracted stream. The SDR and SI-SDR of the origianl mixture speech are −0.12dB and −3.3dB. 105
5.14 SDRi (dB), SI-SDRi(dB) and PESQ in a comparative study on the WHAMR! dataset under the open condition, where the mixture is corrupted with both additive noise and reverberation. "Ref: Avg. 7.3s" means the duration of the reference speech in the test set is random with an average duration of 7.3s. "Ref:60s" indicates the duration of the reference speech is fixed as 60s in the test set. The duration of reference speech in SpEx is always random during training. For speech separation (SS) task, we report the results evaluated on the oracle-selected streams. For speaker extraction (SE) task, we report the results evaluated on the SpEx-extracted stream. The SDR and SI-SDR of the origianl mixture speech are −3.49dB and −6.1dB. 106
5.15 SDRi (dB), SI-SDRi(dB) and PESQ in an universal study of the SpEx system, which is trained on 4 conditions and tested on each condition individually. "Ref: Avg. 7.3s" means the duration of the reference speech in the test set is random with an average duration of 7.3s. "Ref:60s" indicates the duration of the reference speech is fixed as 60s in the test set. The duration of reference speech in SpEx is always random during training. 106

6.1 Performance of speaker verification (SV) system with and without speaker extraction. "Training" represents the type of training data. "Eval" represents the type of evaluation test data. "TSE" represents whether or which target speaker extraction method is used. "Baseline" represents the zero-effort test case where SV system is trained with clean data and evaluated on mixture data. "Upper Bound" represents the case where clean speech data are used in both training and testing, which offers the upper bound performance of multi-talker SV system. "OSD-SV" represents the case where we replace the speaker extraction network in Figure 6.1 with an oracle speaker diarization (OSD) system. "DCF08" represents the minimum detection cost with $P_{Target} = 0.01$. "DCF10" represents the minimum detection cost with $P_{Target} = 0.001$. †, ‡ and * represent the extracted data by SBF-MTSAL, SBF-MTSAL-Concat, and SpEx, respectively. The details of experimental setup can be referred to section 6.3.4. 121

Symbols and Acronyms

Symbols

$\mathbb{R}^n$	n-dimensional Euclidean space
$\|\cdot\|_F$	Frobenius Norm
$\otimes$	element-wise product
$\langle\cdot,\cdot\rangle$	inner product of two vectors
$x_{i,j}$	i-th component of a vector x at time j
$\bar{x}$ or $\hat{x}$	estimated value close to x
$\mathbf{1}$	all-ones column vector
$\mathcal{C}^T$	transpose of a vector or matrix
$F(\cdot)$	a function or a module, i.e., a neural network

Acronyms

ASR	Automatic Speech Recognition
BLSTM	Bidirectional Long Short Term Memory
CADNN	Context Adaptive Deep Neural Network
CASA	Computational Acoustic Scene Analysis
cuPIT	Constrained Utterance-level Permutation Invariant Training
DANet	Deep Attractor Network
DC	Deep Clustering
DNN	Deep Neural Network
FLER	Frame Leakage Error Rate
GMM	Gaussian Mixture Model
GNSDR	Global Normalized Signal to Distortion Ratio
HMM	Hidden Markov Model

IAM	Ideal Amplitude Mask
IBM	Ideal Binary Mask
IRM	Ideal Ratio Mask
ISTFT	Inverse Short-time Fourier Transform
LDA	Linear Discriminate Analysis
LSTM	Long Short Term Memory
MISI	Multiple Input Spectrogram Inversion
MSAL	Magnitude Spectrum Approximation Loss
MSE	Mean Square Error
MTL	Multi-task Learning
MTSAL	Magnitude and Temporal Spectrum Approximation Loss
NMF	Non-negative Matrix Factorization
PESQ	Perceptual Evaluation of Speech Quality
PIT	Permutation Invariant Training
PLDA	Probabilistic Linear Discriminant Analysis
PSM	Phase Sensitive Mask
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
SAR	Signal to Artifacts Ratio
SBF	SpeakerBeam Frontend
SDC	Shifted Delta Coefficient
SDR	Signal to Distortion Ratio
SI-SDR	Scale-invariant Signal to Distortion Ratio
SIR	Signal to Interferences Ratio
SNR	Signal to Noise Ratio
STFT	Short-time Fourier Transform
SV	Speaker Verification
TasNet	time domain audio separation network
TCN	Temporal Convolutional Network
TTS	Text-to-Speech
UBM	Universal Background Model
uPIT	Utterance-level Permutation Invariant Training
VAD	Voice Activity Detection
WFM	Wiener-filter Like Mask

Chapter 1

Introduction

Speech is the most natural way of human machine interaction. With the recently dramatic development of speech perception (listening and understanding) and speech generation (speaking) technologies, speech has been widely adopted in many applications for near-field human machine interaction. However, speech perception (i.e., speech recognition) and speech generation (i.e., text-to-speech) technologies are vulnerable to the presence of reverberation, background noises, and interference speech. Without solving the problems caused by these adverse conditions in far-field communications, especially, interference speech, the applications of speech technologies are greatly limited. Therefore, how to come up a solution to make human machine interaction robust against these adverse environments in real world applications is still a challenging problem, such as, hearing aids [1], teleconferencing [2], smart devices [3], and robotics [4].

This chapter starts with an introduction of speech separation that aims to solve the cocktail party problem in Section 1.1. Section 1.2 describes the existing problems in speech separation and the motivations of this thesis. The contributions of this thesis are summarized in Section 1.3. Finally, Section 1.4 presents the outline of this thesis.

1.1 Background

Humans have the remarkable ability to focus auditory attention on a particular voice by masking out the acoustic background in the presence of multiple talkers and background noises [5, 6]. This is called cocktail party effect or cocktail party problem. The remarkable ability is implemented with three stages: accurate processing of low-level stimulus attributes, segregation of auditory information into coherent voices, and selectively attending to a voice at the exclusion of others to facilitate higher level processing [7]. Speech separation aims to solve this cocktail party problem by mimicking human's ability of the first two stages of accurate processing and segregation. Specifically, there are two possible scenarios: a single channel recording of multi-talkers speech, or multi-channel recordings of multi-talkers speech. Speech separation learns distinguishable patterns from the original single or multi-channel mixture signals, and then divides each speaker's voice into an individual output stream.

The multi-channel speech separation problem is a simpler task as compared to speech separation given only a single channel recording. This is because multi-channel speech separation could be benefit from spatial information. This thesis mainly focuses on the challenging single channel speech separation unless otherwise explicitly mentioning the multi-channel input case. Because the acquisition of single channel signal is much easier and more common in most real world applications. In addition, a well-developed single channel speech separation could be easily extended to be applicable to multiple channels.

Inspired by human auditory scene analysis, computational auditory scene analysis (CASA) [8–13] is the first attempt of speech separation techniques to solve the cocktail party problem. To avoid the human designed features in CASA methods, non-negative matrix factorization (NMF) [14–22], and gaussian mixture model together with hidden markov model (GMM-HMM) [23–25], apply machine learning techniques to automatically discover some patterns that distinguish different speakers in a multi-taker speech. However, these approaches don't generalize well to unseen speakers due to the issue of speaker permutation ambiguity problem.

Recent studies have seen major progress by addressing the local speaker permutation ambiguity problem within a frame or utterance with the ability of deep learning, such as, deep clustering (DC) [26–28], deep attractor network (DANet)

[29, 30], permutation invariant training (PIT) [31, 32], and time-domain audio separation network (TasNet) [33–35]. These speech separation methods have been equipped the ability to separate a multi-taker speech with new speakers that are unseen by the system during training. Similar to the traditional methods, these deep learning based speech separation methods still require knowing or estimating the number of speakers in the mixture in advance. However, the number of speakers couldn't always be known in advance in real world applications.

To address the limitation of unknown number of speakers, the studies [36, 37] try to iteratively separate and reconstruct the speech signal for each individual speaker. The iteration procedure is terminated by either a stop-flag or a threshold of the residual mask in [36]. In [37], a simple deep neural network based binary classifier is used to predict whether the residual signal of the input is speech or not in every iteration. The recursion step continues only when the residual signal is predicted as speech. However, the iteration procedure limits the run-time process to be real-time.

Furthermore, speech separation methods may suffer from what is called global speaker permutation ambiguity problem, where the separated voice for the same speaker may not stick to the same output stream across frames in an online system, and long pauses or utterances in an offline system. Because the separation is done frame by frame in an online system or utterance by utterance in an offline system. In this work, we focus on addressing the aforementioned limitations to make speech separation possible in real world applications.

1.2 Motivations

As we known human achieves selective auditory attention with three stages, speech separation only implements the first two stages without attending to a voice at the exclusion of others, as discussed in Section 1.1. Speech separation mimics human's bottom-up sensory-driven attention, for example, a loud explosion that would attract attention [38]. In addition, human has another top-down task specific attention, for example, a flight announcement of one's interest in a busy airport [38]. Human's attention is non-static, and modulated rapidly at real-time in response to

the bottom-up acoustic stimulus and the top-down attention task in the cognitive process.

Different from speech separation, speaker extraction emulates human brain's top-down selective auditory attention. The idea of speaker extraction is to provide a reference speech of a target speaker to direct the attention to the attended speaker. Given the reference speech of the target speaker (similar to human's specific task in mind), speaker extraction only extracts the target speaker's voice and filters out other competing signals. The reference speech could be different from the speech in the mixture and is used to model the target speaker's characteristics. Similar to speech separation, speaker extraction could obtain a set of separated source signals from the mixed signal by extracting each speaker's voice given a reference speech of that corresponding speaker in parallel. From the point of view, speaker extraction is a special case of speech separation. Meanwhile, each extracted signal is known which speaker it belongs to.

By only extracting the target speaker's voice, speaker extraction overcomes the problems of unknown number of speakers and global speaker permutation ambiguity in a speech separation approach. In addition, speaker extraction is able to handle different conditions in an unified framework, such as, two-speaker mixture, three or more speaker mixture, one speaker with background noise, and multi-talker speech with reverberation and noise. Because speaker extraction always takes a mixture speech and a reference speech as its inputs, and outputs a single stream no matter what kind of condition the input mixture is. It's different from a speech separation approach that has different number of output streams when the number of speaker in the input mixture is various. Therefore, it is important to implement and improve speaker extraction systems to solve the cocktail party problem.

Speech applications always suffer from performance degradation in a cocktail party environment. The performances of these speech applications need to be studied when speech separation is applied as a front-end processing in a tandem system. This work studies the effectiveness of speaker extraction in improving the performance of speaker verification in a multi-talker speech environment, where the reference speech of the target speaker is available through an enrollment process. To the best of my knowledge, there is no solution to address the overlapped multi-taker speaker verification problem.

1.3 Contributions

In this thesis, we firstly address the frame leakage problem where some frames of one speaker are leaked to another speaker's output stream during an utterance processing in an offline speech separation system. Then, we study a frequency-domain and a time-domain speaker extraction approach to overcome the problems of unknown number of speakers and global speaker permutation ambiguity across frames and utterances in an online and offline speech separation systems, respectively. Finally, we propose a tandem system as the first attempt to solve the overlapped multi-talker speaker verification problem in a real world application. The details of four contributions are summarized as below.

Multi-task Learning of Neural Networks for Temporal Continuity in Speech Separation: In the first study, we introduce a temporal objective function and a grid LSTM to learn spectro-temporal features together with a multi-task learning framework. Previous frequency-domain speech separation methods always calculate a mean square error (MSE) in spectral feature dimension as an objective function without leveraging temporal information. The temporal continuity of the output stream is broken by the frame leakage problem within an utterance. The proposed framework exploits the temporal information in both objective function and feature learning to address the frame leakage problem. Specifically, the temporal objective function exploits dynamic information within a contextual window to compute MSE for network optimization. The grid LSTM captures spectro-temporal features on top of an image-like spectrum in both spectral and temporal dimensions through inner communications. Finally, a novel multi-task learning framework incorporates the speech separation task with a sub-task that predicts time-frequency attributes (silence, single and overlapped) of each time-frequency bin.

Top-down Selective Auditory Attention with Speaker Extraction: The second and third contributions are to address the problems of unknown number of speakers and global speaker permutation ambiguity by a frequency-domain and a time-domain speaker extraction techniques, respectively. The idea of speaker extraction is to mimic human's ability of top-down selective auditory attention by taking a reference speech of the target speaker

as a cue. Speaker extraction consists of four components, namely speaker encoder, speech encoder, speaker extractor, and speech decoder. Specifically, the speech encoder converts the mixture speech into feature representations. The speaker encoder learns to represent the target speaker with a speaker embedding, which guides the top-down auditory attention. The speaker extractor takes the feature representations and the target speaker embedding as inputs and estimates a receptive mask. The speech decoder reconstructs the target speaker's speech from the masked feature representations. The frequency-domain speaker extraction approach applies short-time Fourier transform (STFT) and invert STFT (iSTFT) as the speech encoder and speech decoder, respectively. However, such an approach is adversely affected by the inherent difficulty of phase estimation.

The time-domain speaker extraction network (SpEx) adopts a multi-scale encoding and decoding scheme with a trainable speech encoder and a trainable speech decoder instead of STFT and iSTFT. SpEx doesn't decompose the speech signal into magnitude and phase spectra, therefore, it avoids the phase estimation problem. The multi-scale encoding and decoding scheme enables SpEx to capture multiple temporal resolution for high voice quality. A multi-task learning framework is further proposed to jointly optimize the SpEx network with a cross-entropy loss for speaker classification to improve the quality of speaker embedding and a multi-scale scale-invariant signal-to-distortion ratio (SI-SDR) loss for speaker extraction to evaluate the quality of the estimated signal. Finally, an unified SpEx system is studied under various cocktail party environments.

Multi-talker Speaker Verification with Speaker Extraction: Although a reference speech of the target speaker is required, speaker extraction is practical to the applications where only register speakers are need to be responded, for example, speaker verification. To the best of my knowledge, the fourth contribution of this thesis is the first solution to solve the speaker verification problem when the test utterance is corrupted by interference speakers most of the time. The multi-talker speaker verification system with speaker extraction as a front-end is called as SE-SV. The proposed speaker extraction methods in both frequency-domain and time-domain are exploited to extract the speech of the speaker characterized by the enrollment speech from the mixture. Then, a traditional speaker verification system makes a decision to

decline or accept the extracted speaker by comparing whether the extracted speech and the enrollment speech are the same speaker.

The first contribution on multi-task learning of neural network for temporal continuity discussed in Chapter 3 is published in the conferences: ICASSP 2018 [39] and INTERSPEECH 2018 [40]. The second contribution on the top-down selective auditory attention with speaker extraction discussed in Chapter 4 and Chapter 5 is published in the conferences: ICASSP 2019 [41], ASRU 2019 [42], and its extension is published in the journal: IEEE/ACM Transactions on Audio, Speech and Language Processing [43]. The third contribution on multi-talker speaker verification with speaker extraction discussed in Chapter 6 is published in the conference: INTERSPEECH 2019 [44].

1.4 Thesis Outline

The rest of this thesis is organized as follows:

Chapter 2 provides a background of speech separation, and reviews the state-of-the-art speech separation methods.

Chapter 3 proposes a temporal objective function, and learns spectro-temporal features by a grid LSTM with a multi-task learning framework to make separated speeches temporal continuous.

Chapter 4 proposes a frequency-domain and a time-domain speaker extraction approaches, which mimic human's ability of top-down selective auditory attention. Meanwhile, speaker extraction solves the problems of unknown number of speakers and global speaker permutation ambiguity across frames an utterances in an online and an offline speech separation approaches, respectively.

Chapter 5 evaluates and analyzes the frequency-domain and time-domain speaker extraction approaches proposed in Chapter 4. Various cocktail party environments are studied, such as, various speaker mixture condition, noisy mixture condition, reverberant mixture condition, and the mixture condition with both noise and reverberation.

Chapter 6 proposes a tandem system as the first attempt to address the overlapped multi-talker speaker verification.

Chapter 7 concludes the findings of this thesis and discusses future research directions.

Chapter 2

Overview of Speech Separation

This chapter starts with an introduction of speech separation and its signal definition in Section 2.1. Since the cocktail party problem has been studied for decades, we review some previous representative works in Section 2.2 and 2.3. Section 2.2 summarizes the traditional speech separation technologies before deep learning era. Section 2.3 reviews the deep learning based speech separation technologies.

2.1 Background

2.1.1 Definition of Speech Separation

Speech signals recorded with distant microphones are inevitably corrupted by interfering speaker, noise and reverberation, which severely degrade the perceptual quality of the captured speech signals. When there is a single distant microphone as the recording device, the received discrete signal $y(n)$ will be expressed as,

$$y(n) = \sum_{c=1}^{C} h_c(n) * s_c(n) + b(n) \tag{2.1}$$

where there are number of C speakers talking at the same time and n is a time position of each sample. $s_c(n)$ is a speech signal of an individual speaker. $h_c(n)$ is the room impulse response between the c_{th} speaker and the microphone. $b(n)$ is a background noise, and $*$ is a convolution operator.

Speech separation aims to estimate each individual signal $\hat{s}_c(n)$ that is close to the original speech signal $s_c(n)$. To estimate the signal $\hat{s}_c(n)$, most speech separation methods firstly transform the time-domain mixture signal into its magnitude and phase spectra with a STFT for a frequency-domain processing.

By using a STFT, the time-domain mixture signal $y(n)$ is transformed as,

$$Y(t,f) = \sum_{n=0}^{N-1} y(n+tL)w(n)\exp(-j2\pi nf/N) \tag{2.2}$$

where a time frame $t \in [0, T-1]$ is obtained by shifting an amount of L samples, and a frequency bin $f \in [0, N-1]$ is associated with a frequency of $(f/N)f_s$ Hz at a sampling rate of f_s Hz. $w(n)$ is a hamming or hanning window of length N.

With the time-domain signal model as Eq. 2.1, the frequency-domain mixture signal can be decomposed as,

$$Y(t,f) = \sum_{c=1}^{C} H_c(t,f)S_c(t,f) + B(t,f) \tag{2.3}$$

where $H_c(t,f)$, $S_c(t,f)$, and $B(t,f)$ are the complex spectra of corresponding time-domain speech signal, reverberation, and background noise.

In a typical setup, speech separation estimates the complex spectrogram $\hat{S}_c$ for each speaker, which is close to the original signal S_c. To obtain the complex spectrogram $\hat{S}_c$, the magnitude $|\hat{S}_c|$ and phase $\angle\hat{S}_c$ spectra need to be estimated. Since phase spectrogram shows little temporal and spectral regularities, it's a challenging problem to manipulate the phase spectrum to be a better one. Most speech separation methods use the phase spectrogram of the mixture signal as the phase for each speaker ($\angle\hat{S}_c = \angle Y$). Therefore, frequency-domain speech separation methods typically focus on how to improve the quality of the estimated magnitude spectrogram $|\hat{S}_c|$.

The goal of the separation is to make the estimated magnitude $|\hat{S}_c|$ as close as the clean magnitude $|S_c|$ for each source. When the magnitude spectrogram $|\hat{S}_c|$ for c_{th} speaker is estimated, the complex spectrogram $\hat{S}_c$ could be obtained by combining the magnitude $|\hat{S}_c|$ and the phase $\angle Y$ of the mixture signal. Therefore, each estimated time-domain signal $\hat{s}_c(n)$ could be reconstructed by an overlap and

add algorithm,

$$\hat{s}_c(n) = \sum_{t=0}^{T-1} v(n - tL)\hat{s}_{c,t}(n - tL) \tag{2.4}$$

together with an iSTFT,

$$\hat{s}_{c,t}(n) = \frac{1}{N} \sum_{f=0}^{N-1} \hat{S}_c(t, f) \exp(j2\pi nf/N) \tag{2.5}$$

where $v(n)$ is a window similar to $w(n)$ in Eq. 2.2.

2.1.2 Corpus

Recent works are mostly conducted on public available corpus for easy comparisons, such as, two-speaker mixture database without noise and reverberation (WSJ0-2mix), two-speaker mixture database with noise (WHAM!), and two-speaker mixture database with noise and reverberation (WHAMR!). This thesis also conducts the experiments on these well-known speech separation corpora in order to fairly compare with the state-of-the-art speech separation methods[1].

2.1.2.1 WSJ0-2mix Database

WSJ0-2mix database aims to simulate the recording condition, where there are only two speaker talking at the same time. This condition simplifies the mixture signal model in Eq. 2.1 as a special case without noise and reverberation,

$$y(n) = \sum_{c=1}^{C} s_c(n) \tag{2.6}$$

where C is equal to 2 in this WSJ0-2mix database.

The two-talker mixed WSJ0-2mix dataset[2], was mixed by randomly choosing utterances of two speakers from the WSJ0 corpus [45]. The database was simulated at a sampling rate of 8kHz for a manageable computation load and memory cost.

[1] Since WHAM! and WHAMR! are released during the time-domain speaker extraction work of Chapter 5, the performances on these two databases are only compared in Chapter 5.

[2] Available at: http://www.merl.com/demos/deep-clustering. The database used in this work was simulated with the released script and file list in [26].

It was composed of three sets: training set ($20,000$ utterances $\approx 30h$), development set ($5,000$ utterances $\approx 8h$), and test set ($3,000$ utterances $\approx 5h$). Specifically, the utterances from 50 male and 51 female speakers in the WSJ0 training set (si_tr_s) were randomly selected to generate the training and development set in WSJ0-2mix at various signal-to-noise (SNR) ratios uniformly chosen between 0dB and 5dB. Similarly, the test set was created by randomly mixing the utterances from 10 male and 8 female speakers in the WSJ0 development set (si_dt_05) and evaluation set (si_et_05). Since the speakers in the development set of WSJ0-2mix dataset were the same as those in the training set, the development set was used as closed condition (CC) to tune parameters. Moreover, the test set was considered as open condition (OC) evaluation, because the speakers in the test set were different from those in the training and development sets, and unseen during training.

2.1.2.2 WHAM! Database

WSJ0 Hipster Ambient Mixtures (WHAM!)[3] database was created to simulate a more realistic condition, where two speaker's voices are mixed together with background noise [46]. This database described the mixture signal model by only considering interference speech and background noise,

$$y(n) = \sum_{c=1}^{C} s_c(n) + b(n) \tag{2.7}$$

where C is equal to 2 in this WHAM! database.

The WHAM! database was created by adding real ambient noise into the two-speaker mixtures same as those in WSJ0-2mix database. The noise samples were recorded in coffee shops, restaurants, bars, office buildings, and parks. The noise was mixed in by a random SNR chosen from a uniform distribution between -6dB and +3dB. In each mixture, two speaker's voices were used same as those in the original WSJ0-2mix database. The noise was randomly selected from a noisy recording at a random start point. The first speaker's voice to simulate the mixture was applied with a gain to make the SNR between this speaker and the noise equal to the random chosen SNR value. The same gain was then applied to the second

[3]Available at: http://wham.whisper.ai

TABLE 2.1: Configuration of room impulse response parameters.

	Length (m)	Wide (m)	Height (m)
Room	$U(5, 10)$	$U(5, 10)$	$U(3, 4)$
Microphone Center	$\frac{L_{Room}}{2} + U(-0.2, 0.2)$	$\frac{W_{Room}}{2} + U(-0.2, 0.2)$	$U(0.9, 1.8)$
	High (s)	Middle (s)	Low (s)
T_{60}	$U(0.4, 1.0)$	$U(0.2, 0.6)$	$U(0.1, 0.3)$
	Height (m)	Distance (m)	θ (°)
Sources	$U(0.9, 1.8)$	$U(0.66, 2)$	$U(0, 360)$

speaker's voice. This resulted in the training set, development set and test set of 20, 000, 5, 000 and 3, 000 paired utterances, respectively.

2.1.2.3 WHAMR! Database

Although WHAM! database introduced noise to the two-speaker mixture WSJ0-2mix database, there was no reverberation into consideration. The reverberation was always presented in indoor recordings outside of soundproof studios. To include the reverberation, WHAMR![4] database augmented the WHAM! database by applying synthetic reverberated sources. This database was simulated with the condition described in Eq. 2.1 with interference speech, background noise and reverberation.

The synthetic reverberated sources were generated by convolving the anechoic signal with room impulse response (RIR). The RIR was simulated using pyroomacoustics [47] according to the random room configuration shown in Table 2.1. The WHAMR! database defined four core separation tasks: 1) clean mixture condition: anechoic clean mixture to anechoic sources (same as WSJ0-2mix); 2) noisy mixture condition: anechoic noisy mixture to anechoic sources (same as WHAM!); 3) reverberant mixture condition: reverberant mixture to anechoic sources; 4) noisy and reverberant mixture condition: noisy and reverberant mixture to anechoic sources. Since the clean and noisy mixture conditions were simulated same as WSJ0-2mix and WHAM!, this thesis only used WHAMR! to evaluate the performance of speech separation under the reverberant mixture condition, and the noisy and reverberant mixture condition.

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

2.1.3 Evaluation Metrics

The speech separation systems could be evaluated by both objective evaluation and subjective evaluation. The objective evaluation is used to measure the distortion and quality between the separated and target clean signals. The subjective evaluation is to examine the perceptual quality and intelligibility of the separated signal.

2.1.3.1 Objective Evaluation

Signal-to-distortion ratio (SDR), signal-to-interferences ratio (SIR), and signal-to-artifacts ratio (SAR) [48] calculates the energy ratios expressed in decibels between the estimated wanted signal and the distortion, the interferences and the artifacts, respectively. They are calculated by firstly decomposing the estimated signal $\hat{s}$ as

$$\hat{s} = s_t + e_i + e_n + e_a \tag{2.8}$$

where $s_t = f(s)$ is the modified source signal s with an allowed distortion $f(\cdot)$. e_i, e_n, and e_a are the inteferences, sensor noise, and burbling artifacts error terms, respectively.

SDR, SIR, and SAR are then calculated as,

$$SDR = 10log_{10}\frac{||s_t||^2}{||e_i + e_n + e_a||^2} \tag{2.9}$$

$$SIR = 10log_{10}\frac{||s_t||^2}{||e_i||^2} \tag{2.10}$$

$$SAR = 10log_{10}\frac{||s_t + e_i + e_n||^2}{||e_a||^2} \tag{2.11}$$

The higher SDR, SIR, and SAR means the better performance of the speech separation system. To measure the performance improvement, SDR improvement (SDRi) is computed by using the SDR of the separated speech to minus that of the mixture speech [31, 32]. The SDRi is also defined as global normalized SDR (GNSDR) [49].

Perceptual Evaluation of Speech Quality (PESQ) [50, 51] is recommended as the ITU-T P.862 standard to automatically assess the speech quality instead of the

subjective Mean Opinion Score (MOS). The key of PESQ is to predict the MOS value that would result if real people are evaluating the recorded speech clips. Comparing with MOS, PESQ has the benefit to avoid involving listening subjects into the evaluation. Different from SDR without distinguishing the type of distortion, PESQ incorporates a perceptual model to distinguish between audible distortion (i.e. a noise added to the spectrum) and inaudible distortion (i.e. a spectral component omitted or heavily attenuated) with different weights. Compared to additive components, the omitted or attenuated components may be not easily perceived because of masking effects. The range of the PESQ score is –0.5 to 4.5. The higher score represents the better performance of speech separation systems.

2.1.3.2 Subjective Evaluation

An A/B preference test is a subjective evaluation to compare the speech quality between two systems. For each A/B listening pair, one sample is from the proposed system while the other one is from the comparing baseline. These two samples are randomly presented to listeners. The listeners don't know which sample belongs to which system. Each listener is asked to listen to both samples and chooses the better sample in terms of speech quality. The higher preference rate to the samples of a speech separation system indicates the better performance of the system.

2.2 Speech Separation Technologies Before Deep Learning Era

Speech separation could be formulated as unsupervised learning or supervised learning task according to whether source signals are available. Unsupervised learning based approach is also known as blind source separation (BSS) where both the sources and the mixing process are unknown. BSS aims to recover unknown source signals from the received mixture observations only. The most popular BSS approach is independent component analysis (ICA) [52]. It assumes the source signals are statistically independent and non-Gaussian distribution.

Most ICA methods [53, 54] focus on the case where the number of received observations is larger or equal to the number of sources signals (overdetermined or determined BSS). Some works extend the overdetermined or determined BSS to only separate the target source when a reference signal is available, such as, constrained independent component analysis (cICA) [55], and ICA with reference (ICA-R) [56]. The works in [57, 58] extend ICA to the underdetermined BSS (e.g., single channel speech separation in this thesis). However, the single channel ICA (SCICA) is only practical under certain circumstances. For example, the SCICA framework requires a necessary condition that the independent source processes have disjoint spectral support [58]. Since this thesis only focuses on supervised learning based speech separation techniques, more details of ICA based BSS methods could be found in the handbook [54].

Supervised learning based approach conducts speech separation given a variety of labelled training data, including the mixed signal and its individual source signals. A speech separation system could be trained with these data. And the trained model is applied to decompose the mixed signals into individual sources during inference. With the supervision of labelled data, supervised learning based systems always outperform those of unsupervised learning. Although supervised learning requires a variety of labelled training data, the labelled data is free to be obtained by simulation. Therefore, this thesis mainly focuses on the study of supervised learning based speech separation methods.

Before the deep learning era, the representative speech separation approaches could be grouped into auditory model, decomposition model and generative model, such as, computation auditory scene analysis (CASA) [8–13, 59], non-negative matrix factorization (NMF) [14–22], and gaussian mixture model with hidden markov model (GMM-HMM) [23–25, 60, 61].

2.2.1 Auditory Model

The pioneering works, referred as CASA [8–13, 59], are inspired by the findings on how human listeners separate sources in an acoustic mixture, for example, the auditory scene analysis in psychoacoustic research [62]. According to the mechanism and findings of the separation by humans, a series of heuristic rules are designed in the CASA methods. The frequently-used heuristic rules include pitch continuity

[9], vocal tract continuity [9], temporal continuity [59], harmonic [11], and common onset or offset [13].

According to the heuristic rules, the CASA methods group the time-frequency bins belonging to the same speaker together. Then, the signal of expected speaker is extracted from the mixture based on a binary mask formed by grouping time-frequency bins. Among these heuristic rules, pitch contour is the most useful one due to its characteristics of continuity. In addition, different speakers may have different pitch values. Since a speaker can only produce one instantaneous pitch, there will be two or more speakers if two or more pitch contours are detected.

Since the extraction of features that clearly show the grouping effects is the most important step, CASA methods suffer from the main drawback of inaccurate trackers (i.e., pitch tracker). In addition, the designed rules may be suboptimal and can not generalize well to unseen cases because of the limited observations. Furthermore, the extraction of the signal for the expected speaker is based on the binary mask, which has been shown to be suboptimal [63].

2.2.2 Decomposition Model

To overcome the drawbacks of human designed heuristic rules in CASA, a decomposition model, such as NMF [14–22], was an early attempt to automatically find the complex inherent characteristics from data. The decomposition model always assumes that the audio spectrogram is a low rank matrix. It means that only a small number of basis could be enough to represent the non-negative spectrogram $Y \in \mathbb{R}^{F \times T}$, as follows,

$$Y \approx BW \tag{2.12}$$

where $B \in \mathbb{R}^{F \times K}$ and $W \in \mathbb{R}^{K \times T}$ are the non-negative basis matrix and weight matrix. Typically, the hyper-parameter K ($K < F < T$) results in a compressed representation and a low-rank approximation of the data. F and T are the dimension of frequency features and the number of frames.

To find B and W given Y with the non-negative constraints $B \geq 0, W \geq 0$, an optimization criterion is formulated as following,

$$\min_{B,W \geq 0} D(Y||BW) \tag{2.13}$$

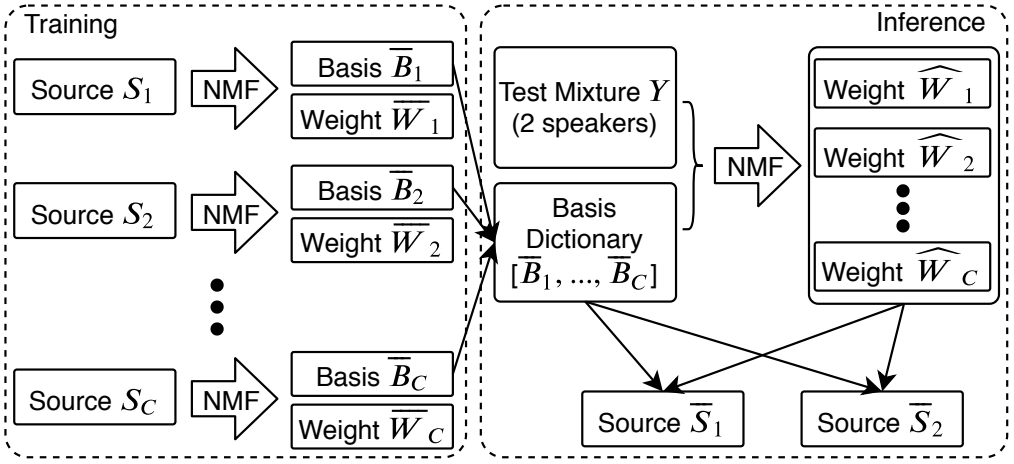


FIGURE 2.1: The NMF framework of training and inference procedure. Two speakers' mixture is taken as an example.

where $D(\cdot)$ can be Euclidean distance or Kullback-Leibler (KL) divergence. The optimization algorithm of NMF with KL divergence as a criterion is summarized in Algorithm 1 [64].

Algorithm 1: KL-NMF

Initialize B, W

Repeat

$W \leftarrow W. * \frac{B^T \frac{Y}{BW}}{B^T \mathbf{1}}$

$B \leftarrow B. * \frac{\frac{Y}{BW} W^T}{\mathbf{1} W^T}$

Until *convergence*

Return B, W

The procedure of NMF for speech separation is shown as in Figure 2.1. It first learns a basis dictionary by decomposing the clean spectrogram S_c of each clean source signal into non-negative basis matrix $\bar{B}_c$ and non-negative weight matrix $\bar{W}_c$. The basis dictionary is kept and the weight matrix is discarded, as shown in the left dotted box of Figure 2.1. The decomposition of each source $c \in [1, C]$ is in terms of an optimization objective defined as below,

$$\bar{B}_c, \bar{W}_c = \underset{B_c, W_c \geq 0}{\arg\min}\, D(S_c | B_c W_c) \tag{2.14}$$

During the run-time inference as shown in the right dotted box of Figure 2.1, a weight matrix $\hat{W}_c$ for each source is obtained by decomposing the spectrogram Y

of the mixed signal given the learned basis dictionary $[\bar{B}_1, ..., \bar{B}_C]$, as below,

$$\hat{W}_c = \arg\min_{W \geq 0} D(Y|\bar{B}_c W) \tag{2.15}$$

The learned basis dictionary $[\bar{B}_1, ..., \bar{B}_C]$ is fixed during the decomposition at the run-time inference. Each source signal $\bar{S}_c$ could be estimated by the basis dictionary $[\bar{B}_1, ..., \bar{B}_C]$ and the optimized weight $[\hat{W}_1, ..., \hat{W}_C]$ as following,

$$\bar{S}_c = \bar{B}_c \hat{W}_c \tag{2.16}$$

To further improve the performance, a wienner filtering like mask is typically used to reconstruct the separated source $\hat{S}_c$,

$$\hat{S}_c = Y \otimes \frac{\bar{S}_c}{\sum_c \bar{S}_c} \tag{2.17}$$

The time-domain separated signal $\hat{s}_c$ could be reconstructed using the overlapped-and-add algorithm together with inverse STFT as defined in Eq. 2.4 and Eq. 2.5.

By adding different constraints to the optimization criterion, NMF has been extended to several variants, such as, sparse NMF [19, 21], convolutive NMF [17, 18, 21], discriminative NMF [22]. However, the main limitations of NMF and its variants come from the linear assumption of the decomposition and the expensive computation cost for the optimization during inference. In addition, the decomposition of signal into a basis and a weight doesn't exploit the temporal dynamics of a speech signal. Furthermore, the size of the trained model (basis dictionary) is determined by the number of learnt basis and increased linearly with the number of source signals. Finally, the system is unable to generalize to unknown signal that is unseen during training, because there is no learnt basis for this unknown signal in the basis dictionary. This limits its ability in real-world applications.

2.2.3 Generative Model

In speech processing, we usually assume the observed data belongs to a Gaussian distribution. Due to the limited ability of a single Gaussian, a GMM is typically used to model the complex data. The assumption is that the observed data is

generated by mixing a set of GMM distributed source signals with unknown mean and variance. GMM based technology always conducts the separation on log power spectrum. Here, the distribution of the source signal S_c for speaker $c \in [1, C]$ within a GMM is defined as,

$$p(S_c) = \sum_{k=1}^{K} \phi_k \mathcal{N}(S_c; \mu_{c,k}, \Sigma_{c,k}) \tag{2.18}$$

where K is the number of Gaussian components in the GMM and ϕ_k denotes the weight (a prior probability, $\sum_{k=1}^{K} \phi_k = 1$) of the k^{th} Gaussian component. $\mu_{c,k}$ and $\sigma_{c,k}$ are the mean and covariance matrix of the k^{th} Gaussian for the log power spectrum of c^{th} speaker. $\mathcal{N}(S_c; \mu_{c,k}, \sigma_{c,k})$ is a multivariate Gaussian probability density function.

Band quantization is employed to reduce the heavy computational cost caused by the multivariate GMM [24]. Each of the K Gaussians of each model is approximated to a shared set of d Gaussians ($d \ll K$) in each of the F frequency bands with a mapping $M_f(\cdot)$. Now the distribution of the GMM has become,

$$p(S_c) = \prod_{f=1}^{F} \sum_{k=1}^{d} \phi_k \mathcal{N}(S_{c,f}; \mu_{c,k,f}, \sigma_{c,k,f}) \tag{2.19}$$

where $\mu_{c,k,f}$ and $\sigma_{c,k,f}$ are the mean and variance of the k^{th} Gaussian for the f^{th} dimension of the log power spectrum for c^{th} speaker. $\mathcal{N}(S_{c,f}; \mu_{c,k,f}, \sigma_{c,k,f})$ is the Gaussian probability density function.

HMM is a probabilistic model for sequential data, i.e., speech, which generates unobserved random state sequence by a hidden Markov chain. Each state has an observation that forms an observation sequence. Every point in the sequence is corresponding to a time frame in speech. The parameters of a HMM includes initial state probability distribution, state transition probability distribution and observation probability distribution. The observation probability function in each state is usually a GMM.

To overcome the limitation of NMF without exploiting temporal dynamics of a speech, GMM-HMM approaches [23–25, 60, 61] were studied as a general model to separate a single channel mixture signal. On the 2006 two-talker speech separation and recognition challenge, GMM-HMM generative models outperform CASA

and NMF approaches. When a factorial hidden Markov model (FHMM) [65] was applied, the generative separation model achieved the best performance and even outperformed human's performance on multi-talker speech recognition [24, 66].

Specifically, a FHMM models the acoustic dynamics and grammatical dynamics via state transition probabilities and a GMM models the observation probability distribution of the mixture signal. Speaker dependent GMM-HMM models are first trained with single source data. Each source is modeled with an HMM trained on the data belonging to this source. The trained models are then combined in a FHMM framework, which includes two or more Markov chains evolving independently. The observation of the mixture signal Y at each frame depends on the states of all Markov chains.

Given the observation of the mixture signal, a joint underlying state sequence is obtained over all Markov chains by a 2-D Viterbi search algorithm. Each time-frequency bin of the mixture is estimated to be dominated by one speaker. It forms a binary mask to separate the mixture signal.

With the increasing of number of Gaussian mixtures, GMM-HMM based speech separation improves the performance. However, the computation complexity increases explosively along with the increase of the number of Gaussians. To address the chicken-and-egg problem, a method that measures the Kullback–Leibler divergence between two GMMs is proposed in [67]. Under this framework, the separation efficiency is significantly improved without degrading the separation performance. Since each speaker needs to have a trained HMM model, GMM-HMM remains as a speaker-dependent system and couldn't generalize well to speakers that are unseen during training. In addition, the complexity of the system would be increased dramatically when the number of sources is increased.

2.3 Deep Learning based Speech Separation Technologies

In the deep learning era, the performances of speech enhancement and separation tasks have been significantly improved by the powerful learning ability of neural networks with deep structure. For instance, the deep neural network learns a

mapping between the noisy or mixed speech and its corresponding clean speech during training by a variety of neurons with non-linear activation function. In the inference stage, the noisy speech or mixed speech is transformed into the clean speech space by the learned mapping.

Two types of spectral mapping are first exploited in Section 2.3.1 and 2.3.2. Since speaker permutation problem is not solved in these frameworks, they are called as conventional regression techniques and conventional time-frequency masking techniques.

We further review several deep learning based techniques to solve the speaker permutation problem, including deep clustering (DC), deep attractor network (DANet) and permutation invariant training (PIT) in Section 2.3.3, 2.3.4, and 2.3.5, respectively. By solving the speaker permutation problem, the systems could be generalized well to unseen speakers. Since frequency-domain approaches typically cause phase estimation problem, an iterative phase reconstruction approach is summarized in Section 2.3.6. Besides of spectral mapping in frequency-domain, a time-domain audio separation network (TasNet) is also studied in Section 2.3.7.

2.3.1 Conventional Regression Techniques

The obvious idea is to perform speech separation by learning a spectral mapping between the mixed speech and corresponding clean speech of each source. The spectral mapping could be estimated using a deep learning structure, such as, deep neural network (DNN), convolutional neural network (CNN), and recurrent neural network (RNN). It's formulated as a regression task, as shown in Figure 2.2.

In the regression framework, the magnitude spectrum ($|S_c(t,f)|$) of each original speech is always used as clean target to train the networks. The model is optimized by minimizing a summed MSE between each estimated magnitude spectrum ($|\hat{S}_c(t,f)|$) and clean magnitude spectrum ($|S_c(t,f)|$) over all speakers ($c \in [1, C]$),

$$J = \frac{1}{T \times F \times C} \sum_{c=1}^{C} \sum_{t=1}^{T} \sum_{f=1}^{F} || \, |\hat{S}_c(t,f)| - |S_c(t,f)| \, ||_F^2 \tag{2.20}$$

where $||\cdot||_F$ is the Frobenius norm.

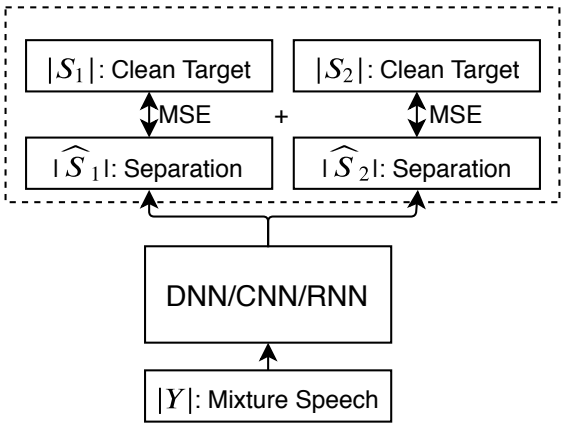


FIGURE 2.2: The regression framework of deep learning based speaker dependent systems. Two speakers' mixture is taken as an example.

However, the speaker permutation problem is not solved in this regression framework. For example, the system separates a mixed speech with speaker A and speaker B. Speaker A is forced to be the first output stream and speaker B is the second stream, as shown in Figure 2.2. Given the second mixed speech with speaker A and speaker C, speaker A and speaker C are forced to the first and second stream, respectively. Then, it causes a confusion given another mixed speech with speaker B and speaker C. Because both speaker B and speaker C would be assigned to the second stream according the learning of previous two mixed speeches (A,B) and (A,C).

The estimated magnitude $|\hat{S}_c(t,f)|$ and the phase of mixed speech $\angle Y(t,f)$ are used to reconstruct the time-domain waveform $\hat{s}_c$ by an inverse discrete Fourier transform and an overlap and add operation, as defined in Eq. 2.5 and 2.4. The phase of mixed speech is always directly used to reconstruct the separated signals, because the phase estimation still remains a challenging problem.

The regression framework may cause distortions and artifacts in the estimated magnitude, especially in low signal-to-noise ratio (SNR) condition. To address this problem, the time-frequency masking techniques in Section 2.3.2 are subsequently proposed to estimate masks that are between 0 and 1. Then, the estimated magnitude spectrum is obtained by filtering the mixture using the estimated masks.

2.3.2 Conventional Time-Frequency Masking Techniques

According to the commonly used masking approach [63, 68–74], the magnitude spectrum $|\hat{S}_c(t,f)|$ of individual source is estimated by

$$|\hat{S}_c(t,f)| = M_c(t,f) \otimes |Y(t,f)| \tag{2.21}$$

where $\otimes$ indicates an element-wise multiplication.

The commonly used masks, $M_c(t,f)$, include ideal binary mask (IBM) [72], ideal ratio mask (IRM) [73], ideal amplitude mask (IAM) [63], wiener-filter like mask (WFM) [74] and phase sensitive mask (PSM) [74].

The IBM is defined as,

$$M_c^{ibm}(t,f) = \begin{cases} 1, & if\ SNR(t,f) > \delta \\ 0, & otherwise \end{cases} \tag{2.22}$$

where $SNR(t,f)$ is the local SNR at time-frequency bin (t,f). The threshold δ is usually set to 5dB smaller than the SNR of the mixture to perserve enough speech information as suggested in [63].

Since the IBM is defined to be either 0 or 1, the time-frequency bin (t,f) is forced to belong to only one speaker. However, the time-frequency bin (t,f) of the overlapped speech can be belonging to different speakers. The IRM is subsequently proposed to predict soft mask with values between 0 and 1 as,

$$M_c^{irm}(t,f) = \frac{|S_c(t,f)|}{\sum_{c=1}^{C} |S_c(t,f)|} \tag{2.23}$$

Similarly, the WFM is defined as,

$$M_c^{wfm}(t,f) = \frac{|S_c(t,f)|^2}{\sum_{c=1}^{C} |S_c(t,f)|^2} \tag{2.24}$$

The IAM is defined as the magnitude of source speech diveded by the mixture,

$$M_c^{iam}(t,f) = \frac{|S_c(t,f)|}{|Y(t,f)|} \tag{2.25}$$

The PSM considers the phase differences between the source signals and the mixture, thus represents the best performance. The ideal PSM for each source is defined as

$$M_c^{psm}(t,f) = \frac{|S_c(t,f)| \otimes cos(\theta_y(t,f) - \theta_c(t,f))}{|Y(t,f)|} \tag{2.26}$$

where $\theta_y(t,f)$ and $\theta_c(t,f)$ are the phases of the mixture $Y(t,f)$ and speaker $S_c(t,f)$.

To estimate the parameters of the masks, we usually minimize the MSE between the estimated mask $\hat{M}_c$ and the ideal mask M_c, which is known as the mask approximation loss.

$$J = \frac{1}{T \times F \times C} \sum_{c=1}^{C} ||\hat{M}_c - M_c||_F^2 \tag{2.27}$$

However, such estimation is sub-optimal as it does not directly minimize the signal reconstruction error. Additionally, the estimated masks are undetermined for silence. Recently, an objective function of magnitude spectrum approximation shows a benefit by directly minimizing the MSE between the estimated magnitude and the true magnitude [63]. The magnitude spectrum approximation directly calculates the signal reconstruction error. The estimated magnitude ($\hat{S}_c$) is obtained by element-wise multiplying the estimated masks and the magnitude of the mixture as $\hat{S}_c = \hat{M}_c \otimes Y$. In this case, the masks are estimated using the signal reconstruction error instead of the error to the ground-truth masks, or ideal masks.

$$J = \frac{1}{T \times F \times C} \sum_{c=1}^{C} ||\hat{M}_c \otimes |Y| - |S_c|\ ||_F^2 \tag{2.28}$$

Applying PSM, we have the following objective function in calculating the signal reconstruction error,

$$J = \frac{1}{T \times F \times C} \sum_{c=1}^{C} ||\hat{M}_c \otimes |Y| - |S_c| \otimes cos(\theta_y - \theta_c)||_F^2 \tag{2.29}$$

Same as the conventional regression techniques, we directly use the phase of mixed speech $\angle Y(t,f)$ and the estimated magnitude $|\hat{S}_c(t,f)|$ to reconstruct the signal $\hat{s}_c(t)$.

In the conventional regression and time-frequency masking techniques, the speaker permutation problem arises and results in a performance degradation to unseen

speakers. A discriminative network training criterion is proposed in [49]. Specifically, a discriminative term of Kullback–Leibler divergence is added as a constraint to the objective function as defined in Eq. (2.28). However, speaker permutation problem still remains unsolved. It results in poor performance to unseen speakers.

2.3.3 Deep Clustering

The DC [26, 27] techniques were proposed to address the speaker permutation problem and showed competitive results even for unseen speakers. With the assumption that each time-frequency bin is dominated by a single speaker, the DC method uses bidirectional long-short term memory (BLSTM) to project the spectrogram of the mixture to an embedding space, in which time-frequency bins of different speakers are clustered by using K-means to obtain a mask for each speaker. The masks are then applied on the mixture to obtain the speech of each individual speaker.

In the DC framework, the embedding vectors are learned and assigned to each time-frequency bin, according to an objective function that minimizes the distances between the embeddings of within speakers, while maximizing the distances between speakers. Specifically, the D-dimensional embedding vectors $V \in \mathbb{R}^{N \times D}$ are based on a deep neural network with several BLSTM layers followed by a feed-forward layer using a global function $V = f_\theta(|Y|)$ on the entire inputs $|Y|$. N is the total number of time-frequency bins, which is equal to $T \times F$. Since the network takes the whole utterance as inputs, the global properties of the inputs are considered and thus result in permutation-independent embeddings.

We consider that the D-dimensional embedding vector v_i for each time-frequency bin $i \in [1, N]$ is unit norm as $|v_i|^2 = 1$. The estimated affinity matrix VV^T describes the correlations between different time-frequency bins by calculating the inter-product of their embedding vectors. The meaning is that the time-frequency bin of i and j are dominated by same speaker if $< v_i, v_j >$ is close to 1. Otherwise, they are belonging to different speakers. To estimate such embedding vectors that implicitly represent the attributes of speakers, we need to provide a target speaker label for each time-frequency bin. We consider a binary affinity matrix LL^T that denotes the cluster assignments in a permutation-independent way. For element i and j belong to the same speaker, $(LL^T)_{i,j}$ is set to 1, and $(LL^T) = 0$ otherwise. For any permutation matrix P, $(LP)(LP)^T = LL^T$.

Given the ground truth of the binary affinity matrix LL^T, we can learn the embeddings $V = f_\theta(|Y|)$ by minimizing the distance between the affinity matrix VV^T and LL^T, as shown in the following objective function.

$$\begin{aligned} J &= ||VV^T - LL^T||_F^2 \\ &= \sum_{i,j}(< v_i, v_j > - < l_i, l_j >)^2 \\ &= \sum_{i,j:l_i=l_j}(|v_i - v_j|^2 - 1) + \sum_{i,j} < v_i, v_j >^2 \end{aligned} \tag{2.30}$$

By minimizing the objective function J, the embeddings v_i and v_j become close for same speaker, while the embeddings are pushed apart for different speakers.

However, the K-means clustering collects each time-frequency bin that belongs to a single speaker by forming a binary mask. To overcome the drawback of a binary mask, the soft masks are estimated by an enhancement network stacked on the top of the DC system to further improve the performance in [27]. For each speaker S_c, the enhancement network takes the separated magnitude spectrum $\hat{S}_c$ and the mixture magnitude spectrum $|Y|$ together as inputs. And the outputs are a_c. A softmax is then used to form a mask M_c for source c by combing the outputs a_c across sources as follows,

$$M_c = \frac{e^{a_c}}{\sum_{c'} e^{a_{c'}}} \tag{2.31}$$

The final separation $|\overline{S}_c| = M_c \otimes |Y|$ is obtained by an element-wise multiplication between the mask and the mixture magnitude spectrum. The objective function for the enhancement network is direct signal reconstruction errors over all permutations $p \in [1, C!]$,

$$J = \min_p \sum_C || \, |S_c| - |\overline{S}_{c,p}| \, ||_F^2 \tag{2.32}$$

Since the objective function of DC is the distance between the affinity matrix VV^T and LL^T in the embedding space, as defined in Eq. 2.30, DC is trained without directly minimizing the signal reconstruction error. Although the enhancement network tries to minimize the signal reconstruction errors, the DC network doesn't benefit from the errors of the enhancement network, since they are trained separately. Thus, the jointly training of the deep clustering and enhancement network is explored by proposing a soft K-means algorithm. The soft K-means algorithm is similar to a weighted expectation maximization (EM) algorithm for a GMM with

tied circular covariance. The assignment of every embedding v_i to each centroid and updating the centroids are alternated computed,

$$\gamma_{c,i} = \frac{e^{-\alpha|v_i-\mu_c|^2}}{\sum_{c'} e^{-\alpha|v_i-\mu'_c|^2}} \tag{2.33}$$

$$\mu_c = \frac{\sum_i \gamma_{c,i} w_i v_i}{\sum_i \gamma_{c,i} w_i} \tag{2.34}$$

where $\gamma_{c,i}$ is the estimated assignment of embedding v_i to the cluster c. μ_c is the estimated centroid of cluster c. α is used to control the hardness of the clustering. The algorithm becomes close to K-means when α increases. The weight w_i is set to 0 for silence, otherwise 1.

2.3.4 Deep Attractor Network

DANet [29] was proposed to solve the limitation of the objective function defined in the embedding space in conventional DC method [26] by separating signals directly. Borrowing the idea of DC to learn a embedding vector for each time-frequency bin, DANet creates attractor points in embedding space for each source to group the time-frequency bins belonging to each source together. A mask is estimated for each source by calculating the similarity between the embeddings and the attractor point. An end-to-end training scheme is used to learn the mask by direct signal reconstruction errors, as shown in Figure 2.3.

The objective function for signal reconstruction errors is the same as in Eq. (2.28). Since the mask is related to the attractor of each source, there is no permutation problem. The mask $\hat{M}_c$ is estimated by using a D-dimensional embedding vector $V_{i,d}$ of each time-frequency bin $i \in [1, FT]$ and the attractor $A_{c,d}$ for each source $c \in [1, C]$, as follow,

$$\hat{M}_{i,c} = sigmoid(\sum_d A_{c,d} \times V_{i,d}) \tag{2.35}$$

In the training stage, the attractor for each source is defined as,

$$A_{c,d} = \frac{\sum_i B_{c,i} \times V_{i,d}}{\sum_i B_{c,i}} \tag{2.36}$$

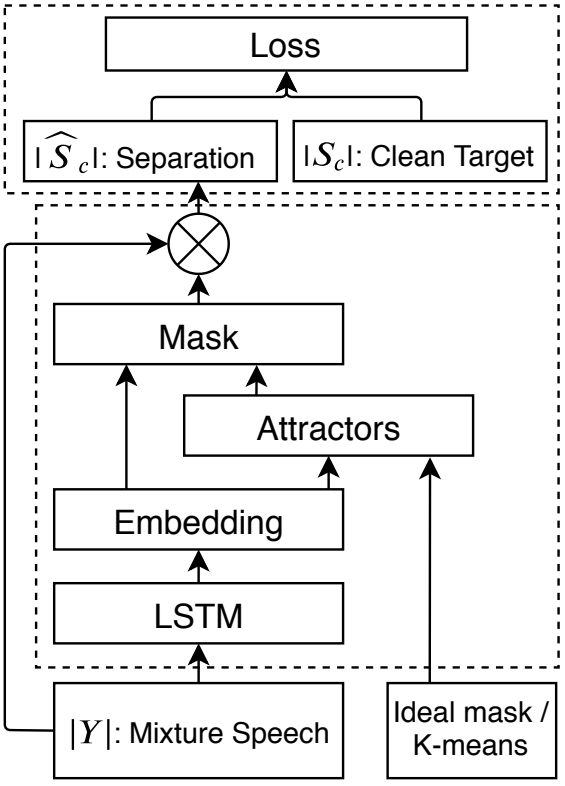


FIGURE 2.3: The system architecture of DANet method for training. During run-time testing, the upper dotted box is not necessary. The systems takes input mixture and outputs the separation.

where $B \in \mathbb{R}^{C \times FT}$ shows the source domination of each time-frequency bin. For example, $B_{c,i} = 1$ if source c dominates the time-frequency bin i at time t and frequency f with high energy comparing to other sources.

During training, the clean target speech of each source is known. We can use the true assignment B to estimate the attractor by finding the centroid of each source as defined in Eq. (2.36). Since the true assignment is unknown in the inference stage, the K-means clustering algorithm can be used to find the centers. Unfortunately, this adds the complexity of forming attractors to the run-time process with K-means clustering.

2.3.5 Permutation Invariant Training

The PIT [31] techniques represent a series of end-to-end approaches to speech separation that pool all possible permutations ($C!$ permutations) for C mixing

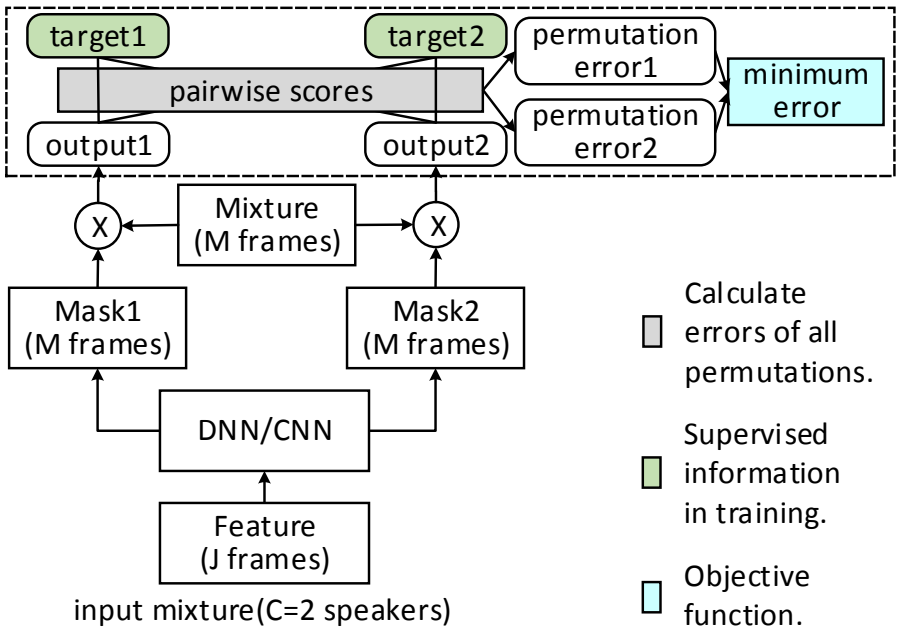


FIGURE 2.4: The framework of PIT technique using DNN or CNN for training. In run-time process, the upper dotted box is not necessary. The systems takes input mixture and outputs the output1 and output2 for the case of two speakers' mixture.

speakers. The permutation with lowest error is minimized to optimize the network in order to solve the permutation problem, as shown in Figure 2.4. The DNN with feed-forward layers or CNN are used to estimate a mask for each individual speaker frame-by-frame. The output stream of each source is obtained by filtering the multi-talker speech with the estimated mask. Since the DNN and CNN usually take several frames of the input features to capture the contextual information, different window sizes of both the inputs and outputs are investigated. The MSE between the separated and clean target spectra is calculated frame-by-frame within a contextual window for all possible permutations. The lowest MSE of the permutation is chosen as the loss to optimize the network.

However, the speaker tracking problem [32] arises during inference in the PIT technique, because the permutations of speakers may be different along the frame-by-frame separation. The switch of speaker in the output streams results in speaker leakage problem, where the frames of one speaker is separated into the stream of other speakers. The utterance-level PIT (uPIT) [32] solves the permutation problem and speaker tracking problem within an utterance by using an utterance-level training criterion with BLSTM [75, 76], as shown in Figure 2.5. BLSTM

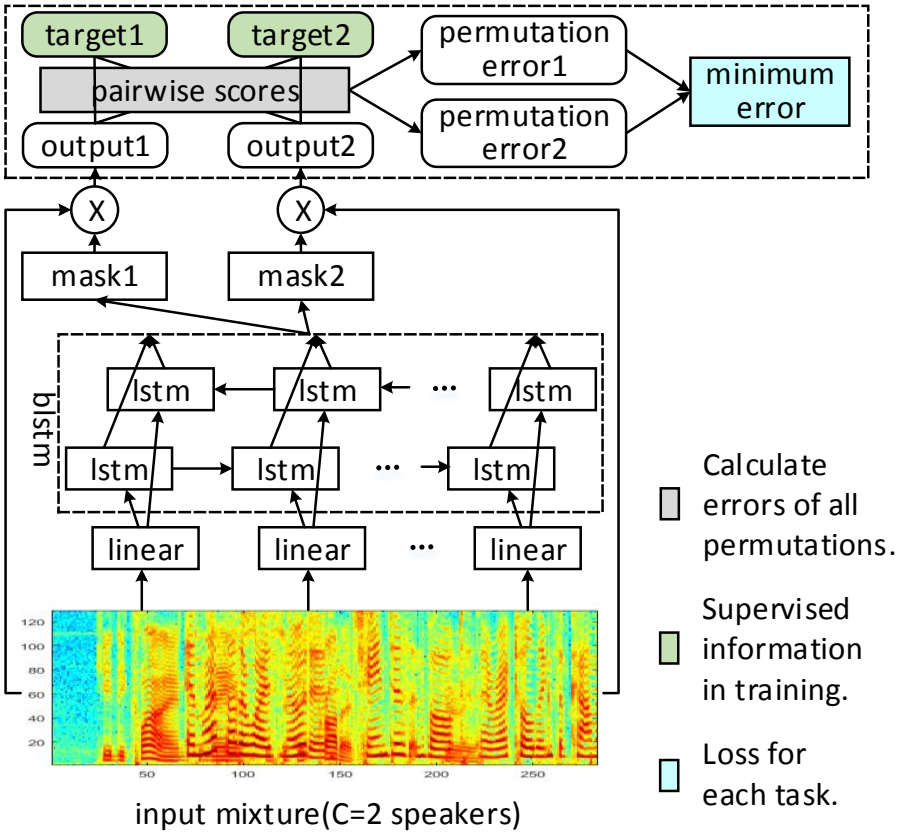


FIGURE 2.5: The architecture of uPIT technique with BLSTM for training in two speaker's mixture. In run-time process, the upper dotted box is not necessary. The systems takes input mixture and outputs the output1 and output2.

leverages the past and further information of the input through a forward pass and a backward pass. As the BLSTM has the ability to remember the contextual information by the recurrent process, there is no need to expand the frame level feature with a contextual window like DNN or CNN in PIT. Different from the PIT, the inputs and outputs of the uPIT have the same feature dimension and number of frames. Thus the dimension and number of frames of the estimated mask are also the same as the inputs. The output stream of each individual speaker is obtained by an element-wise multiplication between the estimated masks and the inputs. After obtaining the separated stream of each individual speaker, the MSEs over all permutations are calculated along the whole utterance. The lowest MSE of the chosen permutation is used as the utterance-level loss. The network is optimizing by minimizing the utterance-level loss.

Different from the conventional time-frequency masking techniques without considering the permutation of sources to calculate the loss in Section 2.3.2, the uPIT

techniques first calculate the loss of MSE over all permutations, which is defined as follows,

$$J_p = \frac{1}{T \times F \times C} \sum_{c=1}^{C} ||\hat{M}_c \otimes |Y| - |S_{c,p}| \, ||_F^2 \tag{2.37}$$

where M_c is the estimated mask for source c and $S_{c,p}$ is the assigned clean reference with permutation p.

The objective function in Eq. (2.37) is firstly optimized to find the permutation $\hat{p}$, where the loss for the network is minimal.

$$\hat{p} = \arg\min_{p} J_p \tag{2.38}$$

With the optimal $\hat{p}$, the network is optimized with the loss $J_{\hat{p}}$ to calculate the gradients and backpropagate to update the weights.

To alleviate the phase mismatch between the mixture and the target clean references, PSM is applied when we estimate the mask in the network. The objection function defined in Eq. (2.37) becomes,

$$J_p = \frac{1}{T \times F \times C} \sum_{c=1}^{C} ||\hat{M}_c \otimes |Y| - |S_{c,p}| \otimes cos(\theta_y - \theta_{c,p}) \, ||_F^2 \tag{2.39}$$

However, such utterance-level loss sums over individual frame-level errors without considering the contextual continuity of the frames, that we call temporal continuity, resulting in the frame leakage problem. The utterance-level loss may be not optimal for every frame in the whole utterance.

2.3.6 Iterative Phase Reconstruction of Speech Separation

In frequency-domain speech separation methods, the time-domain signal is always decomposed into magnitude and phase spectra. These methods always focus on improving the magnitude spectrum and ignore the phase estimation problem, because the phase spectrum shows little temporal and spectral regularities. The phase estimation remains as a challenging problem.

The Multiple Input Spectrogram Inversion (MISI) [77] algorithm iteratively estimates the time-domain signal for each source given the estimations of the source

magnitude spectra and the mixture signal. The magnitude spectra is estimated by using a Chimera network [78] that combines the loss functions of DC and PIT defined in Eq. 2.30 and 2.39. As shown in Algorithm 2, the MISI algorithm is initialised with each initially estimated source $\hat{s}_c^{(0)}$. The initial source $\hat{s}_c^{(0)}$ is reconstructed with the estimated magnitude spectrum $\hat{S}_c$ and the phase spectrum $\phi_c^{(0)}$ of the mixture signal through iSTFT. Then the error $e^{(i-1)}$ between the mixture signal y and the summation $\sum_{c=1}^{C} \hat{s}_c^{(i-1)}$ of each estimated source $\hat{s}_c^{(i-1)}$ is calculated before the next iteration. The error $e^{(i-1)}$ is divided equally and assigned to each source $\hat{s}_c^{(i-1)}$ by $e^{(i-1)}/C$, where C is the total number of sources. The estimated source $\hat{s}_c^{(i)}$ at iteration i is updated by the estimated magnitude spectrum $\hat{S}_c$ and the phase spectrum $\phi_c^{(i)}$ of the estimated signal in previous iteration and the assigned error through iSTFT. During the iterative phase reconstruction, the estimated magnitude spectrum for each source remains unchanged. The iteration process is terminated either by a threshold of the minimum error, or by a maximum number of iterations.

Algorithm 2: MISI algorithm

Input: mixture signal y, estimated magnitudes $\hat{S}_c$ for $c = 1, ..., C$, and the number of iterations K
Output: Reconstructed signal $\hat{s}_c^{(K)}$ for $c = 1, ..., C$
Initialize: $\phi_c^{(0)} = \angle STFT(y)$,
$\hat{s}_c^{(0)} = iSTFT(\hat{S}_c, \phi_c^{(0)})$, for $c = 1, ..., C$
for $i = 1, ..., K$ **do**
$e^{(i-1)} = y - \sum_{c=1}^{C} \hat{s}_c^{(i-1)}$
$\phi_c^{(i)} = \angle STFT(\hat{s}_c^{(i-1)} + \frac{e^{(i-1)}}{C})$
$\hat{s}_c^{(i)} = iSTFT(\hat{S}_c, \phi_c^{(i)})$, for $c = 1, ..., C$
end

In [28, 77], the iterative phase reconstruction is done as a post-processing. To further improve the quality of the estimated signal, the MISI algorithm is included in the objective function during training in [79]. The iterations in the MISI algorithm are unfolded as various deterministic layers on top of the mask estimation layer in a neural network. Being aware of iterative phase reconstruction in an end-to-end training, the network learns to estimate magnitudes that are optimal to the phase reconstruction steps. This is different from the post-processing in [28, 77], where the estimated magnitudes are kept same during the iterative phase reconstruction.

With the end-to-end training scheme, the objective function is defined as the error between the estimated signal and the target clean signal. As the permutation

problem arises between the output streams and the target references, the permutation invariant training is applied to minimize the lowest error among all possible permutations. The objective function is defined as

$$J = \min_{\pi \in P} \sum_{c=1}^{C} ||\hat{s}^{(K)}_{\pi(c)} - s_c||_1 \tag{2.40}$$

where K is the number of unfolded iterations as described in Algorithm 2. P is all possible permutations.

However, the iterative phase reconstruction relies on an assumption that the mixed signal is a linear summation of a set of speech signals. If the mixed signal is corrupted by reverberation and background noise, the MISI algorithm may be failed.

2.3.7 Time-domain Audio Separation Network

A conventional approach is to perform speech separation in frequency-domain, and reconstruct the time-domain signal from the filtered magnitude by time-frequency masking and estimated phase spectra. Others have also studied complex ratio mask [80–82] in speech enhancement. The frequency-domain process relies on short-time Fourier transform (STFT) that faces the windowing effect and phase estimation problem. Although the studies in Section 2.3.6 achieve good results by an iterative phase reconstruction, the phase estimation still remains as a challenging problem, especially in reverberant and noisy environment.

Time-domain speech processing has been used in various applications with the recent success of deep learning technology. For example, an end-to-end speech recognition system directly applies convolutional neural network (CNN) to the time-domain signal by transforming it into representations [83, 84]. The time-domain audio separation network (TasNet) [33–35] provides a novel solution without decomposing the signal into magnitude and phase spectra, therefore, avoiding the phase estimation problem.

Specifically, TasNet uses an encoder-decoder framework, which consists of an encoder, a separator and a decoder, as shown in Figure 2.6. The encoder encodes the time-domain signal into non-negative spectrum-like embedding coefficients by

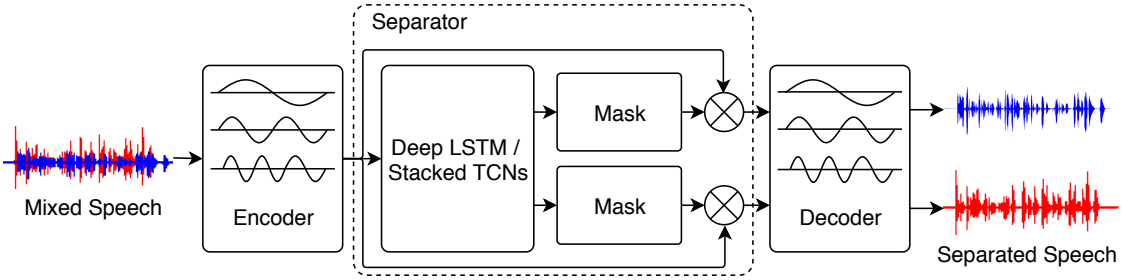


FIGURE 2.6: The general framework of TasNet technique. The separator could be a deep LSTM or a stacked temporal convolutional network (TCN) blocks. The TCN block is a dilated depth-wise separable convolution. The systems takes mixture signal in and outputs two separated source signals for the case of two speakers' mixture.

1-D convolutional neural network. The encoder replaces the STFT step to obtain a data-driven representations through the end-to-end framework. The separator tries to estimate a mask like time-frequency masking for each individual speaker. The mask estimation is conducted on the embedding coefficients from the encoder. The embedding representation for each individual speaker is obtained by applying the estimated mask to filter the embedding coefficients from the encoder. The time-domain signal of each individual speaker is reconstructed from the corresponding embedding representation with a decoder, which is a 1-D deconvolutional neural network to replace the iSTFT step.

The separated signal $\hat{s}_c(t)$ of each individual speaker is,

$$\hat{s}_c(t) = Decoder(M_c \otimes Encoder(y(t))) \tag{2.41}$$

where $E = Encoder(y(t))$ is the representations (embedding coefficients) from the encoder. M_c is the mask estimated for each speaker based on the embedding coefficients E. $Decoder(\cdot)$ is the speech decoder module to reconstruct the time-domain signal from the filtered embedding coefficients.

In [33, 34], several BLSTM layers are applied in the separator to estimate a mask for each individual speaker. To reduce the computational cost caused by deep BLSTMs, a fully convolutional TasNet (Conv-TasNet) [35] is further proposed to replace the BLSTM by stacking temporal convolutional network (TCN) blocks in the separator. The TCN block is a dilated depth-wise separable convolution. The use of depth-wise separable convolution significantly reduces the model size and

computational cost. Without recurrent step, the parallel processing on consecutive segments or frames becomes a reality to greatly speed up the separation.

The scale-invariant source-to-distortion ratio (SI-SDR) [85] is used as the objective function, which is calculated on signal level between the estimated signal $\hat{s}_c(t)$ and clean source $s_c(t)$. Since the permutation problem still exists between the estimated and clean sources, the permutation invariant training is deployed by finding the maximal SI-SDR. Then the best permutation is determinated. The network is optimized by minimizing the negative SI-SDR with the best permutation. By avoiding the phase estimation in frequency domain approaches, TasNet always outperforms its counterparts with STFT and iSTFT as the speech encoder and decoder. However, time-domain speech separation systems are sensitive to temporal distortions caused by reverberation and noise. In addition, the global speaker permutation ambiguity problem needs to be addressed for the speech separation systems, where the output stream of one speaker may switch to the other stream across frames and utterances in an online and an offline systems, respectively.

2.3.8 Speaker Beam for Target Speaker Recognition

The SpeakerBeam [86] addresses the speech recognition problem of a target speaker in a multi-talker speech. One of the solution is to firstly extract the target speaker's voice from the multi-talker speech with SpeakerBeam frontend (SBF) method using IBM as supervision (SBF-IBM). Then, the extracted speech is recognized by a speech recognizer that is trained with single speaker's speech. The SBF-IBM method is illustrated in Figure 2.7.

The SBF-IBM method exploits auxiliary speaker information from a reference speech $x(t)$ of the target speaker s. The auxiliary speaker information is encoded as adaptation weights in the adaptation layer of the context adaptive deep neural network (CADNN) [87]. The adaptation weight α_n is derived from the reference speech as,

$$\alpha = \frac{1}{T_X} \sum_{t=1}^{T_X} g(|X(t,f)|) \tag{2.42}$$

where $\alpha = [\alpha_1, ..., \alpha_N]$ is a vector used as adaptation weights for the target speaker s. $|X(t,f)|$ is the magnitude spectrum features of the reference speech $x(t)$ of the target speaker. This reference speech is different from the signal $s(t)$ of the target

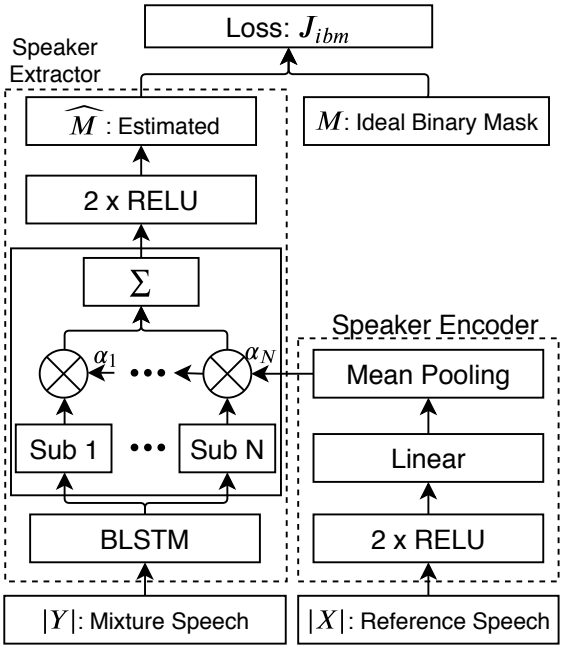


FIGURE 2.7: The framework of the SBF-IBM method to extract the target speaker's voice from a multi-talker speech given the speaker information encoded by the speaker encoder network. ""Sub 1" to "Sub N" are the number of sub-layers that have same network configuration. α_1 to α_N are the adaptation weights that are associated with the target speaker.

speaker in the mixture. T_X is the number of frames of the magnitude spectrum $|X|$. $g(\cdot)$ is a speaker encoder network.

The adaptation layer consists of a weighted sum of sub-layers. Each sub-layer is an affine transformation with same configuration. The sub-layer is defined as,

$$h_n^i = W_n \times h^{i-1} + b_n \tag{2.43}$$

where W_n and b_n are weights and biases of the affine transformation. h_n^i is the activation of the sub-layer n at the current adaptation layer i. h^{i-1} is the activation from the previous BLSTM layer.

Then, the activation of the adaptation layer is defined as,

$$h^i = \sigma(\sum_{n=1}^{N} \alpha_n \times h_n^i) \tag{2.44}$$

where N is the number of sub-layers. α_n is the weight associated with the target speaker, defined in Eq. 2.42. h_n^i is the activation of the affine transformation of

the sub-layer n, defined in Eq. 2.43. $\sigma(\cdot)$ is a non-linear activation function.

To optimize the speaker encoder and speaker extractor networks, the SBF-IBM method applies a mask approximation loss with the IBM as the supervision. The loss is defined as,

$$J_{ibm} = \frac{1}{T}\sum ||\hat{M} - M||_F^2 \tag{2.45}$$

where T is the number of frames of the mixture in frequency-domain after STFT. $\hat{M}$ and M are the estimated mask and the IBM, respectively. $||\cdot||_F$ is the Frobenius norm.

However, the SBF-IBM method has some drawbacks. First, the IBM may not be a good choice, because it only has binary values, either 0 or 1. The mask for the target speaker may be a value between 0 and 1, which is neither 0 nor 1. Second, a smaller error on mask may not result in a smaller error on signal itself, for example, magnitude. Third, the loss doesn't consider the context information across frames. Forth, the adaptation layer needs many sub-layers, which may result in heavy computation cost and parameter explosion. Finally, the speaker encoder doesn't leverage the contextual information of the reference speech to obtain the adaptation weights.

2.4 Summary

This chapter firstly introduces the problem of speech separation and its definition. The mixture signal is defined to consider the real world environment with reverberation and noise. Then the recent well-known corpus with various mixture conditions are introduced. The simulated mixture condition with its signal model are discussed. The objective and subjective evaluation metrics are further introduced. Finally, the representative speech separation works before deep learning era and the deep learning based methods are summarized, individually.

Chapter 3

Multi-task Learning of Neural Networks for Temporal Continuity in Speech Separation

This chapter focuses on solving the frame leakage problem of time-frequency masking based speech separation methods. Our solution is to improve the temporal continuity by considering the temporal information in the objective function and capturing spectro-temporal features with a grid LSTM. Section 3.1 recaps the speech separation with time-frequency masking and introduces the frame leakage problem. Section 3.2 introduces the proposed speech separation framework with the temporal objective function and the spectro-temporal features. Then, a novel multi-task learning framework is applied to further improve the performance of speech separation in Section 3.3. Section 3.4 and 3.5 describe the experimental setups and results. Section 3.6 concludes this work.

The work in this chapter has been published in [39, 40].

3.1 Recapping of Time-frequency Masking based Speech Separation

As discussed in Section 2.1.1 and Section 2.3.2, a general speech separation approach is to transform the time-domain mixture signal into a frequency-domain processing via STFT. With the magnitude spectrum, a neural network, i.e, DNN, or LSTM, is adopted to estimate a mask for each speaker in the mixture. The estimated magnitude spectrum of each speaker is obtained by filtering the magnitude spectrum of the mixture signal with the estimated mask. The signal approximation loss (i.e., MSE between the estimated and target clean magnitude spectra) is applied to optimize the mask estimation network. The conventional time-frequency approach sums the errors over the pairs of the estimated outputs and the targets without finding the optimal permutation assignment, as discussed in Section 2.3.2. The speaker permutation problem arises in the conventional time-frequency approach and results in a generalization problem to unseen speakers.

The permutation invariant training [31, 32] addresses the speaker permutation problem by exploring all possible permutations and finding the best one with lowest error, as discussed in Section 2.3.5. The error is also a MSE between the estimated and the target clean magnitude spectra, which is calculated on either frame level in PIT-DNN [31] or utterance level in uPIT-BLSTM [32]. The PIT-DNN obtains the best speaker assignment frame by frame in the training stage. The frame based processing causes a speaker tracking problem during reference, because the speaker assignment to the following frames may be different from the assignment in the previous frame. The different speaker assignment between frames causes a speaker switch in the output streams and breaks the temporal continuity of the separated speech between frames. The uPIT-BLSTM solves the speaker tracking problem by calculating an utterance based loss to optimize the network and forcing the speaker assignment of all the frames in the whole utterance to be the same. However, the utterance based loss may be not optimal on each frame and causes a frame leakage problem, where some frames belonging to the output stream of one speaker are wrongly assigned to another speaker's output stream.

3.2 Temporal Objective Function and Grid LSTM Spectro-Temporal Feature

To ensure the temporal continuity in the uPIT-BLSTM [32] framework, we first propose temporal objective functions by computing MSE between dynamic features of estimated magnitudes and clean target magnitudes. The dynamic features are obtained with a contextual window. They are considered to capture the temporal structure of the speech. By optimizing the network with the temporal objective function, the separated speech would be temporal continuous. Inspired by the heuristic patterns (i.e., common onset or offset) in the CASA approaches, we further propose a grid LSTM that automatically learns the spectro-temporal patterns from the input mixture simultaneously. With the combination of temporal objective functions and the grid LSTM spectro-temporal feature, we hope to address the frame leakage problem by taking both spectral and temporal patterns into consideration.

3.2.1 Temporal Objective Functions

Dynamic features [88, 89] and the shifted delta cepstral features [90, 91], which consists of the delta cepstral features across multiple frames of speech, have been proven effective as features in speech recognition [88, 92] and language recognition [91, 93]. Because they benefit from the temporal information of the speech. In this chapter, we propose the use of objective functions based on such temporal coefficients or dynamic features (i.e., delta, acceleration, or SDC) instead of static coefficients (i.e., magnitude) in monaural speech separation. We estimate the PSM for each individual source using phase sensitive spectrum approximation. To the best of my knowledge, dynamic features are the first time to be used in the objective function of a speech separation task for temporal continuity.

The dynamic features explore the temporal information of speech spectrogram over a contextual window. SDC captures the temporal information from even a longer speech window, for example, 11 frames in this work. In this way, the spectral transitions in the phonemes or even syllables can be included in the analysis window. Different from the magnitude based objective functions in [31, 32], the temporal objective functions ensures that the temporal continuity of a speaker

is explicitly taken into consideration in the speech separation task. If there is a speaker switch in the separated output stream, such switch is reflected in the computation of temporal coefficients by resulting in a higher output error. With the temporal coefficients in the objective function, such speaker switch is minimized.

The computations of acceleration and SDC are based on the delta coefficient ($f_d(v(t))$) [88] that can be calculated as follows,

$$f_d(v(t)) = \frac{\sum_{l=1}^{L} l \times (v(t+l) - v(t-l))}{\sum_{l=1}^{L} 2l^2} \tag{3.1}$$

where L is set to 2 in this study. $v(t)$ is the spectral feature vector from the frame t of speech. The acceleration coefficient is defined as,

$$f_a(v(t)) = \frac{\sum_{l=1}^{L} l \times (f_d(v(t+l)) - f_d(v(t-l)))}{\sum_{l=1}^{L} 2l^2} \tag{3.2}$$

The SDC $f_{sdc}(v(t))$ [90] extends the delta coefficient by concatenating K (i.e., 4) blocks of delta coefficient with a shift of I (i.e., 2) in this work. For kth coefficients, we have

$$\begin{aligned} f^k_{sdc}(v(t)) &= f_d(v(t + (k-1)I)) \\ &= \frac{\sum_{l=1}^{L} l \times (v(t + (k-1)I + l) - v(t + (k-1)I - l))}{\sum_{l=1}^{L} 2l^2} \end{aligned} \tag{3.3}$$

where $k \in \{1, K\}$. In this way, the SDC vector expands multiple frames (i.e., 11 frames in this work) and contains

$$f_{sdc}(v(t)) = [f_d(v(t)), f_d(v(t+I)), \ldots, f_d(v(t + (K-1)I))] \tag{3.4}$$

We estimate the PSMs of individual speakers in a manner of phase sensitive spectrum approximation by directly minimizing the signal reconstruction error. The temporal objective functions are defined as,

$$J_{*,p} = \frac{1}{T} \sum_{c=1}^{C} ||f_*(\hat{M}_c \otimes |Y|) - f_*(|S_{\phi_p(c)}| \otimes cos(\theta_y - \theta_{\phi_p(c)}))||^2_F \tag{3.5}$$

where $\phi_p(c), p \in [1, P]$ is an assignment of target source (c) to an output stream, and $P = C!$ is the total number of possible permutations. $f_*(\cdot)$ represents the function of $f_d(\cdot)$, $f_a(\cdot)$, and $f_{sdc}(\cdot)$. And the corresponding temporal losses for

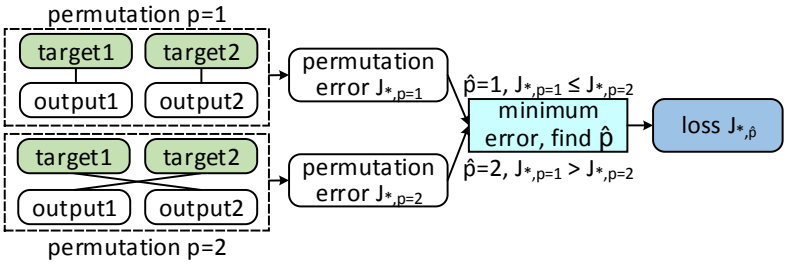


FIGURE 3.1: The computation of the loss with objective functions of speech features that minimizing the permutation error.

the permutation p are $J_{d,p}$, $J_{a,p}$ and $J_{sdc,p}$. Since the loss is calculated on static feature, i.e., magnitude, in the uPIT-BLSTM method [32], the magnitude based loss ($J_{m,p}$) for the permutation p is obtained by replacing the function $f_*(\cdot)$ with $f_m(x) = x$.

The optimal assignment is done by choosing the minimal loss among all permutations of C mixing speakers ($P = C!$). For instance, the losses of $2! = 2$ permutations (error1, error2 when p=1 and 2) in the case of 2 speakers are considered in this work, as shown in Figure 3.1.

$$\hat{p} = \arg\min_{p \in P} J_{*,p} \tag{3.6}$$

And the loss of the magnitude or temporal objective functions used to optimize the network is obtained with the optimal assignment.

$$J_* = J_{*,\hat{p}} \tag{3.7}$$

We would like to study the use of the temporal losses (J_d, J_a, or J_{sdc}) instead of the magnitude based loss (J_m) in reducing speaker leakage.

3.2.2 Grid LSTM Spectro-Temporal Feature

A grid LSTM [84, 94, 95] consists of individual LSTMs that step in both time and frequency axis of the magnitude spectrum. The LSTMs along the time and frequency axis are called as the grid time LSTM (gT-LSTM) and the grid frequency LSTM (gF-LSTM). The gT-LSTM and gF-LSTM communicate internally with the activations from both previous time and frequency steps through the regression.

With the peephole operation, the states from both previous time and frequency steps are also used to calculate the weights for the input, forget and output gates. In the time axis, the gT-LSTM moves forward frame by frame iteratively associated with the gF-LSTM. In the frequency axis, we fold the frequency features ($N = 129$) with a sliding window of F (i.e., 29) and a stride of A (i.e., 10). Then, the gF-LSTM unrolls over frequency iteratively by an amount of $B = (N - F)/A + 1$. At each time-frequency step $(t, k), t \in [1, T], k \in [1, B]$, the computations of the grid LSTM in either time or frequency dimension $j, j \in \{t, k\}$ are defined as,

$$i_{t,k}^{(j)} = \sigma(W_{ix}^{(j)} y_{t,k} + W_{ih}^{(t)} h_{t-1,k}^{(t)} + W_{ih}^{(k)} h_{t,k-1}^{(k)} + W_{ic}^{(t)} c_{t-1,k}^{(t)} + W_{ic}^{(k)} c_{t,k-1}^{(k)} + b_i^{(j)}) \quad (3.8)$$

$$f_{t,k}^{(j)} = \sigma(W_{fx}^{(j)} y_{t,k} + W_{fh}^{(t)} h_{t-1,k}^{(t)} + W_{fh}^{(k)} h_{t,k-1}^{(k)} + W_{fc}^{(t)} c_{t-1,k}^{(t)} + W_{fc}^{(k)} c_{t,k-1}^{(k)} + b_f^{(j)}) \quad (3.9)$$

$$c_{t,k}^{(t)} = f_{t,k}^{(t)} \odot c_{t-1,k}^{(t)} + i_{t,k}^{(t)} \odot g(W_{cx}^{(t)} y_{t,k} + W_{ch}^{(t)} h_{t-1,k}^{(t)} + W_{ch}^{(k)} h_{t,k-1}^{(k)} + b_c^{(t)}) \quad (3.10)$$

$$c_{t,k}^{(k)} = f_{t,k}^{(k)} \odot c_{t-1,k}^{(k)} + i_{t,k}^{(k)} \odot g(W_{cx}^{(k)} y_{t,k} + W_{ch}^{(t)} h_{t-1,k}^{(t)} + W_{ch}^{(k)} h_{t,k-1}^{(k)} + b_c^{(k)}) \quad (3.11)$$

$$o_{t,k}^{(j)} = \sigma(W_{ox}^{(j)} y_{t,k} + W_{oh}^{(t)} h_{t-1,k}^{(t)} + W_{oh}^{(k)} h_{t,k-1}^{(k)} + W_{oc}^{(t)} c_{t,k}^{(t)} + W_{oc}^{(k)} c_{t,k}^{(k)} + b_o^{(j)}) \quad (3.12)$$

$$h_{t,k}^{(j)} = o_{t,k}^{(j)} \odot \sigma(c_{t,k}^{(j)}) \quad (3.13)$$

where $i_{t,k}^{(j)}, f_{t,k}^{(j)}, o_{t,k}^{(j)}$ are the weights to control the corresponding input, forget and output gates at time-frequency (t, k) for gT-LSTM $(j = t)$ and gF-LSTM $(j = k)$. $c_{t,k}^{(j)}$ represents the states of the memory cell to remember the history information internally at time-frequency (t, k) for gT-LSTM $(j = t)$ and gF-LSTM $(j = k)$. $h_{t,k}^{(j)}$ are the outputs of the gT-LSTM $(j = t)$ and gF-LSTM $(j = k)$ at time-frequency step (t, k).

Finally, the outputs of the gF-LSTM $\{h_{t,1}^{(k)}, ..., h_{t,B}^{(k)}\}$ and gT-LSTM $\{h_{t,1}^{(t)}, ..., h_{t,B}^{(t)}\}$ are concatenated at each time step t as,

$$h_t = [h_{t,1}^{(k)}, ..., h_{t,B}^{(k)}, h_{t,1}^{(t)}, ..., h_{t,B}^{(t)}] \quad (3.14)$$

The spectro-temporal features are learnt as h_t by the grid LSTM. The spectro-temporal features are given to a linear layer to reduce the number of parameters and the computation cost. To further reduce the number of parameters and the computation cost, the corresponding weights in gT-LSTM and gF-LSTM are shared in Eq. 3.8 to 3.12.

3.3 Multi-Task Learning Framework

We now study a novel multi-task learning framework, named as SDC-G-MTL, for monaural speech separation with a subtask to predict the attribute for each time-frequency bin of the input mixture, as shown in Figure 3.2. Due to the time-frequency sparse characteristics of a speech signal, the time-frequency bins of the mixed speech could be classified into one of the three categories {silence, single, overlapped}. The attribute for each time-frequency bin is already available in the training. By using such attributes to supervise the training explicitly, the BLSTMs and grid LSTMs are optimized according to both the temporal objective function and a cross-entropy supervising the prediction of the attributes through a multi-tasking learning.

In this framework, each time-frequency bin is tagged with one of the attributes in {silence, single, overlapped} in the training stage, which explicitly informs the mask learning process with the exact location of the silence, single and overlapped time-frequency bin. We note that the speech separation task is to separate the overlapped time-frequency bins into individual speakers. The ideal amplitude masks for the overlapped parts are always less than 1. For the time-frequency bins tagged with "single", the ideal amplitude masks are 1 for the speaker that they belong to, otherwise, they are 0. As the time-frequency attributes are directly related to the masks, we believe that by incorporating time-frequency attributes as a learning sub-task, we improve the quality of the mask estimation.

A softmax function is applied to 3 hidden nodes, which predict the attribute for each time-frequency bin in the subtask. For each frame, there are 129 frequency bins. In total, the number of hidden nodes in the output layer is $387(= 3 * 129)$. The cross entropy loss is calculated over frames.

$$J_{ce} = -\frac{1}{T}\sum_{t=1}^{T} g_t^H \times \log \hat{g}_t \tag{3.15}$$

where $g_t \in \mathbb{R}^{387\times 1}$ is the true probability vector over all frequency bins at frame t. And $\hat{g}_t$ is the predicted probability vector over all frequency bins at frame t.

Then the multi-task learning objective function is defined as the weighted sum of the SDC loss and cross-entropy loss. And the network in the multi-task learning

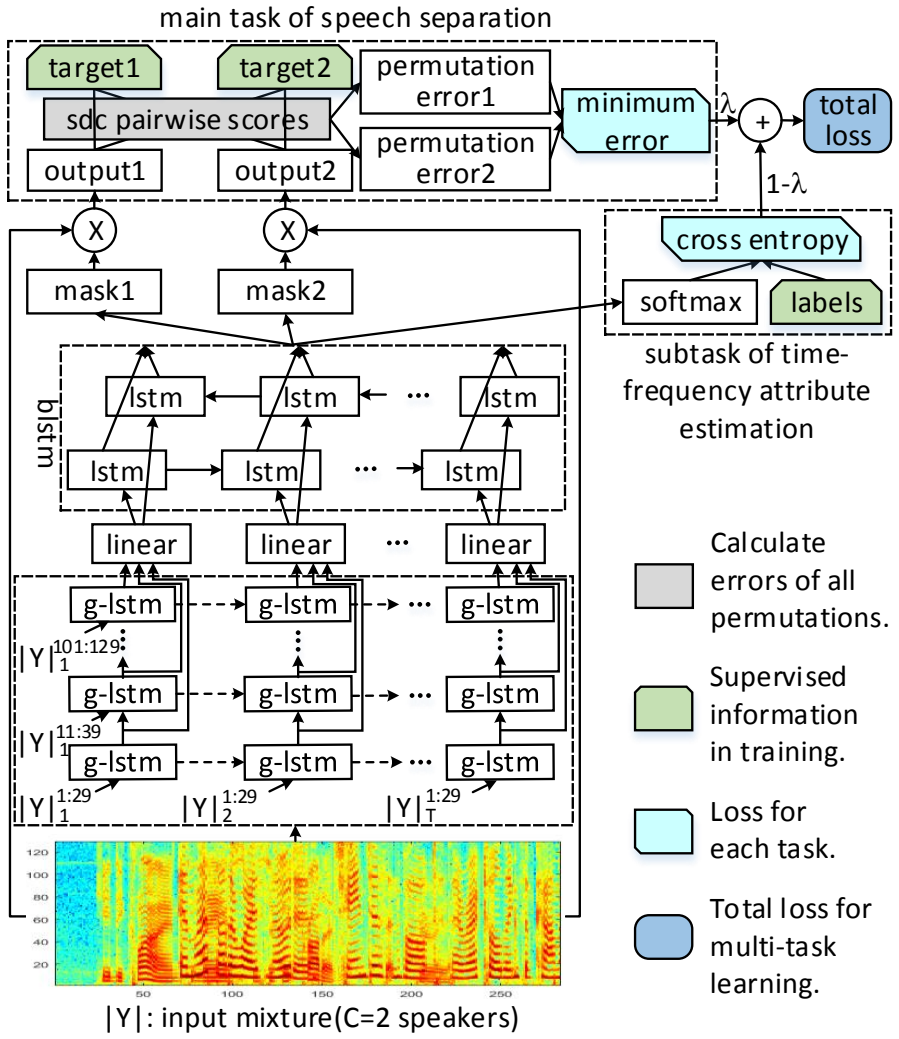


FIGURE 3.2: The training schema of the proposed multi-task learning framework for monaural speech separation. At run-time, the upper dotted box and the subtask in the upper right dotted box are not necessary. The system takes the input mixture and separates it into output1 and output2, where two-speaker mixture is taken as an example.

framework is optimized from scratch with the loss.

$$J_{mtl} = (1 - \lambda) \times J_{sdc} + \lambda \times J_{ce} \tag{3.16}$$

where the λ is the weight to balance the two loss functions.

3.4 Experimental Setup

This section describes the details of the experimental setup, including the dataset, network setting and evaluation metrics.

3.4.1 Dataset and Network Configuration

The proposed methods are evaluated on the well-known two-talker mixed WSJ0-2mix dataset, as discussed in Section 2.1.2.1. A normalized square root of 32ms hamming window with a shift step size of 16ms is applied for STFT analysis. Both the FFT length and frame length are 256 samples. The 129-dim spectral magnitude features were obtained as input features. Since the PSMs are estimated for each speaker with a signal approximation loss, the magnitudes of the two clean speakers' signals together with phase differences between the clean speeches and the mixture were obtained as the targets to train the network.

To obtain the true time-frequency attributes ({silence, single, overlapped}) as the targets in the subtask, we firstly label each time-frequency bin of each speaker's clean speech with "1" and "0" tags in the training stage. The time-frequency bin of each speaker's clean speech is labelled as "1" if the magnitude is greater than a threshold that is 40dB below the maximum [26]. If the corresponding time-frequency bins of both speakers are labelled as "1", the attribute for this time-frequency bin of the input mixture is assigned as "overlapped". If only one is labelled as "1", the attribute is "single". Otherwise, the attribute of "silence" is given.

For a fair comparison with the previous work [32], 3 BLSTM layers with 896 units in each layer were kept the same as our proposed architecture. The units of the grid LSTM cell were set to 64 and the input dimension and shift were 29 and 10 for the grid frequency LSTM. The outputs of the grid LSTM [84, 94, 95] were reduced by a linear layer from $1,408$ to 896. A random dropout with a dropout rate of 0.5 was used in the the BLSTM layers[1]. The ReLU activation function was applied in the mask estimation layer. Since there were 129 frequency bins and each frequency bin had 3 possible tags for every frame, the output layer of

[1]The recurrent dropout was not applied across time steps for fairly comparisons, although it was known to be effective and used in [27].

the subtask had $387 (= 3 * 129)$ nodes. The learning rate was initialized as 0.0005 and scaled down by 0.7 when the training loss increased on the development set. Each minibatch had 16 randomly selected utterances. The number of minimum epoch was set to 30 and the early stopping criterion was that the relative loss improvement was lower than 0.01. The model was optimized with Adam algorithm [96] and implemented using Tensorflow[2].

3.4.2 Evaluation Metrics

We evaluate the performance using GNSDR, SIR, and SAR, as introduced in Section 2.1.3.1. In addition, a subject evaluation of A/B preference test is operated to compare our SDC-G-MTL method with the uPIT-BLSTM [32] baseline. The detail of the A/B preference test could be found in Section 2.1.3.2.

Similar to word error rate in speech recognition, this work proposes a frame leakage error rate (FLER) to report the extent of the frame leakage problem in speech separation when the target references are available,

$$FLER = \frac{N_I + N_D + N_S}{N} \tag{3.17}$$

where N_I, N_D, N_S is the number of insertions, deletions and substitutions, respectively. N is the total number of frames in the separated stream including silence frames. We count one insertion when a silence target frame is assigned with a speech frame as a result of the separation. We count one deletion when a target speech frame is wrongly assigned with a silence frame. One substitution is reported when the one speech frame from a speaker is wrongly separated to the output stream of another speaker.

When we count the type of errors (insertion, deletion, or substitution), we firstly decide which speaker the separated output streams belongs to. For example, if there are two separated output streams (o_1 and o_2) and two clean target speeches of two speakers (s_1 and s_2), we calculate the MSE over all the permutations (p_1: ($o_1 \rightarrow s_1$, $o_2 \rightarrow s_2$), and p_2: ($o_1 \rightarrow s_2$, $o_2 \rightarrow s_1$)) between the output streams and the clean speeches over an utterance. If the MSE of permutation p_2 is smaller, we

[2]https://www.tensorflow.org/

assume the output stream o_1 belongs to speaker s_2. Meanwhile, the output stream o_2 belongs to speaker s_1.

Then an energy based voice activity detection (VAD) is applied on each separated and clean target speech to obtain the voiced and silent frames, respectively. Finally, the distance (MSE) of the separated frame is calculated by comparing with the corresponding frames in the clean target speech of speaker s_1 and speaker s_2, respectively. For example, if the frame from the separated output stream o_1 is closer to the reference frame of speaker s_1 than speaker s_2, the separated frame belongs to speaker s_1. But this separated output stream has classified as speaker s_2 in previous stage. A substitution error is thus counted when the separated and reference frames are voiced frames. If the reference frame is a silent frame and the separated frame is a voiced frame, an insertion error is counted. If the reference frame is a voiced frame and the separated frame is a silent frame, a deletion error is counted.

3.5 Experimental Results

3.5.1 Effect of Temporal Objective Functions

We would like to obtain the parameter settings for delta coefficient and SDC that are used in the temporal objective functions (J_d and J_{sdc}). Figure 3.3 shows the performances of speech separation by tuning the parameter L that indicates the contextual window size in delta coefficient calculation in Eq. 3.1. The parameter L is tuned with the delta based objective function J_d on the development set in terms of GNSDR. Meanwhile, the network structure is kept same as the uPIT-BLSTM baseline. From Figure 3.3, we observe that the best performance of speech separation is achieved when L is set to 2 in delta-based objective function (J_d) in Eq. 3.5. It is also same as the configuration of delta and acceleration based dynamic features in the speech processing, such as, MFCC feature with delta and acceleration ($L = 2$) for speech recognition [97] and speaker verification [98]. The same setting ($L = 2$) is used in the acceleration and SDC calculation in Eq. 3.2 and Eq. 3.3.

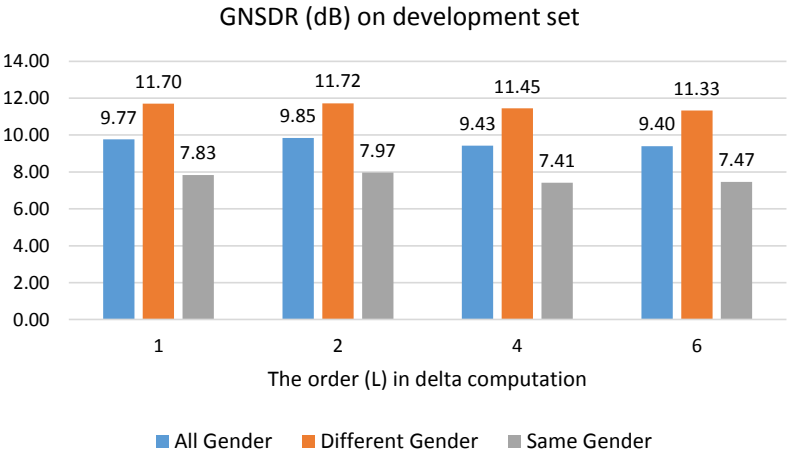


FIGURE 3.3: The GNSDR (dB) with different order L of the delta coefficient (See Eq. 3.1) objective function on the development set. 'All gender' is the overall result of the development set that has both different gender and same gender speakers in the mixtures. 'Different gender' is evaluated on the subset of the development set that only has different gender speakers in the mixtures (female and male mixture). Similarly, 'Same gender' is evaluated on the other subset of the development set that only contains same gender speakers in the mixtures (female and female mixture, male and male mixture).

We further tune the parameters I and K for SDC in Eq. 3.3, which is used in the SDC-based objective function (J_{sdc}) in Eq. 3.5. We report the speech separation performance for different I and K settings in Figure 3.4 on the development set in terms of GNSDR. The network structure is still kept same as uPIT-BLSTM baseline in order to fairly comparing with magnitude or delta based objective function. To tune K, we firstly fix I to 3 and observe that $K = 4$ provides the best result. Then we fix K to 4 to tune the parameter I. With this parameter tuning scheme, the best GNSDR (dB) is achieved when I is 2 and K is 4. With $L = 2$, $I = 2$ and $K = 4$, the SDC at frame t is $[f_d(t), f_d(t+2), f_d(t+4), f_d(t+6)]$ following Eq. 3.4. Since L is set to 2, the SDC expands the long contextual temporal information to a span of 11 frames (192ms in this work).

To evaluate the effectiveness of contextual information in temporal objective functions (J_d, J_a, and J_{sdc}), we compare them with the magnitude based objective function (J_m) in the uPIT-BLSTM baseline [32]. Table 3.1 suggests that the proposed delta and SDC objective functions based systems significantly outperform the uPIT-BSLTM baseline with a magnitude based objective function at the significant value p (< 0.05). Although BLSTM is already capable of modelling the temporal structure in the input space which should also be reflected in the output

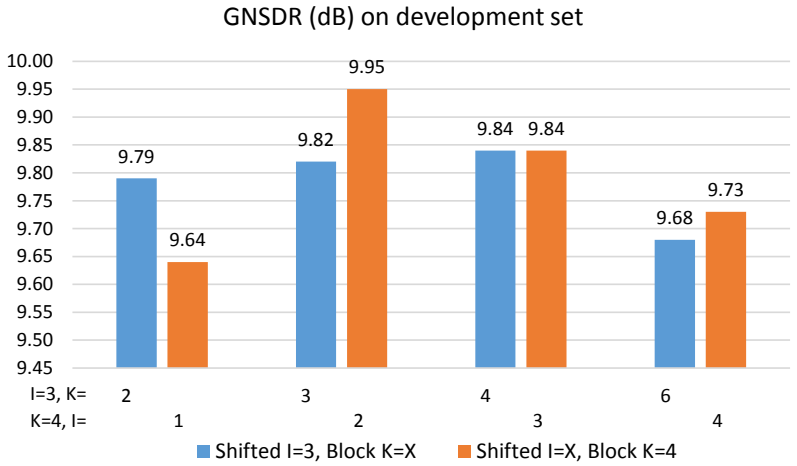


FIGURE 3.4: The GNSDR (dB) with different shift I and block K with the SDC objective function on the development set.

TABLE 3.1: A comparison of GNSDR, SIR and SAR (dB) with 95% confidence intervals over different objective functions calculated on magnitude, delta, acceleration and SDC on WSJ0-2mix test sets using 3 BLSTM layers same as the network configuration in the uPIT-BLSTM baseline. The order L is set to 2 in delta, acceleration and SDC calculation. The shift I and block K are 2 and 4 when computing SDC. w_d, w_a and w_{sdc} are tuned to be 4.5, 10.0 and 5.0.

Sys.	Objective Function	GNSDR	SIR	SAR
1	Magnitude, J_m[32]	9.5±0.1	16.5±0.1	11.2±0.1
2	Delta, J_d	9.7±0.1	16.9±0.1	11.3±0.1
3	Acceleration, J_a	9.6±0.1	16.7±0.1	11.1±0.1
4	SDC, J_{sdc}	9.8±0.1	17.1±0.1	11.3±0.1
5	J_m+w_d*J_d+w_a*J_a	9.8±0.1	17.1±0.1	11.4±0.1
6	J_m+w_{sdc}*J_{sdc}	9.8±0.1	17.1±0.1	11.4±0.1

space, the experimental results show that the performance could be further improved by adding the temporal information into the objective function. Because the temporal structure of the reference speech is explicitly captured in the temporal objective functions to constrain the output to be temporally continuous. It is noted in Figure 3.2 that the objective functions are used to optimize the system to produce good speaker separation masks during training. Mask estimation has been a focus point to study in speech separation. An interesting finding in our research is that the temporal objective functions clearly outperform magnitude based objective function in terms of mask estimation. The results in Table 3.1 have validated our proposal of temporal objective functions presented in Section 3.2.1.

As delta and SDC capture the differences between speech frames in a near neighborhood, by minimizing the mean square errors with uPIT, the frames with minimal differences are separated into the same output stream. Only when the frames are from the same speaker, the minimal differences are obtained, thus the speaker temporal continuity is ensured. Additionally, as SDC is calculated over a longer range of contextual window than that of delta coefficient, SDC-based objective function shows a better separation performance than delta-based objective function. We notice that the acceleration-based objective function is not as effective as other dynamic features. The main reason may be that the acceleration coefficient is the second order dynamics while the delta coefficient is the first order dynamics. The minimizing of the acceleration coefficient is to optimize network to make the delta coefficient temporal continue, instead of the magnitude (static features) directly.

By integrating the static and dynamic objective functions as System 5 and System 6 in Table 3.1, we observe that the systems with the weighted objective functions outperform the single feature objective function, either based on static or dynamic features. Since the average value of static features is greater than that of each delta, acceleration and SDC features, the weights w_d, w_a and w_{sdc} are tuned to be great than 1, i.e., 4.5, 10.0 and 5.0 in this work. The values are determined by evaluating the performance on the development set in terms of GNSDR. As the difference of performance between the System 4 and System 6 is small, we only apply the SDC objective function to optimize the network in the following experiments.

3.5.2 Effect of Grid LSTM Spectro-Temporal Feature

Inspired by the idea of capturing time-frequency patterns with a grid LSTM followed by LSTM and DNN layers for acoustic modelling (grid-LDNN) [84], we explore to use a grid LSTM as a feature extractor followed by extra BLSTMs. We name it as SDC-G method. In [84], the grid-LDNN achieves the best performance of speech recognition among the comparisons with its counterparts LDNN, CLDNN, F-LDNN and TF-LDNN. The CLDNN, F-LDNN and TF-LDNN replace the grid LSTM layer in grid-LDNN with a convolution, frquency LSTM, and a time-frequency LSTM layer, respectively. Therefore, this work only studies the performance of adding an additional grid LSTM layer in the uPIT-BLSTM baseline.

TABLE 3.2: GNSDR, SIR and SAR (dB) with 95% confidence intervals in a comparative study with and without a grid LSTM on WSJ0-2mix test set. The SDC method refers to the network with 3 BLSTM layers, that is the same as in uPIT-BLSTM, with a SDC-based objective function (J_{sdc}). The SDC-G method further inserts one grid LSTM layer between the input and the BLSTM layers. * indicates our re-implementation

Method	GNSDR	SIR	SAR
uPIT-BLSTM* [32]	9.5±0.1	16.5±0.1	11.2±0.1
SDC	9.8±0.1	17.1±0.1	11.3±0.1
SDC-G	**10.1±0.1**	**17.4±0.1**	**11.6±0.1**

We now compare SDC-G with SDC and uPIT-BLSTM baseline. We report that SDC-G achieves 10.1dB on GNSDR, 17.4dB on SIR, and 11.6dB on SAR, respectively, as reported in Table 3.2. By adding a grid LSTM layer to learn the temporal and spectral patterns, the proposed SDC-G method significantly outperforms the SDC method by 3.1% on GNSDR, 1.8% on SIR and 2.7% on SAR with a significant value p (< 0.05). The SDC-G results validate the effectiveness of the grid LSTM in learning temporal and spectral patterns, such as, pitch continuity in time-domain. The SDC-G achieves 6.3%, 5.5% and 3.6% relatively significant improvement over the uPIT-BLSTM baseline [32] in terms of GNSDR, SIR and SAR with a significant value p (< 0.05), respectively.

3.5.3 Effect of Multi-task Learning

To make use of the attributes ({silence, single, overlapped}) of the time-frequency bins, we propose a novel multi-task learning framework that combine the SDC-based objective function (J_{sdc}) and a cross entropy (J_{ce}) objective function in the learning. The multi-task learning system is called SDC-G-MTL. The total cost function (J_{mtl}) is a weighted sum of J_{sdc} and J_{ce} using a weight λ. By tuning the weight λ between 0 to 1 on the development set, we set λ to 0.2 as shown in Figure 3.5.

From Table 3.3, we observe that our SDC-G-MTL method further significantly improves the performance relatively by 4.0% on GNSDR, 3.4% on SIR and 2.6% on SAR over the SDC-G approach that is a single task learning, respectively. The significant value is p (< 0.05). This suggests that the proposed multi-task learning framework that uses a subtask to predict the time-frequency attributes is effective

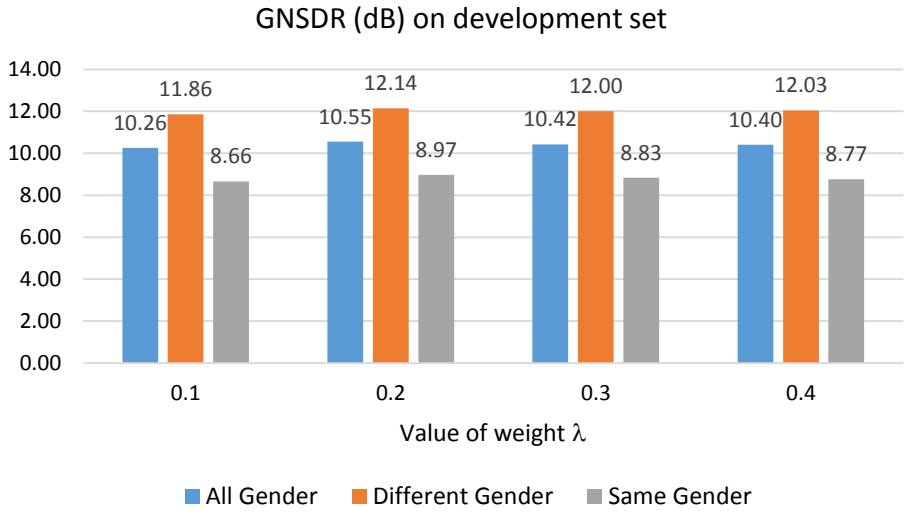


FIGURE 3.5: The GNSDR (dB) of tuning the weight λ in our SDC-G-MTL multi-task learning system using 1 Grid LSTM layer and 3 BLSTM layers evaluated under three conditions on the development set. All gender: all data are used to calculate GNSDR. Different gender: GNSDR is computed on the data that the gender of the input mixture is different. Similarly, same gender: GNSDR is computed on the data that the gender of the input mixture is same.

TABLE 3.3: GNSDR, SIR and SAR (dB) with 95% confidence intervals in a comparative study of with or without multi-task learning on WSJ0-2mix test set. The weight λ is set to 0.2 for multi-task learning systems. * indicates our re-implementation.

Method	MTL	GNSDR	SIR	SAR
uPIT-BLSTM* [32]	No	9.5±0.1	16.5±0.1	11.2±0.1
SDC	No	9.8±0.1	17.1±0.1	11.3±0.1
SDC-G	No	10.1±0.1	17.4±0.1	11.6±0.1
uPIT-BLSTM-MTL	Yes	10.0±0.1	17.3±0.1	11.5±0.1
SDC-MTL	Yes	10.1±0.1	17.6±0.1	11.6±0.1
SDC-G-MTL	Yes	**10.5±0.1**	**18.0±0.1**	**11.9±0.1**

by leveraging the explicit information of where the speech is silence, single or overlapped. The same J_{ce} subtask can also be applied to uPIT-BLSTM baseline to form a multi-task learning, that is called uPIT-BLSTM-MTL. We observe that uPIT-BLSTM-MTL obtains a similar improvement over uPIT-BLSTM baseline. Overall, the proposed SDC-G-MTL approach obtains a relatively significant improvement by 10.5% on GNSDR, 9.1% on SIR and 6.3% on SAR over the uPIT-BLSTM baseline [32], respectively.

TABLE 3.4: GNSDR, SIR and SAR (dB) in a comparative study of same and different gender combinations on WSJ0-2mix test sets (Def Assign.). 'Diff.' is evaluated on the subset of the test set that only has different gender speakers in the mixtures (female and male mixture). Similarly, 'Same' is evaluated on the other subset of the test set that only contains same gender speakers in the mixtures (female and female mixture, male and male mixture). The weight λ is set to 0.2 for multi-task learning systems. * indicates our re-implementation.

Method	MTL	GNSDR		SIR		SAR	
		Same	Diff.	Same	Diff.	Same	Diff.
uPIT-BLSTM* [32]	No	7.3	11.5	13.6	19.1	9.5	12.6
SDC	No	7.6	11.7	14.2	19.7	9.6	12.8
SDC-G	No	8.1	11.9	14.8	19.7	10.0	13.0
uPIT-BLSTM-MTL	Yes	7.7	11.9	14.4	19.8	9.7	13.0
SDC-MTL	Yes	8.0	12.0	14.8	20.1	9.8	13.1
SDC-G-MTL	Yes	**8.6**	**12.2**	**15.5**	**20.2**	**10.4**	**13.2**

3.5.4 Same vs. Different Gender

Note that systems achieve better separation for speakers of different gender than same gender. We report the same gender and different gender results separately in Table 3.4. The proposed techniques improve the performance consistently in both cases and achieve 8.6dB and 12.2dB in terms of GNSDR for same and different gender under open condition. We note that the separation of same gender speakers remains a challenge. However, we are encouraged by the fact that our proposed SDC-G-MTL method achieves a 6.1% and a 17.8% relative improvement in terms of GNSDR over the uPIT-BLSTM baseline under same and different gender mixture conditions, respectively.

3.5.5 Evaluation of Frame Leakage

Figure 3.6 shows an example of the female-female mixture separation using the uPIT-BLSTM baseline and our proposed SDC-G-MTL method. By comparing Figure 3.6 (b) and (d), we observe that some frames of speaker s_2 are wrongly separated into the output stream of speaker s_1, as shown in ellipse in Figure 3.6 (d), which is classified as insertion error. We first apply a voice activity detection to distinguish the speech frames from silence in both the target references and the separated streams. As the silence frames can go to either output stream o_1 or stream o_2, we color them in white in Figure 3.6 (h) and (i). When the target

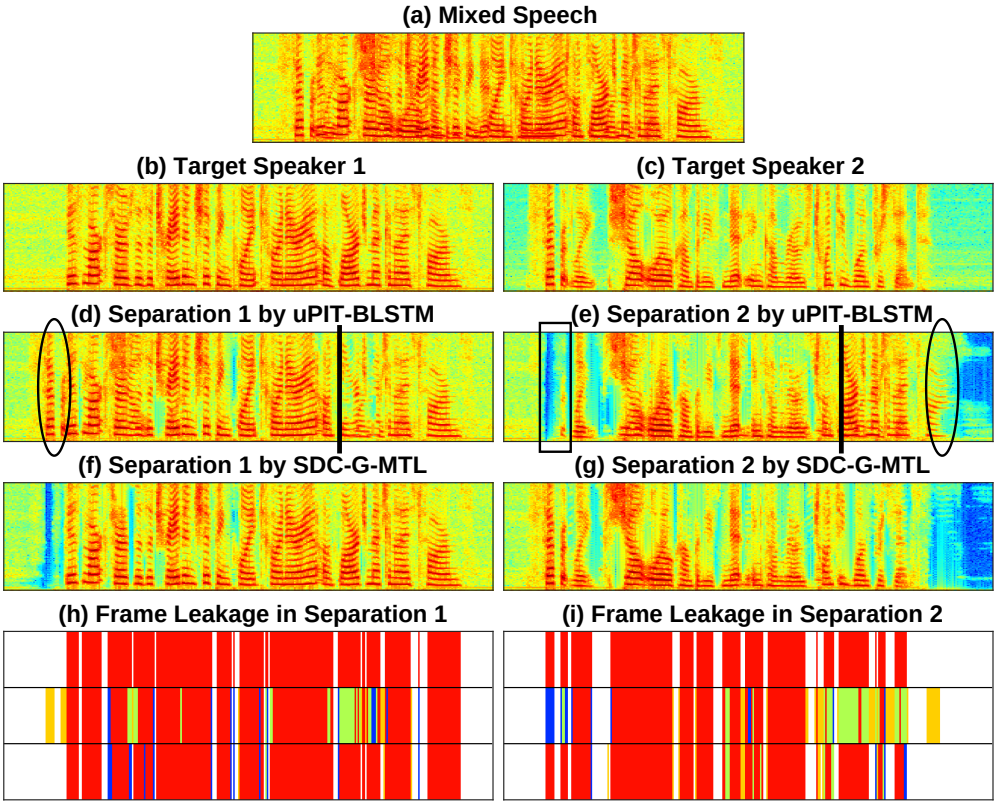


FIGURE 3.6: Spectra of the mixture, two target speech of a female-female mixed example ('050a050i_2.1935_421c020b_-2.1935') from the test set are shown in (a), (b) and (c). The spectra of two separated output streams by the uPIT-BLSTM baseline and the proposed SDC-G-MTL method are shown in (d) and (e), (f) and (g). The speech frames of target speaker are marked in red in the upper panel of (h) and (i). The frame assignments of the uPIT-BLSTM baseline are shown in the middle panel, while those of the SDC-G-MTL method are in the lower panel.

frame is speech but assigned as silence during the separation, a deletion error occurs as illustrated in rectangle in Figure 3.6 (e). When the target and separated frames are both speech and the frame level speaker label of the separated frame is different from the utterance level speaker label of this separated stream, we report a substitution error as shown in vertical line in Figure 3.6 (d).

The speech frames of the reference target speaker s_1 and s_2 are marked in red in the upper panel in Figure 3.6 (h) and (i), respectively. The frame assignments of the uPIT-BLSTM baseline are illustrated in the middle panel with insertion (yellow), deletion (blue) and substitution (green) in Figure 3.6 (h) and (i). Similarly, the frame assignments of the SDC-G-MTL method are shown in the lower panel in Figure 3.6 (h) and (i). When comparing the middle and lower panel in Figure

TABLE 3.5: FLER (%) in a comparative study of the proposed techniques and the baseline. The pitch continuity is evaluated to show the separation continuity. "VUV (%)" and "CORR" represent the voiced and unvoiced error rate and correlation calculated on the pitches between the separated and clean speech. The results are obtained on the WSJ0-2mix test set. * indicates our re-implementation.

Method	FLER	Pitch Continuity	
		VUV	CORR
uPIT-BLSTM* [32]	11.44	7.69	0.54
SDC	10.53	7.30	0.55
SDC-G	9.68	6.90	0.56
SDC-G-MTL	**8.47**	**6.44**	**0.57**

3.6 (h) and (i), we notice that SDC-G-MTL method reports a much lower frame leakage than the uPIT-BLSTM baseline. We observe that SDC-G-MTL effectively deals with the frame leakage problem. In addition, the speaker switch happens when there are insertion and substitution error. The speaker switch is significantly reduced in the SDC-G-MTL method by the SDC objective function even if the speaker switch occurs during a silence region in the uPIT-BLSTM. Because the SDC objective function ensures the continuity of the speaker within 11 frames (192ms in this work).

We summarize the FLER results of the baseline and the proposed techniques in Table 3.5. We observe that SDC-G-MTL method achieves a relative reduction of 26.0% in terms of FLER over the uPIT-BLSTM baseline, which represents a significant reduction of frame leakage. To evaluate the temporal continuity of the separated speech, we further examine the pitch continuity, where the pitch is extracted by a robust pitch extractor [99]. We calculate the voiced and unvoiced error rate (VUV) and correlation (CORR) based on the pitch contours of the separated and clean speech. We observe that the SDC-G-MTL method obtains a better pitch contour than the uPIT-BLSTM baseline.

3.5.6 Comparisons with Other Methods

Table 3.6 summarizes the performance of a series of competitive systems as the reference baselines, that include DC [26] and its variants (DC+, DC-Enh-Joint) [27], DANet [29], PIT-DNN [31], PIT-CNN [31], and uPIT-BLSTM [32] methods on the WSJ0-2mix database. DC-Enh-Joint is a two stage end-to-end system that

TABLE 3.6: GNSDR (dB) in a comparative study among the competitive methods on the WSJ0-2mix dataset with optimal frame level assignment (re-alignment with reference speeches) and default assignment (actual assignment) on closed (CC) and open (OC) conditions. † is a two-stage model by stacking. ‡ is with curriculum learning. * indicates our re-implementation. IRM and IPSM are the upper-bound performance by reconstructing the signal using IRM or IPSM with the phase of the mixture signal.

	Method	Opt Assign		Def Assign	
		CC	OC	CC	OC
Baselines	DC [26]	-	-	5.9	5.8
	DC+‡ [27]	-	-	-	10.3
	DC-Enh-Joint†‡ [27]	-	-	-	10.8
	DANet‡ [29]	-	-	-	10.5
	PIT-DNN [31]	7.3	7.2	5.7	5.2
	PIT-CNN [31]	8.4	8.6	7.7	7.8
	uPIT-BLSTM [32]	10.9	10.8	9.4	9.4
	uPIT-BLSTM-ST† [32]	11.7	11.7	10.0	10.0
Ours	uPIT-BLSTM* [32]	10.8	10.7	9.6	9.5
	SDC-G-MTL	11.4	11.4	10.6	10.5
Upper-bounds	IRM [32]	12.4	12.7	12.4	12.7
	IPSM [32]	14.9	15.1	14.9	15.1

incorporates the DC with soft clustering and an enhancement network on top. Similarly, uPIT-BLSTM-ST stacks an additional uPIT-BLSM with the mixture and separated magnitudes as inputs on top of the first uPIT-BLSTM. It's a two-stage model. The results with reference masks, i.e., IRM and IPSM [32], are also included. We observe that SDC-G-MTL achieves comparable performance with DC+ [27] and DANet [29] models, which used some advanced techniques such as recurrent dropout [100] and curriculum learning [101], and outperforms the group of reference baselines in general. We have not applied the recurrent dropout and curriculum learning techniques in SDC-G-MTL. This is to ensure a fair comparison with uPIT-BLSTM baseline. As we use the ground truth to obtain the optimal assignment, the results of the optimal assignment can be seen as the upper bound of the default assignment.

3.5.7 Subjective Evaluation

To evaluate the listening quality and intelligibility, we conduct a A/B preference test on the separated audio samples using our SDC-G-MTL method and the uPIT-BLSTM baseline by randomly selecting 20 pairs of listening examples from the

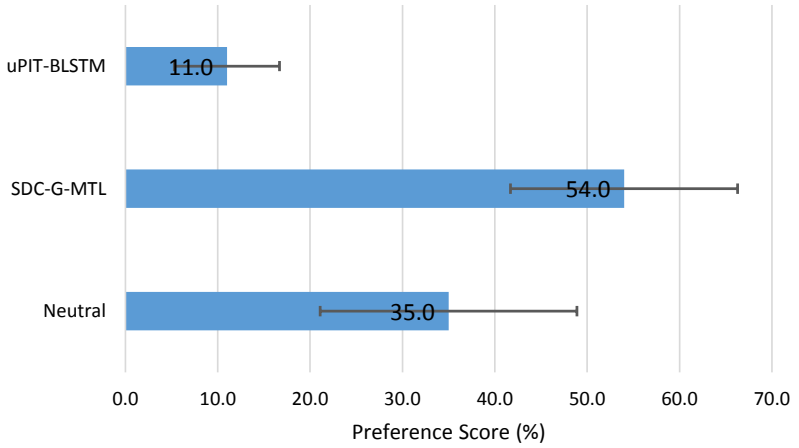


FIGURE 3.7: The A/B preference test result of the reconstructed speech waveform between the proposed SDC-G-MTL method and the uPIT-BLSTM baseline. We conducted t-test using a significance level of $p < 0.05$ which is depicted with the error bars.

reconstructed waveforms and the original test set[3]. Each pair includes 3 audio samples with same content (One is the separated output stream from our SDC-G-MTL method, another is the separated output stream from the uPIT-BLSTM baseline and the third one is the corresponding original speech.). The two separated output streams are played one-by-one randomly with original speech. A group of 10 subjects are asked to answer their preference according to the quality and intelligibility. The less interference of the other speaker means the better quality and intelligibility. The percentage preference is shown in Figure 3.7. We observe that the listeners clearly prefer the SDC-G-MTL method to the uPIT-BLSTM baseline. Most subjects prefer our SDC-G-MTL method based audio samples because there are less distortion and speaker interference than the baseline, especially in the same gender case. The examples from the SDC-G-MTL method and their corresponding original samples are available for listening[4].

3.5.8 Discussions

In this chapter, we propose the use of temporal objective functions to address the frame leakage problem. Being evaluated only on uPIT-BLSTM, the proposed objective functions are also applicable to other systems, such as DC-Enh-Joint with

[3]https://sites.google.com/site/xuchenglin28/demos/taslp18-subject-evaluation
[4]https://sites.google.com/site/xuchenglin28/demos/taslp18

end-to-end structure [27], DANet [29] and Chimera++ [28] methods, which will be further studied in the future work.

While the deep learning based methods have significantly advanced the state-of-the-art of multi-taker speech separation, the number of speaker has to be known in prior. The uPIT [32] and Chimera [28, 78] methods require such prior information during training. DC [26, 27] and DANet [29] don't need to know the number of speakers at the training stage, however they need the information during inference. Since the number of speakers is not always known in practice, the requirement of speaker information has limited the usage of the speech separation methods in real-world applications. To address this problem, we consider two research directions. One is to iteratively reconstruct the speech for speakers one after another [36]. The iteration procedure is terminated by either a stop-flag or a threshold of the residual mask. Another is to extract only the target speaker's voice from a mixture given the speaker information [86]. The solution to frame leakage problem that we discuss in this chapter is also applicable to either the iterative reconstruction or target speaker extraction frameworks.

3.6 Conclusion

In this chapter, we explore the use of temporal objective functions in uPIT for single channel speech separation. Inspired by the findings in CASA research, we propose to simultaneously learn the spectro-temporal patterns from the input mixture using a grid LSTM. A multi-task learning framework is further proposed to incorporate a subtask that supervises the learning of time-frequency attributes (silence, single, overlapped). Experimental results show that our proposed SDC-G-MTL method significantly outperforms the uPIT-BLSTM baseline by improving the temporal continuity in the separated streams. The frame leakage problem in the uPIT-BLSTM baseline is significantly reduced by the temporal objective functions, the grid LSTM, and the time-frequency attribute sub-task learning.

Chapter 4

Top-down Selective Auditory Attention with Speaker Extraction

In this chapter, we focus on addressing the problems of unknown number of speakers and the global speaker permutation problem across utterances. Our solution is to extract the target speaker's voice given a reference speech by mimicking human's ability of selective auditory attention. Section 4.1 first introduces human's ability of selective auditory attention and the motivation. The relation to speech separation is discussed in Section 4.2. Section 4.3 and Section 4.4 describe the proposed frequency-domain and time-domain speaker extraction systems, individually. Section 4.5 summarizes the work in this chapter.

4.1 Motivation

Infants, as young as five months, have developed the ability to give special attention to their own names [102]. Behavioral studies have shown that both the abilities to selectively attend to relevant stimuli and to effectively ignore irrelevant stimuli are developed progressively with increasing age across childhood [103]. As

The work in this chapter has been published in [41–43].

discussed in Chapter 1, these remarkable selective auditory attention abilities are implemented with accurate processing of low-level stimulus attributes, segregation of auditory information into coherent voices, and selectively attending to a voice at the exclusion of others to facilitate higher level processing [7].

Recent physiological studies reveal that such attentional modulation takes place both locally by transforming the receptive field properties of the individual neurons and globally throughout the auditory cortex by rapid neural adaptation , or plasticity, of the cortical circuits [38]. Computationally, the selective attention to an acoustic stimulus $E(t)$ of interest can be described by a spectro-temporal receptive field, $M(t)$, which acts as a spectro-temporal mask. The modulated response $S(t)$ to $E(t)$ can be expressed as the element-wise multiplication between the stimulus and the mask, $S(t) = M(t) \otimes E(t)$, where $M(t)$ can be seen as the modulation of the input stimulus by a top-down voluntary focus, or top-down attention.

The top-down attention tasks vary with the application scenarios, for example, the flight announcement from a busy airport, the singing vocal from a music, or the speech of particular speaker from a multi-talker acoustic environment. Different from the speech separation framework in Chapter 3, this chapter mainly focuses on how to pay a selective attention to a target speaker, that is called speaker extraction. The idea is to provide a reference speech from a new speaker that is unseen during training. The system then uses such reference speech to direct the attention to the attended speaker, that emulates human's top-down voluntary focus, as shown in Figure 4.1. By only focusing on the target speaker, speaker extraction addresses the problem of unknown number of speakers and global permutation ambiguity across utterance in speech separation.

Speaker extraction is highly demanded in real-world applications, such as, hearing aids [1], speech recognition [97, 104, 105], speaker verification[44], speaker diarization[106], and voice surveillance. A speaker independent speaker extraction system is expected to work for any target speaker unseen during the training, that we call open condition. However, machines have yet to achieve the same attention ability as humans in the presence of background noise or interference of competing speakers. The question is how to equip a network the ability to estimate the mask at run-time for a new speaker that is unseen by the system during training.

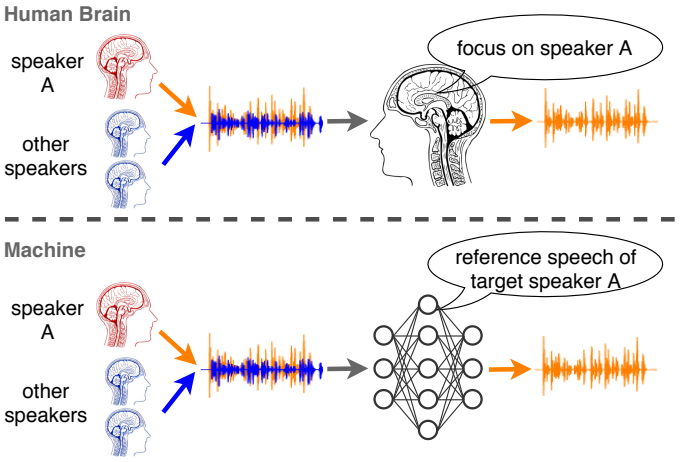


FIGURE 4.1: Emulating humans' ability of selective auditory attention with speaker extraction network, where a reference speech of target speaker is used to direct the top-down voluntary focus.

4.2 Relation to Speech Separation

Speaker extraction aims to extract the target speaker's voice $s_{c=i}(n)$ from a single channel microphone mixture signal $y(n)$ given a different speech segment $x_{c=i}(n)$ of the target speaker. Suppose that a signal $y(n)$ of N samples is the mixture of the target speaker's voice $s_{c=i}(n)$ and $C-1$ interference voices $s_{c\neq i}(n)$ with reverberation $h_{c=i}(n)$ and $h_{c\neq i}(n)$ or background noise $b(n)$. We have,

$$y(n) = h_{c=i}(n) * s_{c=i}(n) + \sum_{c\neq i}^{C} h_c(n) * s_c(n) + b(n), \quad n = 1, ..., N \tag{4.1}$$

where C might be any number of source signals, and the target speaker's voice is $s_{c=i}(n)$ indexed by i.

Compared with speech separation definition as in Eq. 2.1, speaker extraction is just a special case of speech separation, where one of the source signals is the target speaker's voice and the others are interference speech. By given the reference speech $x_c(n)$ for each speaker in parallel, speaker extraction could parallelly extract each speaker's voice in the mixture, like speech separation. The extraction process could be summarized as,

$$\hat{s}_c(n) = f(y(n), x_c(n)), \quad c = 1, ..., C \tag{4.2}$$

where $\hat{s}_c(n)$ is the estimated target speaker's voice from the mixture signal. $f(\cdot, \cdot)$ is the speaker extraction system that takes two inputs.

Given the mixed signal and enrolled reference signal, the speaker extraction system estimates $\hat{s}_c(n)$ that is close to $s_c(n)$. Different from speech separation, speaker extraction forms attention to the target speaker's voice $s_c(n)$ no matter how many interference speakers there are. The speaker extraction system only gives one output stream in a single thread with the target speaker's reference signal $x_c(n)$. The number of speakers in the mixture, which is required in speech separation, is not necessary in speaker extraction. With the reference signal $x_c(n)$ of the target speaker, the speaker extraction system always focuses on the target speaker's speech. The speaker ambiguity problem across different utterances or segments in speech separation is solved inherently in the speaker extraction. Because the output stream of the speaker extraction is always known as the target speaker's speech.

Although speaker extraction requires an additional reference speech of the target speaker comparing with speech separation, the acquisition of such reference speech is possible in real world applications through either an enrolment process or a speaker diarization system. For example, in speaker verification or personal device speech application, where the reference speech of the target speaker is available through an enrolment process or a pre-registration. Such speaker extraction technique is particularly useful when the system is expected to respond to a specific target speaker. If the enrolment process or pre-registration is not available, we could use a speaker diarization system to cluster the single speaker's voice together by firstly excluding the overlapped speech part, for example, in a meeting.

In addition, the speaker extraction system has no strict requirement about the duration of the reference speech. A few seconds of the reference speech is able to characterize the speaker well. For example, we use a random duration of the reference speech with a mean of 7.3s in this work and compare with other cases with fixed duration in Section 5.4.2.8.

4.3 Frequency-domain Speaker Extraction

4.3.1 A General Frequency-domain Framework

To extract the target speaker's voice from the mixture, a common approach is to perform speaker extraction in frequency-domain, and reconstruct the time-domain signal from the extracted magnitude and estimated phase spectra. A frequency-domain speaker extraction network can be generally described in Figure 4.2. The network consists of a speaker encoder and a speaker extractor. The speaker encoder simulates a top-down voluntary focus of cognitive process with the target speaker as the attention task. The speaker extractor estimates a filter (i.e., mask) that only lets pass the target speaker's voice given the speaker information encoded in the speaker encoder module.

The problem has been formulated as a supervised learning task. Specifically, the speaker extraction system estimates a filter (i.e., mask) for the target speaker with the supervised information, i.e., IBM, or clean target speech signal. The speaker encoder and speaker extractor could be jointly optimized with a signal reconstruction error, i.e., MSE between the estimated magnitude and the target clean magnitude. Then, the target speaker's magnitude spectrum $|\hat{S}(t,f)|$ is extracted by applying the estimated mask to filter the magnitude spectrum of the multi-talker speech as below,

$$|\hat{S}(t,f)| = M(t,f) \otimes |Y(t,f)| \tag{4.3}$$

where $\otimes$ indicates element-wise multiplication. $M(t,f)$ is the estimated mask and $|Y(t,f)|$ is the magnitude spectrum representation of the mixed signal.

Finally, the time-domain signal $\hat{s}(t)$ of the target speaker is reconstructed with the extracted target speaker's magnitude spectrum $|\hat{S}(t,f)|$ and the original phase spectrum $\angle Y(t,f)$ of the multi-talker speech.

$$\hat{s}(t) = OLA(iSTFT(M(t,f) \otimes |Y(t,f)|, \angle Y(t,f))) \tag{4.4}$$

where $OLA(\cdot)$ is the overlap and add algorithm. $iSTFT(\cdot)$ is the inverse short-time Fourier transform.

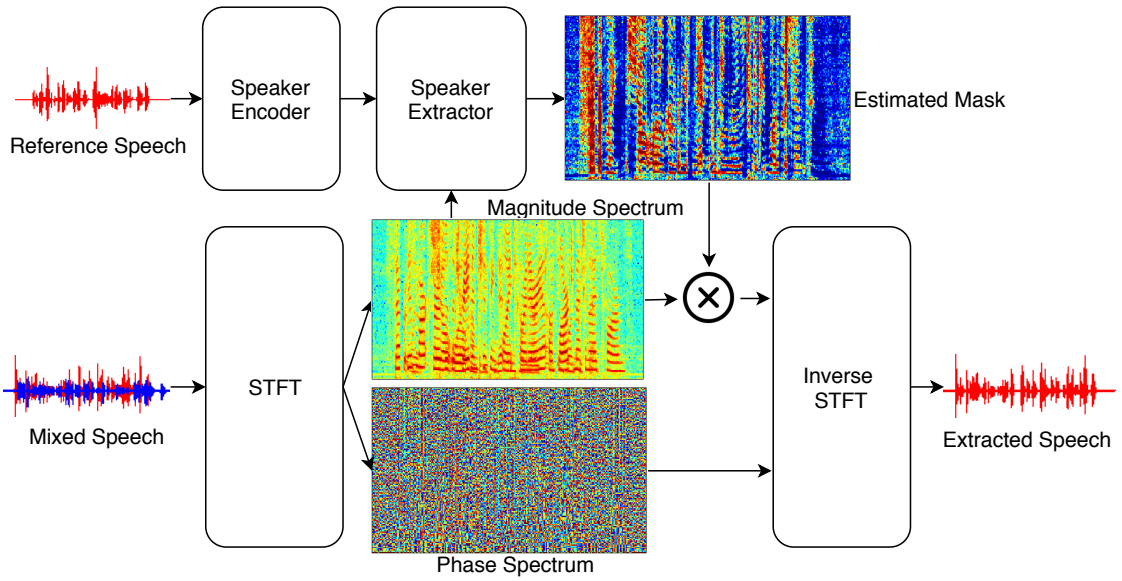


FIGURE 4.2: The block diagram of a general frequency-domain speaker extraction network, which consists of a speaker encoder and a speaker extractor.

4.3.2 Proposed Speaker Extraction with Magnitude and Temporal Spectra Approximation Loss

Inspired by the target speech recognition based on speaker beam [86], this section proposes a frequency-domain speaker extraction framework to improve the perceptual quality of a mixture speech. As discussed in Section 2.3.8, the front-end process for speaker extraction with SBF-IBM has some drawbacks, such as, the explosion of the adaptation layer with the increasing dimension of speaker embedding vector, and no contextual information in speaker encoder. To overcome these limitations, the proposed speaker extraction system applies a concatenation framework instead of the adaptation layer, a BLSTM layer in speaker encoder to capture contextual information. As we known in Chapter 3, the temporal information used in the objection function could improve the performance of speech separation for temporal continuity. This section applies a magnitude and temporal spectra approximation loss (MTSAL) to leverage the temporal information of a speech. This concatenation based speaker extraction network is named as SBF-MTSAL-Concat.

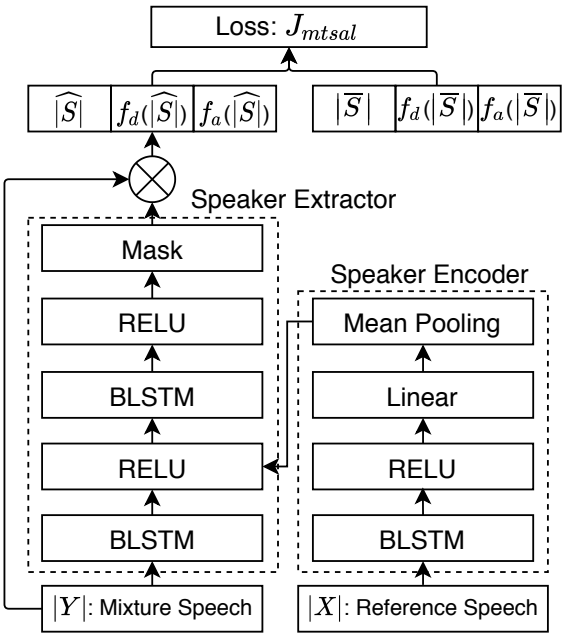


FIGURE 4.3: The framework of SBF-MTSAL-Concat for speaker extraction with magnitude and temporal spectra approximation loss. $|\overline{S}|$ is the clean magnitude of the target speaker with a phase difference, which is equal to $|S| \otimes cos(\theta_y - \theta_s)$, as defined in Eq. 4.6. $f_d(\cdot)$ and $f_a(\cdot)$ are delta and acceleration computation function.

4.3.2.1 The Concatenation Framework

Speaker information, i.e., i-vector [107], has been used for speaker adaptation of acoustic model in speech recognition [108–110]. The performance of speech recognition has been improved by simply concatenating the i-vector with the acoustic feature frame by frame repeatedly. Inspired by this idea, we propose a speaker extraction network with a simple concatenation framework instead of the adaptation layer in the SBF-IBM method. Without the adaptation layer, the computation cost caused by many sub-layers in the adaptation layer is significantly mitigated. As shown in Figure 4.3, the speaker extraction network consists of a speaker extractor that estimates a phase sensitive mask and a speaker encoder that learns a speaker embedding. The speaker embedding contains the target speaker's characteristics. To be aware of the target speaker in every frame of the representations learned from the mixture magnitude, the speaker embedding is repeatedly appended to the representations from a BLSTM frame by frame.

The SBF-IBM method encodes the speaker information by a DNN without leveraging the context of the speech. In this work, we exploit a BLSTM to model

the history and future information of the speech with its recurrent. The speaker embedding is obtained by averaging the activations of the frames over a whole utterance within the speaker encoder. The speaker encoder takes a reference speech of the target speaker as its input and outputs a D dimensional speaker embedding $V \in \mathbb{R}^{D\times 1}$.

4.3.2.2 Magnitude and Temporal Spectra Approximation Loss

To address the problems caused by the mask approximation loss (i.e., MSE between estimated and ideal mask), a magnitude spectrum approximation loss is always used to extract the target speaker in frequency-domain. The magnitude spectrum approximation loss is calculated as MSE between the extracted magnitude and the target clean magnitude.

To mitigate the inconsistency of the estimated magnitude and the mixture phase during the signal reconstruction with iSTFT, the phase sensitive mask [74] is applied. The phase difference between the mixture and the target clean speech are considered to estimate the phase sensitive mask in the objective function. The magnitude spectrum approximation loss is thus defined as,

$$\begin{aligned} J_{msal} &= \frac{1}{T}\sum || \ |\hat{S}| - |S| \otimes cos(\theta_y - \theta_s) \ ||_F^2 \\ &= \frac{1}{T}\sum || \ \hat{M} \otimes |Y| - |S| \otimes cos(\theta_y - \theta_s) \ ||_F^2 \\ &= \frac{1}{T}\sum || \ f(|Y|, |X|) \otimes |Y| - |S| \otimes cos(\theta_y - \theta_s) \ ||_F^2 \end{aligned} \tag{4.5}$$

where $|\hat{S}|$ and $|S|$ are the extracted magnitude and the target clean magnitude, respectively. θ_y and θ_s are the angles of the mixture phase and the target clean phase. $cos(\theta_y - \theta_s)$ is the phase difference between the mixture and the target clean speech. The extracted magnitude $|\hat{S}|$ is estimated by element-wise multiplying the estimated phase sensitive mask $\hat{M}$ to the mixture magnitude $|Y|$. The phase sensitive mask is further estimated by the speaker extractor network $f(\cdot, \cdot)$ given the mixture magnitude $|Y|$ and the target speaker's reference speech $|X|$.

In Chapter 3, we observe that the static and dynamic information, i.e., magnitude, delta and acceleration, contributes the temporal continuity of the separated speech. To make the extracted target speaker's voice becoming temporal continuous, we

propose to compute a magnitude and temporal spectrum approximation loss to optimize the speaker extraction network.

The loss first calculates MSE between the extracted magnitude and the target clean magnitude with phase difference. The dynamic information, i.e., delta and acceleration, of the extracted magnitude is further computed. The delta and acceleration of the target clean magnitude with phase difference are also obtained as the supervised information. Then, the MSEs between these delta and acceleration are calculated as the constraints. The loss consists of the weighted sum of the contributions of the MSEs on magnitude, delta and acceleration. The loss is formulated as,

$$\begin{aligned} J_{mtsal} = \frac{1}{T}\sum(||\hat{M} \otimes |Y| - |S| \otimes cos(\theta_y - \theta_s)||_F^2 \\ + w_d||f_d(\hat{M} \otimes |Y|) - f_d(|S| \otimes cos(\theta_y - \theta_s))||_F^2 \\ + w_a||f_a(\hat{M} \otimes |Y|) - f_a(|S| \otimes cos(\theta_y - \theta_s))||_F^2) \end{aligned} \tag{4.6}$$

where $\hat{M}$ is the estimated phase sensitive mask for the target speaker. $|Y|$ and $|S|$ are the mixture magnitude and the target clean magnitude. θ_y and θ_s are the angles of the mixture phase and the target clean phase. w_d and w_a are the weights to balance the importance of the contributions of the MSEs on magnitude, delta and acceleration. $f_d(\cdot)$ and $f_a(\cdot)$ are the delta and acceleration computation function. The delta computation function [88] is defined as,

$$f_d(v(t)) = \frac{\sum_{l=1}^{L} l \times (v(t+l) - v(t-l))}{\sum_{l=1}^{L} 2l^2} \tag{4.7}$$

where $v(t)$ is a feature vector of the magnitude at time frame t. L is the contextual window. The acceleration computation function is obtained by computing the delta twice.

4.4 Multi-scale Time-domain Speaker Extraction

In the prior work [86], a common approach is to perform speaker extraction in frequency-domain, and reconstruct the time-domain signal from the extracted magnitude spectrum and the phase spectrum of the mixture signal. Others have also studied complex ratio mask [80–82] in speech enhancement. The frequency-domain

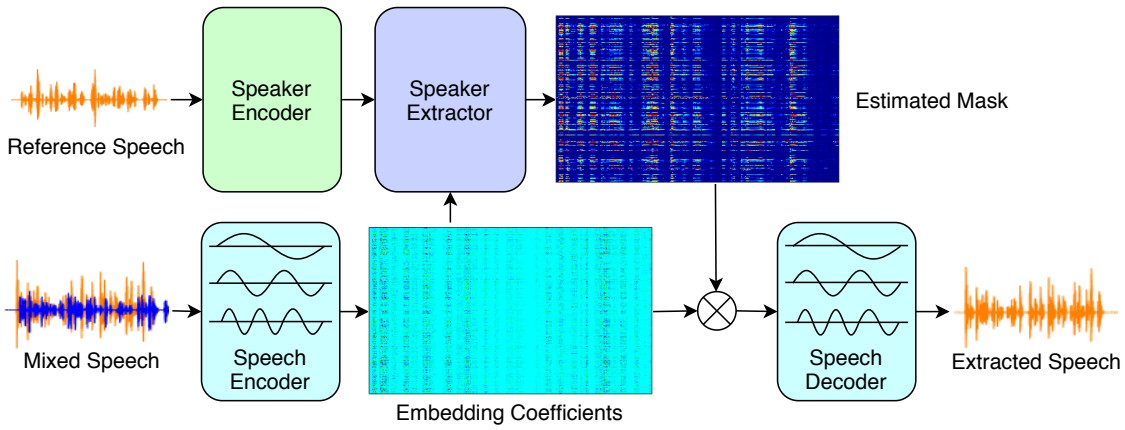


FIGURE 4.4: The block diagram of a general speaker extraction network, that consists of a speaker encode (in green), a speech encoder (in cyan), a speaker extractor (in purple), and a speech decoder (in cyan). The network components in Figure 4.4 and 4.5 share the same color codes for ease of cross reference. The speaker encoder simulates a top-down voluntary focus of cognitive process with the target speaker as the attention task.

process relies on STFT that faces the windowing effect, and phase estimation problem.

Inspired by Conv-TasNet [35, 111] for speech separation, we propose a novel end-to-end network architecture for speaker extraction (SpEx), as shown in Figure 4.4. SpEx converts the mixture speech into multi-scale embedding coefficients instead of decomposing the speech signal into magnitude and phase spectra. In this way, we avoid phase estimation.

In a frequency-domain implementation, a STFT module serves as the speech encoder that transforms the time-domain speech signal into magnitude and phase spectra, while an inverse STFT serves as the speech decoder. SpEx opts for trainable neural networks to serve as the speech encoder and the speech decoder in the time-domain speaker extraction. The speaker encoder is trained to convert the time-domain speech signal into spectrum-like embedding, also called embedding coefficients. The speech decoder reconstructs the embedding coefficients into a time-domain signal. The proposed SpEx framework is depicted in Figure 4.5 in detail.

4.4.1 SpEx Architecture

SpEx is composed of four network components: The *speaker encoder* encodes the reference speech $x(n)$ into a speaker embedding, that is the feature representation of the target speaker. The *speech encoder* encodes the time-domain mixture speech $y(n)$ into spectrum or spectrum-like feature representation. The *speaker extractor* estimates a mask that only lets pass the target speaker's voice. Finally the *speech decoder* reconstructs the time-domain speech signal from the masked spectrum of the mixture speech. From the viewpoint of selective auditory attention, the masked spectrum is called the modulated response [38]. The SpEx architecture allows the joint training of all these four modules to take place with a multi-task learning algorithm.

4.4.1.1 Speaker Encoder

In text-independent speaker recognition, it is common that we represent the speech with a fixed dimensional vector, such as i-vector [107], x-vector [112] and other similar feature representations [113], that characterize the voiceprint of a speaker. The model that converts speech samples $x(n)$ into feature representation is called speaker encoder $g(\cdot)$, and the resulting feature representation $g(x)$ is called speaker embedding.

A speaker encoder could be pre-trained independently to extract an i-vector or x-vector for the target speaker. As such speaker encoders are pre-trained independently of speaker extraction systems, they are not optimized directly for speaker extraction purposes. Another idea is to train speaker encoders jointly with the speaker extraction system [86] with the loss (i.e., mean square error) between the extracted and clean speeches. Such speaker encoders are trained to optimize the signal reconstruction for speaker extraction, however, they do not aim directly to characterize nor discriminate the speakers.

To benefit from the idea of speaker encoder and task-oriented optimization, we propose a multi-task learning algorithm to incorporate the speaker encoder as part of the SpEx network. The speaker encoder is jointly optimized by weighting a cross-entropy loss for speaker classification and a signal reconstruction loss between the extracted and clean speeches for speaker extraction. In practice, we use a BLSTM

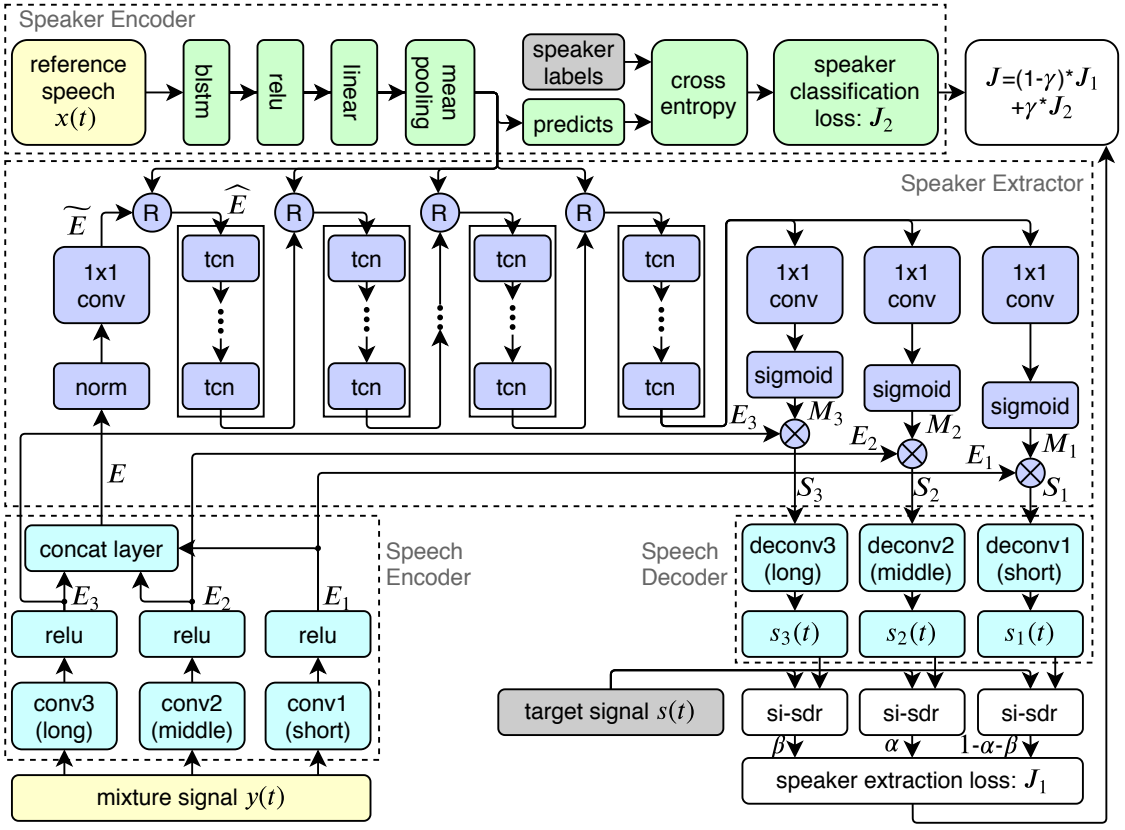


FIGURE 4.5: The block diagram of the proposed SpEx network, that consists of a speaker encoder (in green), a speech encoder (in cyan), a speaker extractor (in purple), and a speech decoder (in cyan). The network components in Figure 4.4 and 4.5 share the same color codes for ease of cross reference. Ⓡ is an operator that concatenates the speaker vector repeatedly to the intermediate representations of mixture speech along the channel dimension. Ⓧ refers to the element-wise multiplication. The "conv" and "deconv" are 1-D convolutional and de-convolutional operations. "relu" and "sigmoid" are the rectified linear unit (ReLU) and sigmoid functions. The structure of the "tcn" block is similar to Conv-TasNet as shown in Figure 4.6. The extracted signal s_1 is chosen as the ultimate output of the system at run-time inference.

to encode the context information of the reference speech into a speaker embedding with a mean pooling layer. In the multi-task learning process, the gradients from both the cross-entropy loss and the signal reconstruction loss are back-propagated to optimize the speaker encoder network. The details of the learning algorithm will be discussed in Section 4.4.2 and 4.4.3.

4.4.1.2 Speech Encoder

There have been studies on how to address the phase estimation problem for frequency-domain methods. One is to optimize the real and imaginary parts separately [80–82], another is to compensate the phase in the training process [28, 77, 79]. Such attempts have achieved limited successes due to the inexact phase estimation. Similar to Conv-TasNet [35, 111], we opt for a time-domain approach, that transforms the time-domain mixture signal directly into a feature representation using a convolutional network.

In a frequency-domain approach, by applying Fourier transform, a speech signal is decomposed into an alternate representation, characterized by sines and cosines. Similarly, in a time-domain approach, we can consider the filters in the convolutional layers as the basis functions in analogy to the sines and cosines in the frequency-domain [114]. The feature representations are considered as the embedding coefficients. After all, the time-domain encoding is different from Fourier transform in that a) the feature representations don't handle the real-and-imaginary parts separately; b) the basis functions are not pre-defined as sines or cosines, but rather trainable from the data.

The input mixture speech $y(n) \in \mathbb{R}^{1\times T}$ can be encoded to embedding coefficients using a convolutional neural network in a similar way as other end-to-end speech processing systems [35, 115, 116]. Inspired by [117, 118], this paper proposes to encode the mixture speech into multi-scale speech embeddings using several parallel 1-D CNNs with N filters each for various temporal resolutions. While the number of multiple scales can vary, we only study three different time scales in this work, without loss of generality. The filters of the parallel 1-D CNNs are of different lengths, $L_1(short), L_2(middle), L_3(long)$ samples, to cover different window sizes. The CNNs are followed by a rectified linear unit (ReLU) activation function to produce the speech embedding $E = [E_1 E_2 E_3] \in \mathbb{R}^{K\times 3N}$.

To concatenate the embeddings across different time-scale, we align them by keeping the same stride, $L_1/2$, across different scales. With the varying filter lengths, the encoder learns representations in multiple scales, for example, the short window has good resolution at high frequency and long window has high resolution at low frequency. Without trading the temporal resolution with frequency resolution like in STFT, we encode the time-domain signal into three temporal resolutions in the

embedding E. The embedding coefficients E_i in each scale consist of a sequence of vectors, $E_{i,k}$, which are defined as,

$$E_{i,k} = \text{ReLU}(y_{i,k}U_i), \quad k = 1, ..., K, i = 1, 2, 3 \tag{4.8}$$

where $K = 2(T - L_1)/L_1 + 1$, and $y_{i,k} \in \mathbb{R}^{1\times L_i}$ is the k^{th} segment of $y(t)$ that has a window of L_i samples shifting every $L_1/2$ samples. $U_i \in \mathbb{R}^{N\times L_i}$ is also called the encoder basis.

4.4.1.3 Speaker Extractor

One of the earliest theories of attention is Broadbent's filter model [119]. In psychoacoustic experiments, the stimuli are first processed according to their physical properties such as color, loudness, and pitch. The selective filters of listeners then allow for certain stimuli to pass through for further processing while other stimuli are rejected. The selective filter can be modelled by a mask that has been well studied in speech separation literature, such as ideal binary mask (IBM) [72], ideal ratio mask (IRM) [73], ideal amplitude mask (IAM) [63], wiener-filter like mask (WFM) [74] and phase sensitive mask (PSM) [74].

In the SpEx framework, the speaker embedding describes the physical properties of the auditory source, a target speaker in this case, as the focus of the attention. The speaker extractor, as shown in Figure 4.5, is conditioned on the speaker embedding both during training and at run-time inference to estimate a filter mask, that is referred to as the receptive mask. We obtain the modulated response S_i [38] for each scale $i = 1, 2, 3$ of the target speaker by applying the receptive mask M_i on the embedding coefficients E_i of the mixture signal in each scale,

$$\begin{aligned} S_i &= M_i \otimes E_i \\ &= f(E, g(x)) \otimes E_i \end{aligned} \tag{4.9}$$

where $\otimes$ is an operator for element-wise multiplication. E is the multi-scale embedding coefficients. $f(\cdot, \cdot)$ and $g(\cdot)$ are the functions representing the speaker extractor and speaker encoder. $x(n)$ is the reference speech of the target speaker to form an attention.

Specifically, the multi-scale embedding coefficients E are firstly normalized by its mean and variance on channel dimension scaled by the trainable bias and gain parameters. Then, a 1-D CNN with 1×1 kernel size, that is called 1×1 CNN, is applied. The 1×1 CNN with O filters is performed to adjust the number of channels for the inputs and residual path of the subsequent blocks of temporal convolutional network (TCN). In this way, we have the multi-scale embedding coefficients as $\widetilde{E} \in \mathbb{R}^{K \times O}$. At the same time, the speaker embedding vector $g(x) \in \mathbb{R}^{1 \times D}$ from the speaker encoder is concatenated repeatedly to $\widetilde{E}$. The multi-scale embedding coefficients with speaker information are then defined as $\widehat{E} \in \mathbb{R}^{K \times (O+D)}$. Similarly, the speaker embedding vector is also concatenated repeatedly with the representations along the subsequent TCN blocks as shown in Figure 4.5.

Similar to Conv-TasNet, we stack the TCN blocks by exponentially increasing the dilation factor to capture the long-range dependency of the speech signal. Each TCN block, as shown in Figure 4.6, applies a dilated depth-wise separable convolution to reduce the number of parameters. The dilated depth-wise separable convolution consists of a dilated depth-wise convolution ("d-conv" in Figure 4.6) and a following 1×1 CNN with O filters. Since the number of input channels of the TCN block may be different from the number of the filters of the dilated depth-wise convolution, a 1×1 CNN with P filters is applied in advance to turn the number of input channels to P. The dilated depth-wise convolution has a kernel size of $1 \times Q$, a number of P filters and a dilation factor of $2^{(B-1)}$. B is the number of TCN blocks in a stack. Such a stack is repeated for R times as shown in the speaker extractor in Figure 4.5.

To apply the mask M_i on E_i, M_i must have the same dimensions as E_i. As the output channels O from the last TCN block may be different from the channels N of the encoded representations $E_i \in \mathbb{R}^{K \times N}$, we apply one 1×1 CNN to transform the dimension of the output channels from the last TCN block to be same as the encoded representations $E_i \in \mathbb{R}^{K \times N}$. The elements of the mask $M_i \in \mathbb{R}^{K \times N}$ are estimated through a Sigmoid activation function to keep the range within $[0, 1]$. Finally, the masked embedding coefficients $S_i \in \mathbb{R}^{K \times N}$ of the target speaker, that are also called the modulated responses [38], are estimated by Eq. 4.9.

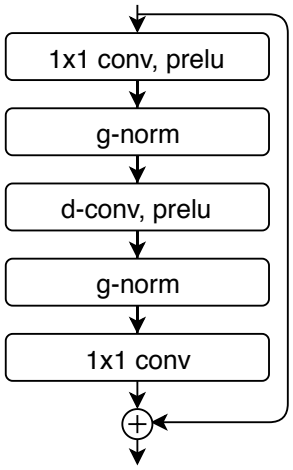


FIGURE 4.6: The structure of the "tcn" block is temporal convolutional network used in Figure 4.5. $\oplus$ denotes the residual connection. The "d-conv" is depth-wise convolution which forms a depth-wise separable convolution together with the last "1x1 conv". "prelu" is the parametric rectified linear unit (PReLU). "g-norm' is the mean and variance on both dimensions of time frames and channels scaled by the trainable bias and gain parameters.

4.4.1.4 Speech Decoder

The decoder reconstructs the time-domain speech signal from the modulated responses. Embedding coefficients at each scale lead to a modulated response. We reconstruct the multi-scale modulated response into time-domain signals (s_1, s_2, s_3) with the decoder bases $V_1 \in \mathbb{R}^{N \times L_1}$, $V_2 \in \mathbb{R}^{N \times L_2}$, and $V_3 \in \mathbb{R}^{N \times L_3}$ through a de-convolutional process. The decoder basis consists of the learned filters during training just as a Fourier basis that is composed of sine and cosine functions.

4.4.2 Multi-scale Encoding and Decoding

Speech has a rich temporal structure over multiple time scales presenting phonemic, prosodic and linguistic content [118]. It was shown that speech analysis of multiple temporal resolutions leads to improved speech recognition performance[120]. As shown in Figure 4.5, we implement multi-scale encoding in speech encoder and speaker extractor. The speaker encoder first encodes the mixture signal into a multi-scale embedding coefficients $E = [E_1 E_2 E_3]$. The speaker extractor then estimates multi-scale masks M_1, M_2, M_3, and generates the multi-scale modulated

responses S_1, S_2, S_3. We finally reconstruct the multi-scale modulated responses into time-domain signals s_1, s_2, s_3 at multiple scales with the speaker decoder.

During training, we calculate a multi-scale scale-invariant signal-to-distortion ratio (SI-SDR) loss, defined as J_1, that aims to minimize the signal reconstruction error,

$$J_1 = -[(1-\alpha-\beta)\rho(s_1, s) + \alpha\rho(s_2, s) + \beta\rho(s_3, s)] \tag{4.10}$$

where α and β are the weights. s_1, s_2 and s_3 are the signals reconstructed from the modulated responses S_1, S_2 and S_3, respectively. s is the clean speech signal serving as the training target. We use the SI-SDR loss [85], denoted as $\rho(\cdot, \cdot)$, as the measure of reconstruction error.

$$\rho(\hat{s}, s) = 10\log_{10}\left(\frac{||\frac{\langle \hat{s}, s\rangle}{\langle s, s\rangle} s||^2}{||\frac{\langle \hat{s}, s\rangle}{\langle s, s\rangle} s - \hat{s}||^2}\right) \tag{4.11}$$

where $\hat{s}$ and s are the extracted and target signals of the target speaker, respectively. $\langle \cdot, \cdot \rangle$ is the inner product. To ensure scale invariance, the signals $\hat{s}$ and s are normalized to zero-mean prior to the SI-SDR calculation.

The calculation of multi-scale SI-SDR J_1 loss is required only during training and not at run-time inference. At run-time inference, we evaluate the quality of the signals reconstructed at multiple scales individually, i.e. s_1, s_2, s_3, and collectively as a weighted summation s_w=(1-α-β)s_1+αs_2+βs_3.

4.4.3 Multi-task Learning

We propose to train the speaker encoder together with three other network components as a whole. The speech encoder, speaker extractor, and speech decoder are optimized to minimize the multi-scale SI-SDR loss, while the speaker encoder is optimized with two objective functions, the multi-scale SI-SDR loss and the cross-entropy loss for speaker classification.

The cross-entropy loss J_2 for speaker classification is defined as,

$$J_2 = -\sum_{i=1}^{N_s} p_i \log(\hat{p}_i) \tag{4.12}$$

where N_s is the number of speakers in the speaker classification task. p_i is the true class label for speaker i and $\hat{p}_i$ is the predicted speaker probability.

We combine J_1 with J_2 to optimize the speaker encoder network in a multi-task learning, as J_1 and J_2 represent two different optimization tasks. With the multi-task learning, the speaker encoder network is trained not only to characterize the unique properties of the target speaker, but also to contribute to the overall speaker extraction objective. The total loss J is a weighted sum of J_1 and J_2,

$$J = (1 - \gamma)J_1 + \gamma J_2 \tag{4.13}$$

4.4.4 Relationship with TasNet

SpEx network can be seen as an extension to Conv-TasNet [35, 111] from speech separation to speaker extraction applications. A comparison with TasNet framework will help the understanding of SpEx.

BLSTM-TasNet [33, 34] and Conv-TasNet [35, 111] represent a successful attempt to address the phase estimation problem in frequency-domain speech separation. The techniques employ an encoder-separator-decoder pipeline, and learn trainable basis functions with a 1-D convolution and de-convolution instead of Fourier series consisting of sine and cosine functions. Speech separation is performed by estimating a mask for each speaker in the mixture using either a BLSTM in BLSTM-TasNet or a fully convolutional neural network (CNN) in Conv-TasNet. Conv-TasNet uses a TCN architecture together with a dilated separable depthwise convolution that represents an effective time-domain implementation.

The idea of SpEx is similar to that of Conv-TasNet in the sense that the speaker extractor of SpEx is based on the same TCN architecture[111], and the encoder-extractor-decoder pipeline is inspired by the encoder-separator-decoder pipeline of Conv-TasNet. However, SpEx is also different from Conv-TasNet in the following ways:

- **Top-down voluntary focus:** SpEx features a speaker encoder as the top-down voluntary focus in selective auditory attention. It learns to single out one voice from the multi-talker mixture by modulating the input stimulus with a top-down attention. However, Conv-TasNet doesn't employ such a

mechanism. It learns to segregate two competing voices by estimating two filtering masks. Just like other speaker extraction techniques, SpEx addresses the issues of global speaker permutation ambiguity and unknown number of speakers that we face in speech separation.

- **Multi-task learning:** As Conv-TasNet doesn't involve a speaker encoder, it is trained only to optimize the reconstruction errors, equivalent to the SI-SDR loss in this paper. SpEx adopts a multi-task learning algorithm to jointly optimize all network components, with a cross-entropy loss for speaker classification and a SI-SDR loss for signal reconstruction. The speaker encoder is optimized by the total loss defined in Eq. 4.13. Such a total loss is different from the prior works, where the speaker extraction systems train the speaker encoder with either speaker classification loss [121], or signal reconstruction loss [86, 122, 123] as a single task.

- **Multi-scale encoding and decoding:** The TCN architecture in Conv-TasNet works well for single time scale embedding coefficients [35, 111]. Multi-scale encoding is effective in deep neural networks approach to speech recognition[118]. We believe that, if the TCN architecture is trained with multi-scale embedding coefficients, it learns to reconstruct the rich temporal structure of speech in greater detail. This will be an interesting study of the TCN architecture.

4.5 Summary

In this chapter, we propose a frequency-domain and a time-domain speaker extraction systems that emulate humans' ability of selective auditory attention. The system performs speaker extraction without the need of the information about the number of speakers. It forms a top-down voluntary focus by using the reference speech of the target speaker. Although the target speaker information is required, the proposed technique is particularly useful for the system to respond to the registered speakers, for example, in speaker verification where the target speaker is known to the system through enrollment.

Chapter 5

Evaluation and Analysis of Speaker Extraction

In Chapter 4, the proposed frequency-domain and time-domain speaker extraction systems have been introduced. The models attempt to address the problems of both unknown number of speakers in real-world environment and global speaker permutation ambiguity across utterances in current speech separation techniques. This chapter presents the experimental setups, evaluation results and discussions on the proposed models.

Section 5.1 firstly introduces the speaker extraction database and the speech separation database. The speech separation database is extended to be used in the evaluation of speaker extraction for directly comparisons with other speech separation methods. The experimental setups of the proposed frequency-domain and time-domain speaker extraction systems are described in Section 5.2 and Section 5.3, respectively. Section 5.4 and Section 5.5 report the performances of the proposed speaker extraction methods under two-speaker mixture and three-mixture condition, respectively. Furthermore, The comparisons between proposed speaker extraction methods and other speech separation methods are conducted on the WSJ0-2mix database under clean two-speaker mixture condition in Section 5.6. Finally, Section 5.7 and Section 5.8 conduct the comparisons on the WHAM! and

The work in this chapter has been published in [41–43].

WHAMR! databases for noisy, reverberant, and noisy and reverberant two-speaker mixture conditions, respectively. The conclusions are drawn in Section 5.9.

5.1 Database

Unlike in speech separation, speaker extraction technique requires a reference speaker's voice to supervise the voluntary attention. We firstly re-organize the well-known WSJ0-2mix and WSJ0-3mix [26] with "max" data structure by selecting the first chosen speaker as the target speaker, while keeping the mixing utterances and SNR values the same. We rename the simulated databases as WSJ0-2mix-extr and WSJ0-3mix-extr in this work to differentiate from the original WSJ0-2mix and WSJ0-3mix database, as described in Section 5.1.1.

To compare with speech separation techniques, we further extend the well-known WSJ0-2mix, WHAM! and WHAMR! databases for speaker extraction by providing a reference speech for each speaker in the mixture, as discussed in Section 5.1.2.

5.1.1 WSJ0-2mix-extr and WSJ0-3mix-extr

We simulated a two-speaker (WSJ0-2mix-extr) and a three-speaker (WSJ0-3mix-extr) mixture databases[1] according to the well-known WSJ0-2mix and WSJ0-3mix [26]. The speech signals are sampled at a sampling rate of 8kHz based on the WSJ0 corpus [45]. Each database has three datasets: training set ($20,000$ utterances), development set ($5,000$ utterances), and test set ($3,000$ utterances).

Same as [26], the training set and development set are generated by randomly selecting two utterances for two-speaker database, and three utterances for three-speaker database, from 50 male and 51 female speakers in the WSJ0 "si_tr_s" set at various signal-to-noise ratio (SNR) between 0dB and 5dB. The training set is used for the training of network components, while the development set is for optimizing system configurations.

[1]The code is available at: `https://github.com/xuchenglin28/speaker_extraction`.

Similarly, the utterances from 10 male and 8 female speakers in the WSJ0 "si_dt_05" set and "si_et_05" set are randomly selected to create the test set. Since the speakers in the test set are excluded from the training and development sets, and the reference speech is not used in any of the speech mixing, the test set is developed for speaker independent evaluation, also called open condition evaluation.

To include both overlapping and non-overlapping speech in the dataset, we keep the maximum length of the mixing utterances as the length of the mixture. The speaker of the first randomly selected utterance is regarded as the target speaker. At run-time, the speaker extraction process is conditioned on a reference speech from the target speaker.

As the reference speech is randomly selected, the duration of the reference speech varies in training, development and test sets. We call this experimental condition as "Random". In the test set, the average duration of the reference speech is 7.3s with a standard deviation of 2.7s, a maximum length of 19.6s, and a minimum length of 1.6s. The experiments are conducted under this "Random" condition if not stated otherwise.

In two-speaker database, we also group the reference speech for the test set into four duration groups, i.e. 7.5s, 15s, 30s and 60s, for the experiment on duration of reference speech, as reported in Section 5.4.2.8.

While most of the comparative experiments are conducted on the two-speaker database, we also extend the experiments beyond two-speaker mixture. A three-speaker database is constructed in a similar way as the two-speaker database, except that the duration of the reference speech in the test set is kept as 15s and 60s. In the experiment for three-speaker mixture, we train the speaker extraction network under three conditions, two-speaker mixture only, three-speaker mixture only, and two-speaker and three-speaker mixture in combination. The trained speaker extraction systems are then evaluated on the two-speaker and three-speaker mixture test set, respectively.

5.1.2 WSJ0-2mix, WHAM! and WHAMR!

As discussed in Section 2.1.2, WSJ0-2mix, WHAM! and WHAMR! were well-known databases for speech separation. To compare with the state-of-the-art speech separation techniques, we conducted speaker extraction experiments on these well-known databases by providing a reference signal of each speaker in the mixture to form the auditory attention. Then the voices of each speaker in the mixture could be obtained, like speech separation as discussed in Section 4.2. Since the mixture signal and clean target signals were kept same, the performances between the proposed speaker extraction methods and the state-of-the-arts speech separation techniques were comparable.

Different from WSJ0-2mix-extr database, WSJ0-2mix, WHAM! and WHAMR! were generated with "min" scheme. The duration of the mixture signal was kept same as the minimum duration of the two speaker's voices. The speaker's voice with long duration was trimmed to be equal to the short one. In addition, WHAM! and WHAMR! extended the WSJ0-2mix by adding noise and reverberation. WHAM! was used to study the performance of speaker extraction under two-speaker mixture with noise condition. The reverberate-only, and reverberate and noisy two-speaker mixture conditions were studied with WHAMR! database.

5.2 Frequency-domain Speaker Extraction Setup

The speech signal was transformed into frequency-domain via a STFT with a window length of 32ms and a shift of 16ms. A normalized square root hamming window was applied. The magnitude features were thus obtained with a dimension of 129 for both the mixture speech and the reference speech.

In the SBF-IBM baseline, as shown in Figure 2.7, the speaker encoder consisted of 2 non-linear ReLU layers, a linear layer and a mean pooling layer. The ReLU layer had 512 hidden nodes. The linear layer outputed the adaptation weights associated with the target speaker, which were used to weight the contributions of the sub-layers in the adaptation layer. Since the number of sub-layers were equal to the dimension of the adaptation weights, the number of hidden nodes in the linear layer was set to 30 to have a manageable computation cost. The

adaptation weights were obtained by averaging the 30 dimensional outputs over the frames of an utterance. In the speaker extractor, a BLSTM first transformed the mixture magnitude into representations by using the contextual information of the inputs in forward and backward directions. The BLSTM had 512 cells in forward and backward directions, respectively. Then, an adaptation layer was followed with 30 sub-layers. Each sub-layer had 512 hidden nodes with 1024 dimensional inputs from the previous BLSTM. The 30 dimensional weights associated with the target speaker were used to weight sum the contributions of these sub-layers. Two additional non-linear ReLU layers were used. The output layer was followed with 129 hidden nodes to predict the mask for the target speaker with the IBM as the supervision. The IBM was obtained with a threshold that was 40dB below the maximum energy of that utterance by following the work in [26]. If the energy of the time-frequency bin was greater than the threshold, the IBM was set to 1. Otherwise, the IBM was 0. The mask approximation loss defined in Eq. 2.45 was used to optimize the network.

To evaluate the performance of direct signal approximation loss, we further replace the mask approximation loss by a magnitude spectrum approximation loss, and a magnitude and temporal spectrum approximation loss, as defined in Eq. 4.5 and 4.6, respectively. The network configures were kept same as the SBF-IBM baseline. We called them SBF-MSAL and SBF-MTSAL, individually. Different from the SBF-IBM baseline to estimate binary mask, the phase sensitive mask was predicted.

In the SBF-MTSAL-Concat approach, the magnitude of the reference speech was feeded into a BLSTM to capture the contextual information in the speaker encoder. The BLSTM had 256 cells in both forward and backward directions, respectively. A non-linear ReLU layer with 256 hidden nodes was further added. Then, a linear layer with 30 nodes was used to obtain a 30 dimensional speaker embedding with a mean pooling over the frames of an utterance. Although the increasing of the dimension of the speaker embedding might not result in the parameter explosion problem, we kept the dimension of the speaker embedding same as the dimension of the adaptation weights in the SBF-IBM baseline [86]. In the speaker extractor, a BLSTM with 512 cells in both forward and backward directions was built on top of the magnitude of the mixture. The speaker embedding was repeatedly concatenated with the representations from the previous BLSTM frame by frame.

With the concatenation at each frame, the network was able to know the speaker information when it learned to extract the target speaker from the representations. Then, a non-linear ReLU layer was used to transform the concatenated representations with the target speaker information. A BLSTM and an additional non-linear ReLU layer were used. The number of cells and hidden nodes were set to 512. The mask estimation layer had a dimension of 129. The estimated phase sensitive masks were element-wisely multiplied with the magnitude of the mixture to extract the target speaker. The network was optimized with a loss defined in Eq. 4.6.

All experiments were done with a learning rate started from 0.0005 and scaled down by 0.7 when the training loss increased on the development set. The minibatch size was 16. The utterances of the training set and the development set were sorted ascendingly by their duration during training. The network was forced to be trained with minimum 30 epochs and stopped when the relative loss reduction was lower than 0.01. The network was optimized with the Adam algorithm [96] and implemented with Tensorflow [2].

5.3 Multi-scale Time-domain Speaker Extraction Setup

The network is optimized by the Adam algorithm [96]. The learning rate begins with 0.001 and halves when the loss increases on the development set for at least 3 epochs. Early stopping scheme is applied when the loss increased on the development set for 10 epochs. The minibatch size is set to fully occupy the GPU memory, i.e., 10 with a 32GB V100 GPU for a SpEx system. The utterances in the training and development set are broken into 4s segments[3].

[2] `https://www.tensorflow.org/`

[3] We discard the segments less than 4s or containing only silence for the target speech for the experiments conducted on WSJ0-2mix-extr and WSJ0-3mix-extr with Tensorflow. To fairly compare with speech separation techniques on WSJ0-2mix, WHAM! and WHAMR!, we keep the segments less than 4s by padding with zeros. Since these databases are simulated with "min" data structure, there are no segments contained only silence. We implement the SpEx system with Pytorch (`https://pytorch.org/`).

5.3.1 Speaker Encoder

The speaker encoder in Figure 4.4 translates the reference speech of the target speech into a top-down voluntary focus that the speaker extractor network can act upon. In Figure 4.5, we propose a detailed implementation, that is to repeatedly concatenate the speaker embedding vector with the inputs to TCN blocks. In this thesis, we advocate the idea to incorporate the speaker encoder network as an integral part of the SpEx architecture during training and at run-time inference. As a contrastive experiment, we would like to know how such speaker encoder performs differently from a traditional i-vector extractor [107]. We choose i-vector because it has been one of the most effective techniques for text-independent speaker characterization.

5.3.1.1 I-vector Extractor

An i-vector extractor converts a speech sample into a low-dimension vector. We first train the UBM and total variability matrix with the single speaker (clean) speech from the training and development sets. The acoustic features include 19 MFCCs together with energy, and their 1st- and 2nd-derivatives, followed by cepstral mean normalization [124] with a window size of 3 seconds. The 60-dimensional acoustic features are extracted from a window length of 25ms with a shift of 10ms. A Hamming window is applied. An energy based voice activity detection method is used to remove the silence frames. The i-vector extractor is based on a gender-independent UBM with 512 mixtures and a total variability matrix with 400 total factors.

5.3.1.2 Speaker Encoder

We use the same acoustic features as in the training of i-vector extractor. To leverage the temporal information of the whole reference speech, a BLSTM with 256 cells in each forward and backward direction is used to capture the speaker information from the acoustic features, which is same as the configuration in proposed frequency-domain speaker extraction approach. A non-linear layer with ReLU activation function with 256 nodes is followed by the BLSTM. Another linear layer with

400 nodes followed by a mean pooling is applied to extract the speaker embedding vector, that has the same dimension as the i-vector for fair comparison.

5.3.2 Speaker Extraction Pipeline

The speaker extraction pipeline includes speech encoder, speaker extractor, and speech decoder. The parameters that are quoted in this section have been tuned empirically for the best performance on the development set.

5.3.2.1 Speech Encoder

In the SpEx implementation detailed in Figure 4.5, the speech encoder encodes the mixture speech input $Y \in \mathbb{R}^{1\times T}$ into embedding coefficients by three parallel 1-D convolution of $N(= 256)$ filters each, followed by a ReLU activation function. To learn multi-scale embeddings with different time resolutions, the three 1-D convolutions had filter lengths of $L_1(short), L_2(middle), L_3(long)$ with a stride of $L_1/2$ samples. L_1, L_2, L_3 windows are tuned to cover $20(2.5ms), 80(10ms), 160(20ms)$ samples in this work.

5.3.2.2 Speaker Extractor

As shown in Figure 4.5, a mean and variance normalization with trainable gain and bias parameters is applied to the embedding coefficients $E \in \mathbb{R}^{K\times 3N}$ on the channel dimension, where K is equal to $2(T - L_1)/L_1 + 1$. A 1x1 convolution linearly transformed the normalized embedding coefficients E to the representations $\widetilde{E} \in \mathbb{R}^{K\times O}$ with $O(= 256)$ channels, which determined the number of channels in the input and residual path of the subsequent 1×1 CNN. Same as Conv-TasNet [111], the number of input channels P and the kernel size $1 \times Q$ of each depthwise convolution are set to 512 and 1×3. $B(= 8)$ TCN blocks are formed as a stack and repeated for $R(= 4)$ times.

TABLE 5.1: SDR (dB), SI-SDR(dB) and PESQ in a comparative study of the frequency-domain speaker extraction systems under open condition. "Mixture" refers to original input mixture with zero effort. "SBF-MSAL" replaces the mask approximation loss in the SBF-IBM baseline with a magnitude spectrum approximation loss, as definded in Eq. 4.5. "SBF-MTSAL" further adds the temporal constraint to form a magnitude and temporal spectrum approximation loss, as defined in Eq. 4.6. "SBF-MTSAL-Concat" is the novel concatenation framework with magnitude and temporal spectrum approximation loss instead of the adaptation structure. "#Paras" means the number of parameters of the model.

Methods	#Paras	SDR	SI-SDR	PESQ
Mixture	-	2.6	2.5	2.31
SBF-IBM [86]	19.3M	6.5	6.3	2.32
SBF-MSAL	19.3M	9.6	9.2	2.64
SBF-MTSAL	19.3M	9.9	9.5	2.66
SBF-MTSAL-Concat	8.9M	11.0	10.6	2.73

5.3.2.3 Speech Decoder

The speech decoder in Figure 4.5 reconstructs the time-domain speech signal (s_1, s_2, s_3) from the modulated responses (S_1, S_2, S_3) through a de-convolution process. The filter in the de-convolution has the same configuration as that in the speech encoder, where the number of filters (N) is equal to 256 and the filter lengths (L_1, L_2, L_3) are tuned to be $20(2.5ms), 80(10ms), 160(20ms)$ samples.

5.4 Evaluation of Speaker Extraction on WSJ0-2mix-extr Database

5.4.1 Frequency-domain Speaker Extraction Result

5.4.1.1 Overall Comparisons

The SDR, SI-SDR and PESQ performances of frequency-domain speaker extraction systems are summarized in Table 5.1. Compared with the original mixture, the PESQ of the SBF-IBM baseline shows a slight improvement. The main reason is that the mask approximation loss is not direct signal reconstruction error. The smaller error between the estimated masks and IBMs doesn't lead to the smaller error of the signal itself. The smaller error of mask estimation doesn't mean better

speech quality and intelligibility. The estimated mask may harm the speech context by forcing it to be close to IBM with binary values using mask approximation loss.

When applying the magnitude spectrum approximation loss, the performances of the SDR, SI-SDR and PESQ are significantly improved, as shown in Table 5.1. Compared with the SBF-IBM baseline, the SDR, SI-SDR and PESQ of the SBF-MSAL are relatively improved by 47.7%, 46.0% and 13.8% with a significant value p (< 0.01), respectively. It shows that the magnitude spectrum approximation loss is directly minimizing the error on the signal itself. The better speech quality and intelligibility are thus achieved. The SBF-MTSAL method further improves the performance in terms of SDR, SI-SDR and PESQ significantly ($p < 0.05$). By adding the temporal constraints of dynamic information in the objective function, the system improves the temporal continuity of the extracted speech.

Different from the speaker encoder using a DNN in the SBF-IBM, SBF-MSAL, and SBF-MTSAL approaches, the SBF-MTSAL-Concat approach applies a BLSTM to leverage the context of the reference speech. With the contextual information, the speaker embedding is well learned to capture the target speaker's characteristics. We observe that the SBF-MTSAL-Concat framework further improves the quality and intelligibility of the extracted speech. Compared with the SBF-MTSAL system, the SDR is significantly improved from 9.9dB to 11.0dB with a 11.1% relative improvement under the open condition ($p < 0.01$). Meanwhile, the SI-SDR and PESQ have been improved relatively by 11.6% and 2.6%, individually. Compared with the SBF-IBM baseline, the SBF-MTSAL-Concat method achieves a 69.2%, 68.3% and a 17.7% relative SDR, SI-SDR and PESQ improvements under open condition. The improvements are statistically significant ($p < 0.01$). The number of parameters in the network is also significantly reduced.

5.4.1.2 Same Gender vs. Different Gender

The performances of the same gender and different gender mixture are further analyzed and summarized in Table 5.2. We observe that the performance of the different gender mixture is always better than that of the same gender. The main reason is that the speaker characteristic of the same gender is less discriminating than that of the different gender. Although the speaker extraction of the same gender mixture remains challenging, the proposed techniques still improve the speech

TABLE 5.2: SDR (dB) and PESQ in a comparative study of the same gender and different gender mixture under open condition.

Method	SDR		PESQ	
	Diff.	Same	Diff.	Same
Mixture	2.5	2.7	2.29	2.34
SBF-IBM [86]	7.6	5.1	2.42	2.19
SBF-MSAL	12.0	6.9	2.82	2.43
SBF-MTSAL	12.3	7.2	2.85	2.44
SBF-MTSAL-Concat	**12.9**	**8.8**	**2.90**	**2.54**

quality and intelligibility of the same gender mixture. From Table 5.2, we observe that the SBF-MTSAL-Concat could achieve 69.7% and 72.5% relative improvements over the SBF-IBM baseline in terms of SDR for the different and same gender mixture conditions, respectively.

5.4.2 Multi-scale Time-domain Speaker Extraction Result

5.4.2.1 Frequency-domain vs. Time-domain

In this experiment, we would like to compare between two processing paradigms, the frequency-domain and the proposed time-domain methods. For frequency-domain implementation, we adopt STFT and inverse STFT as the speech encoder and decoder in Figure 4.4, respectively. For time-domain implementation, we adopt the speech encoder and decoder proposed in Section 4.4.1. In both systems, we adopt i-vector extractor as the speaker encoder for fair comparisons. As the i-vector extractor is trained independently from the speaker extraction pipeline, this comparison is focused on frequency-domain and time-domain speaker extraction pipeline. As the frequency-domain method uses a fixed short-time window of 256 samples, the time-domain systems are also implemented with a single short-time window, or single scale as opposed to multi-scale as discussed in Section 4.4.1, for fair comparison.

We observe from Table 5.3, that the time-domain speaker extraction systems (System 2-13) consistently outperform the frequency-domain counterpart (System 1), especially when time-domain systems have fewer than or roughly the same number of parameters as the frequency-domain system.

TABLE 5.3: SDR (dB), SI-SDR(dB) and PESQ in a comparative study between frequency-domain and time-domain under open condition. L_1 is the filter length of the convolution in the speech encoder for single scale in this experiment. N, O, P, Q, B, R are the parameters of the extractor defined in Section 4.4.1.3. In the frequency-domain implementation, we use the phase spectrum from the original mixture speech to reconstruct the speech signal. "#Paras" indicates the total number of parameters in the network. i-vector is used as feature representation of reference speaker.

System	Domain	N	L_1	O	P	Q	B	R	#Paras	SDR	SI-SDR	PESQ
1	Frequency	-	256	256	512	3	8	4	9.0M	10.3	9.9	2.85
2	Time	128	20	128	128	3	8	4	1.3M	12.3	11.7	2.85
3		128	20	128	128	3	8	5	1.7M	12.0	11.2	2.82
4		512	20	128	256	3	8	4	2.6M	12.4	11.6	2.83
5		512	20	128	512	3	8	3	3.7M	11.7	10.9	2.78
6		256	20	256	256	3	8	4	4.7M	12.6	11.9	2.88
7		512	20	128	512	3	8	4	4.9M	12.9	12.1	2.89
8		256	20	256	256	3	9	4	5.2M	12.8	12.2	2.89
9		256	20	256	512	3	8	4	9.0M	**13.1**	**12.4**	2.92
10		256	40	256	512	3	8	4	9.1M	12.7	11.9	2.90
11		256	80	256	512	3	8	4	9.1M	13.0	12.4	2.93
12		256	160	256	512	3	8	4	9.1M	12.2	11.5	2.88
13		256	256	256	512	3	8	4	9.2M	12.8	12.2	**2.94**

The results clearly show the advantage of the trainable speech encoder and decoder over the static STFT and inverse STFT in the frequency-domain. We consider that the better performance is attributed to the use of embedding coefficients in place of magnitude and phase spectra in the process, that avoids the need of phase estimation.

5.4.2.2 Single-scale vs. Multi-scale

In this experiment, we would like to validate the idea of multi-scale speech embedding. We continue to use i-vector extractor as the speaker encoder. From the experiments reported in Table 5.3, we observe that systems of more parameters perform better. By varying the filter length of the convolution layer in the speech encoder from System 9-13, we observe that the change of time-frequency resolution of the embedding coefficients has an impact on the system performance. The best SDR is achieved as 13.1dB with a filter length of 20 samples ($2.5ms$). The best SI-SDR is 12.4dB with the filter length of 20 samples ($2.5ms$) and 80 samples ($10ms$). The best PESQ is 2.94 with a filter length of 256 samples ($32ms$). This finding is similar to that in speech recognition experiment [118] .

To benefit from the different time-frequency resolutions, we propose to have three 1-D CNNs with different filter length, short, middle, and long, in the speech encoder.

TABLE 5.4: SDR (dB), SI-SDR (dB) and PESQ in a comparative study between single-scale and multi-scale under open condition. L_1, L_2 and L_3 are the various filter lengths of convolutions in the speech encoder. N (256), O (256), P (512), Q (3), B (8), R (4) are the parameters of the extractor defined in Section 4.4.1.3. α and β are the weights defined in the multi-scale SI-SDR loss J_1 in Eq. 4.10. "#Paras" indicates the total number of parameters in the network. s_w=(1-α-β)s_1+αs_2+βs_3 denotes the weighted summation of the reconstructed signal. The number of parameters during evaluation is less than that of training when only picking s_1 as the reconstructed signal. i-vector is used as feature representation of reference speaker.

System	L_1	L_2	L_3	α	β	Single vs Multiple Scale		Loss Function	Reconstructed Signal	#Paras	SDR	SI-SDR	PESQ
						Speech Encoder	Speech Decoder						
9	20	-	-	-	-	single	single	$\rho(s_1, s)$	s_1	9.0M	13.1	12.4	2.92
14	20	80	160	-	-	multiple	single	$\rho(s_1, s)$	s_1	9.2M	13.6	13.0	3.00
15	20	80	160	0.05	0.05	multiple	multiple	J_1	s_1	9.4M	12.6	11.9	2.84
16	20	80	160	0.10	0.10	multiple	multiple	J_1	s_1	9.4M	**13.9**	**13.3**	3.00
17	20	80	160	0.20	0.20	multiple	multiple	J_1	s_1	9.4M	13.2	12.6	2.94
18	20	80	160	0.33	0.33	multiple	multiple	J_1	s_1	9.4M	12.5	11.8	2.86
19	20	80	160	0.10	0.05	multiple	multiple	J_1	s_1	9.4M	12.4	11.4	2.84
20	20	80	160	0.20	0.10	multiple	multiple	J_1	s_1	9.4M	13.1	12.4	2.93
21	20	80	160	0.30	0.20	multiple	multiple	J_1	s_1	9.4M	13.0	12.4	2.89
22	20	80	160	0.10	0.10	multiple	multiple	J_1	s_2	9.4M	12.2	11.4	**3.01**
23	20	80	160	0.10	0.10	multiple	multiple	J_1	s_3	9.4M	12.1	11.4	3.00
24	20	80	160	0.10	0.10	multiple	multiple	J_1	s_w	9.4M	13.9	13.3	3.00

The speaker extractor and speech decoder are also extended to be compatible for the multi-scale speech embedding, as shown in Figure 4.5. The speaker extractor estimates the mask for the target speaker at each scale. The speech decoder reconstructs the time-domain signal for each scale with the modulated response.

We explore different system configurations that are summarized in System 14-24 of Table 5.4. Comparison between System 9 and System 14 shows that the multi-scale speech encoder achieves better performance than single-scale speech encoder. If the speech decoder has multiple outputs with the multi-scale speech embeddings, we could optimize the SpEx network with a weighted multi-scale SI-SDR loss, as defined in Eq. 4.10. With multi-scale speech encoder and decoder, the best performances of the SDR, SI-SDR and PESQ are achieved at 13.9dB, 13.3dB and 3.00 when the weights α and β in Eq. 4.10 are tuned to be 0.10 and 0.10. Comparing with the single-scale system, the performance of the multi-scale SpEx improves the SDR of 6.1%, the SI-SDR of 7.3%, and the PESQ of 2.7%. Comparisons between System 16 and System 22-24 present that the best performance is achieved by picking the output stream s_1 with short window (high temporal resolution). By only picking the reconstructed signal s_1 instead of a weighted summation (s_w=(1-α-β)s_1+αs_2+βs_3), the number of parameters during evaluation is less than that during training.

5.4.2.3 I-vector vs. Speaker Embedding

We have observed that the i-vector is effective in speaker characterization for both single-scale and multi-scale speaker extraction networks as reported in Tables 5.3 and 5.4. We note that the i-vector is extracted independently of the speaker extraction network. In this experiment, we would like to replace the i-vector extractor with the speaker encoder. The speaker encoder is trained jointly with other components of the network using both the cross-entropy loss for speaker classification and the multi-scale SI-SDR loss for speaker extraction as System 25 to 31 in Table 5.5.

We obtain the best SDR and SI-SDR of 15.1dB and 14.6dB when the weight for the sub-loss of the cross-entropy is tuned to be 0.2. Comparing with the i-vector based system (System 16 in Table 5.4), we observe that the joint optimization of the speaker encoder and the speaker extraction pipeline (System 27 in Table 5.5)

TABLE 5.5: SDR (dB), SI-SDR(dB) and PESQ in a comparative study between i-vector and speaker embedding as feature representations of reference speaker under open condition. L_1 (20), L_2 (80) and L_3 (160) are the various filter lengths of convolutions in the speech encoder. N (256), O (256), P (512), Q (3), B (8), R (4) are the parameters of the extractor defined in Section 4.4.1.3. α and β are the weights defined in the multi-scale SI-SDR loss J_1 in Eq. 4.10. γ is the weight of multi-task learning defined in Eq. 4.13. "MTL" indicates whether the multi-task learning is applied. "#Paras" indicates the total number of parameters in the network. s_w=(1-α-β)s_1+αs_2+βs_3 denotes the weighted summation of the reconstructed signal. The number of parameters during evaluation is less than training when only picking s_1 as the output.

System	α	β	γ	Speaker Characterization	Speaker Encoder Joint Optimization	MTL	Loss Function	Reconstructed Signal	#Paras	SDR	SI-SDR	PESQ
16	0.1	0.1	-	i-vector	no	no	J_1	s_1	9.4M	13.9	13.3	3.00
25	0.1	0.1	-	speaker embedding	yes	no	J_1	s_1	10.8M	14.2	13.7	3.04
26	0.1	0.1	0.1	speaker embedding	yes	yes	J	s_1	10.8M	15.0	14.6	**3.15**
27	0.1	0.1	0.2	speaker embedding	yes	yes	J	s_1	10.8M	**15.1**	**14.6**	3.14
28	0.1	0.1	0.3	speaker embedding	yes	yes	J	s_1	10.8M	14.3	13.8	3.03
29	0.1	0.1	0.2	speaker embedding	yes	yes	J	s_2	10.8M	12.8	12.2	3.15
30	0.1	0.1	0.2	speaker embedding	yes	yes	J	s_3	10.8M	12.8	12.2	3.15
31	0.1	0.1	0.2	speaker embedding	yes	yes	J	s_w	10.8M	14.9	14.4	3.13

TABLE 5.6: SDR (dB), SI-SDR(dB) and PESQ of extracted speech for the proposed SpEx network and other 4 competitive frequency-domain systems under open condition. "Mixture" refers to original input mixture with zero effort. "#Paras" means the number of parameters of the model.

Methods	Domain	#Paras	SDR	SI-SDR	PESQ
Mixture	-	-	2.6	2.5	2.31
SBF-IBM [86]	Frequency	19.3M	6.5	6.3	2.32
SBF-MSAL		19.3M	9.6	9.2	2.64
SBF-MTSAL		19.3M	9.9	9.5	2.66
SBF-MTSAL-Concat		8.9M	11.0	10.6	2.73
SpEx	Time	10.8M	**15.1**	**14.6**	**3.14**

with multi-task learning achieves relative improvements of 8.6% in terms of SDR, 9.8% in terms of SI-SDR, 4.7% in terms of PESQ. As the SpEx network with joint optimization (Figure 4.5) achieves the best performance, we use the configuration hereafter.

5.4.2.4 Benchmark against the Baselines

We compare the SpEx network as illustrated in Figure 4.5 with four competitive frequency-domain speaker extraction systems in Section 5.4.1. As can be seen in Table 5.6, the SpEx network shows 37.3%, 37.7% and 15.0% relative improvements over the best SBF-MTSAL-Concat system in terms of SDR, SI-SDR and PESQ under the open condition. The time-domain speaker extraction architecture has shown three clear advantages over its frequency-domain counterparts.

(1) Because the SpEx network doesn't decompose the speech signal into magnitude and phase spectra, it avoids inexact phase estimation.

(2) The SpEx network benefits from the long-range dependency of the speech signal captured by the stacked dilated depth-wise separable convolution with a manageable number of parameters. Without the recurrent connection, the SpEx method can be easily parallelized for fast training and inference.

(3) The SpEx network takes advantage of multi-scale speech embedding to have a good coverage of time-frequency resolution in the encoding, which doesn't have to trade time resolution with frequency resolution like in short-time frequency analysis.

TABLE 5.7: SDR (dB) and PESQ in a comparative study of different and same gender mixture under open condition.

Methods	SDR		PESQ	
	Diff.	Same	Diff.	Same
Mixture	2.5	2.7	2.29	2.34
SBF-IBM [86]	7.6	5.1	2.42	2.19
SBF-MSAL	12.0	6.9	2.82	2.43
SBF-MTSAL	12.3	7.2	2.85	2.44
SBF-MTSAL-Concat	12.9	8.8	2.90	2.54
SpEx	**17.4**	**12.4**	**3.34**	**2.92**

As an example, we illustrate the speaker extraction from a female-female mixture speech by the competitive baseline systems and the proposed SpEx network in Figure 5.1. From the log magnitude spectrum, we observe that the proposed SpEx network outperforms other baseline systems in terms of the recovered signal quality and purification. Some listening examples are available online [4], of which the first example is the audio illustrated in Figure 5.1.

5.4.2.5 Different Gender vs. Same Gender

Generally speaking, speakers of the same gender sound closer than those of different gender. We further report the results of the experiments in Table 5.6 for different and same gender mixture separately. We observe in Table 5.7 that the performance of different gender mixture is always better than the same gender. This has been observed in human listening test as reported by Treisman [125] in a behavioural study. It was found that difference in voice (i.e., male versus female) allows more efficient rejection of the irrelevant signal when messages are mixed and played to both ears (i.e., diotic).

From Table 5.7, we also observe that the proposed SpEx network achieves 34.9% and 40.9% relative SDR improvement, and 15.2% and 15.0% relative PESQ improvement over the best baseline, SBF-MTSAL-Concat, for different and same gender conditions.

[4] `https://xuchenglin28.github.io/files/taslp2019/index.html`

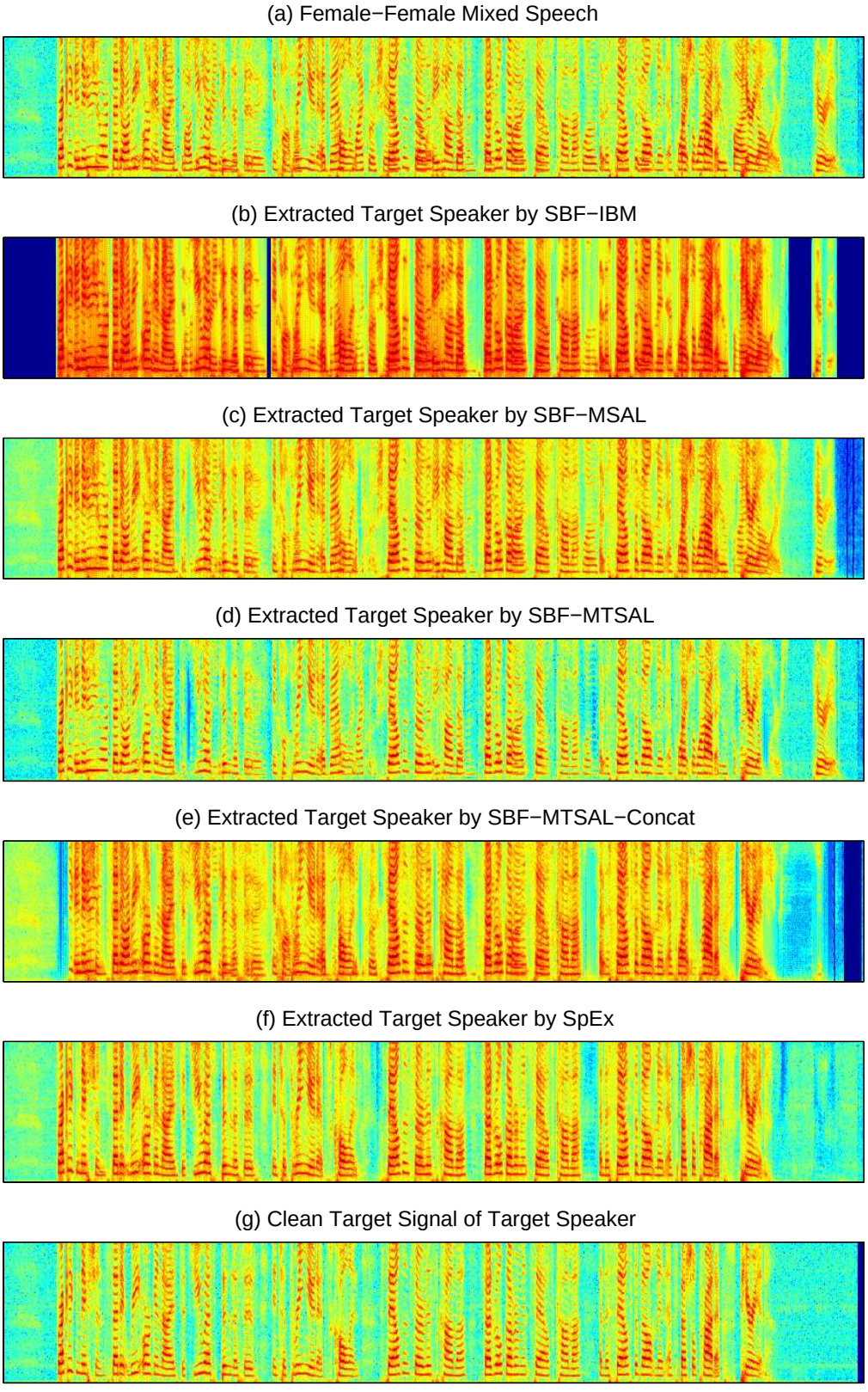


FIGURE 5.1: The log magnitude spectra of a female-female mixture, its extracted speech for target speaker by the four baselines, the proposed SpEx network, and the clean speech from target speaker.

TABLE 5.8: SDR (dB) of extracted speech when we evaluate the same SpEx system on varying duration of reference speech of target speaker at [0, 1)dB, [1, 3)dB, [3, 5]dB.

Methods \ SNR(dB)	[0, 1)	[1, 3)	[3, 5]
Mixture	0.7	2.0	4.2
SBF-IBM [86]	4.0	5.8	8.4
SBF-MSAL	7.1	9.2	11.3
SBF-MTSAL	7.5	9.5	11.5
SBF-MTSAL-Concat	8.7	10.6	12.5
SpEx	**13.3**	**14.8**	**16.3**

5.4.2.6 Mixture with Different SNR

It is of interest to investigate how the proposed SpEx network performs for mixture speech of different SNR, where we consider the target speech as the foreground and the interference as the background noise. We train a SpEx network on the dataset that has the SNR range of [0-5] as described in Section 5.1.1. The same SpEx network has been reported in Tables 5.6 and 5.7.

We divide the test set into 3 SNR groups, namely [0, 1)dB, [1, 3)dB and [3, 5]dB. The results are summarized in Table 5.8. Without surprise, test data of higher SNR performs better than that of lower SNR. We also observe that the proposed SpEx network achieves 52.9%, 39.6% and 30.4% relative SDR improvement over the best baseline system, SBF-MTSAL-Concat, for [0, 1)dB, [1, 3)dB and [3, 5]dB SNR group respectively. Since the SNR of the simulated database is limited from 0dB to 5dB, in the future work, we will investigate various SNR ranges, i.e., from -10dB to 20dB.

5.4.2.7 Subjective Evaluation

Since the SBF-MTSAL-Concat represented the best baseline performance in the objective evaluation, we only conducted an A/B preference test between the proposed SpEx network and the SBF-MTSAL-Concat baseline to evaluate the signal quality and intelligibility in a listening test. We randomly selected 20 pairs of listening examples, including the original target speaker's reference and two extracted signals for the target speaker by the proposed SpEx network and the best baseline system. We invited a group of 13 subjects to select their preference according to

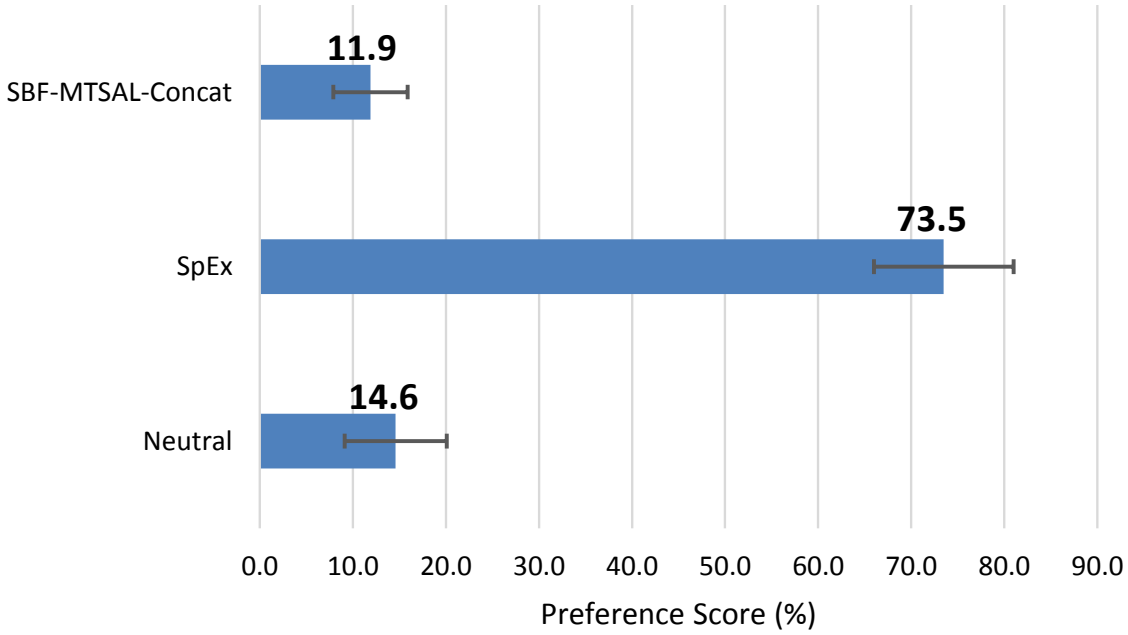


FIGURE 5.2: The A/B preference test result of the extracted target speaker's voice between the proposed SpEx method and the best SBF-MTSAL-Concat baseline. We conducted t-test using a significance level of $p < 0.05$ which is depicted with the error bars.

the quality and intelligibility. The listeners were asked to pay special attention to the amount of perceived distortion and interference from background. For each test, the subject listened to three audios in a group, the reference speech was firstly played, followed by the extracted speech in random order from the two systems. The subject didn't have the information about which speech stemmed from which system.

We observe from Figure 5.2 that the listeners clearly favor the proposed SpEx network with a preference score of 73.5% as opposed to that of 11.9% for the best SBF-MTSAL-Concat system. Most listeners significantly favor the SpEx network with a significance level of $p < 0.05$, because of lower distortion and inter-speaker interference than the best baseline.

5.4.2.8 Duration of the Reference Speech

As speaker extraction relies on the reference speech of the target speaker to develop the top-down voluntary focus, the duration of the reference speech plays a role in the process. We further look into the impact of the duration on speaker extraction

TABLE 5.9: SDR (dB), SI-SDR(dB) and PESQ in a comparative study of different duration of the reference speech. "Random" indicates that the duration of the reference speech is random.

Training	Test	SDR	SI-SDR	PESQ
Random	Random	15.1	14.6	3.14
	7.5s	15.0	14.6	3.14
	15s	15.4	15.0	3.17
	30s	15.5	15.2	3.19
	60s	**15.6**	**15.2**	**3.19**
15s	15s	14.9	14.5	3.13
	60s	15.2	14.8	3.15

performance. In the aforementioned experiments, the duration of the reference speech in training, development and test sets is at "Random" as described in Section 5.1.1. Now let's compare the "Random" setting with different duration groups (7.5s, 15s, 30s and 60s) in the test set. The experimental results are summarized in Table 5.9.

Since the average duration of the reference speech in the "Random" condition of the test set is 7.3s, we firstly evaluate the performance on the test subset with reference speech of a duration 7.5s. It is noted that the results are similar between "Random" condition and the 7.5s subset. When we increase the duration of the reference speech in the test set to 15s, 30s and 60s, we observe that longer duration leads to better results in general. When we fix the duration of the reference speech to 15s for the training and development set, the performance drops slightly when comparing with those under the "Random" condition. However, we continue to observe that longer test speech duration always helps.

5.5 Evaluation of Speaker Extraction on WSJ0-3mix-extr Database

The proposed SpEx network has the inherent ability to extract speech from mixture speech of more than two speakers using the same network architecture. We train the SpEx system under three conditions: only two-speaker mixture data, only three-speaker mixture data, and the combination of two- and three-speaker mixture data. We then evaluate the performance of the trained SpEx systems on two-speaker and three-speaker mixed test data, respectively. From Section 5.4.2.8, we know that the

TABLE 5.10: SDR (dB), SI-SDR(dB) and PESQ in a comparative study of different number of speakers in the mixed speech on WSJ0-2mix-extr and WSJ0-3mix-extr datasets. The duration of the reference speech is random during training. "#speakers" indicates the number of speakers in the mixture. "Dur." indicates the duration of the reference speech.

Training	Test		SDR	SI-SDR	PESQ
	#speakers	Dur.			
2 speakers	2 speakers	15s	15.4	15.0	3.17
	3 speakers	15s	5.2	5.0	2.35
3 speakers	2 speakers	15s	11.5	10.9	2.74
	3 speakers	15s	7.9	7.3	2.40
2 & 3 speakers	2 speakers	15s	15.0	14.6	3.14
	3 speakers	15s	8.9	8.4	2.54
2 speakers	2 speakers	60s	15.6	15.2	3.19
	3 speakers	60s	5.2	5.0	2.36
3 speakers	2 speakers	60s	12.1	11.6	2.81
	3 speakers	60s	8.3	7.8	2.44
2 & 3 speakers	2 speakers	60s	15.5	15.1	3.19
	3 speakers	60s	9.1	8.7	2.57

longer duration of the reference speech in the test set achieves better performance. We keep the duration of the reference speech as 15s and 60s for a comparison for both two-speaker and three-speaker mixed test data in this experiment.

From Table 5.10, we observe that the performance of the two-speaker mixture is always better than the three-speaker mixture in the SpEx systems under three conditions with different training data. This is consistent with the findings in a human's performance of a subject evaluation where both listening comprehension and auditory attention decrease significantly as the number of simultaneous audio channels increased [126]. It further confirms that the longer duration of the reference speech achieves better performance. Because the longer duration of the reference speech derives better speaker embedding.

5.6 Evaluation of Speaker Extraction on WSJ0-2mix Database

Most speech separation methods conducted their experiments on the well-known WSJ0-2mix database. To compare with speech separation methods, we trained the proposed SpEx model on WSJ0-2mix database to extract each speaker in the mixture by giving a reference speech of the corresponding speaker. Since most speech

TABLE 5.11: SDRi (dB), SI-SDR(dB) and PESQ in a comparative study on the WSJ0-2mix dataset under the open condition. "Ref: Avg. 7.3s" means the duration of the reference speech in the test set is random with an average duration of 7.3s. "Ref:60s" indicates the duration of the reference speech is fixed as 60s in the test set. The duration of reference speech in SpEx is always random during training. "#Paras" refers to the number of parameters of the model. ‡ indicates the latest Conv-TasNet with an additional skip-connection in each TCN block. For speech separation (SS) task, we report the results evaluated on the oracle-selected streams. For speaker extraction (SE) task, we report the results evaluated on the SpEx-extracted stream. The SI-SDR on the original mixture speech is 0dB, thus the SI-SDR is same as the value of SI-SDR improvement (SI-SDRi) that are reported in some works.

Task	Methods	Domain	#Paras	SDRi	SI-SDR	PESQ
SS	DC++ [27]	Frequency	13.6M	-	10.8	-
	uPIT-BLSTM-ST [32]		92.7M	10.0	-	-
	DANet [29]		9.1M	-	10.5	-
	SDC-G-MTL		53.9M	10.5	-	-
	CBLDNN-GAT [127]		39.5M	11.0	-	-
	Chimera++ [28]		32.9M	12.0	11.5	-
	WA-MISI-5 [79]		32.9M	13.1	12.6	-
	BLSTM-TasNet [34]	Time	23.6M	13.6	13.2	-
	Conv-TasNet [111]		8.8M	15.0	14.6	3.25
	Conv-TasNet‡ [35]		5.1M	15.6	15.3	3.24
SE	SpEx (Ref:Avg. 7.3s)	Time	10.8M	16.3	15.8	3.35
	SpEx (Ref:60s)		10.8M	**17.0**	**16.6**	**3.42**

separation methods didn't discard the segments less than 4s, we re-implemented the proposed SpEx system using Pytorch with the same scheme that breaks the utterances into 4s and pads zeros to the segments less than 4s [5].

From Table 5.11, we observe that the proposed SpEx achieves best performance over other speech separation techniques. Compared with the Conv-TasNet [111] method with the same TCN architecture, SpEx achieves 13.3%, 13.7% and 5.2% relative improvement in terms of SDRi, SI-SDR and PESQ. In addition, just like other speaker extraction techniques, SpEx offers its unique advantages over other speech separation techniques in real-world applications. The problems of unknown number of speakers and global speaker permutation ambiguity in speech separation are addressed in the SpEx system.

As SpEx relies very much on the quality of the speaker embeddings, we observed that the proposed speaker encoder has outperformed i-vector encoder (refer to

[5] We use this SpEx system implemented with Pytorch hereafter and the activation function in the mask estimation layer is tuned to be ReLU.

TABLE 5.12: SDRi (dB), SI-SDRi(dB) and PESQ in a comparative study on the WHAM! dataset under the open condition, where the mixture is corrupted with additive noise. "Ref: Avg. 7.3s" means the duration of the reference speech in the test set is random with an average duration of 7.3s. "Ref:60s" indicates the duration of the reference speech is fixed as 60s in the test set. The duration of reference speech in SpEx is always random during training. For speech separation (SS) task, we report the results evaluated on the oracle-selected streams. For speaker extraction (SE) task, we report the results evaluated on the SpEx-extracted stream. The SDR and SI-SDR of the origianl mixture speech are −4.23dB and −4.5dB.

Task	Methods	Domain	SDRi	SI-SDRi	PESQ
SS	Chimera++ [46]	Frequency	-	9.9	-
	BLSTM-TasNet [46]	Time	-	9.8	-
SE	SpEx (Ref:Avg. 7.3s)	Time	13.0	12.2	2.41
	SpEx (Ref:60s)		**13.7**	**13.1**	**2.48**

Table 5.5). We will further investigate the performance of SpEx on the speaker database larger than WSJ0-2mix (101 speakers) in the future work.

5.7 Evaluation of Speaker Extraction on WHAM! Database

We further evaluate the SpEx system on WHAM! database, which extends the WSJ0-2mix by adding additive noises into the mixture. Compared with the results between Table 5.11 and Table 5.12, we observe that the additive noise in the mixture significantly degrades the performance of both speech separation and speaker extraction systems. The SDR and SI-SDR of the origianl mixture speech are −4.23dB and −4.5dB. It shows the energy of wanted speaker's voice is lower than that of the interference speaker and noise in the mixture. The separation or extraction of a speaker's voice from a noisy mixture condition is a challenging task. Although the performance is degraded by adding noise, the SpEx system for speaker extraction still achieves significantly better performance than Chimera++ and BLSTM-TasNet baselines for speech separation. Compared with Chimera++ and BLSTM-TasNet methods, SpEx achieves 32.3% and 33.7% relative improvements in terms of SI-SDRi, respectively.

TABLE 5.13: SDRi (dB), SI-SDRi(dB) and PESQ in a comparative study on the WHAMR! dataset under the open condition, where the mixture is corrupted with reverberation. "Ref: Avg. 7.3s" means the duration of the reference speech in the test set is random with an average duration of 7.3s. "Ref:60s" indicates the duration of the reference speech is fixed as 60s in the test set. The duration of reference speech in SpEx is always random during training. For speech separation (SS) task, we report the results evaluated on the oracle-selected streams. For speaker extraction (SE) task, we report the results evaluated on the SpEx-extracted stream. The SDR and SI-SDR of the origianl mixture speech are −0.12dB and −3.3dB.

Task	Methods	Domain	SDRi	SI-SDRi	PESQ
SS	Conv-TanNet [128]	Time	-	7.6	-
	BLSTM-TasNet [128]		-	8.9	-
	Pre-Enh+BLSTM-TasNet [128]		-	9.9	-
SE	SpEx (Ref:Avg. 7.3s)	Time	8.7	9.7	2.68
	SpEx (Ref:60s)		**9.5**	**10.8**	**2.77**

5.8 Evaluation of Speaker Extraction on WHAMR! Database

To further study the influence of reverberation in the mixture, we conduct the speech separation and speaker extraction experiments on WHAMR! database. The WHAMR! database consists of four conditions: the clean mixture, the additive noisy mixture, the reverberant mixture, and the noisy and reverberant mixture. The WHAMR! database is an extension of WSJ0-2mix and WHAM! databases. The clean mixture and the noisy mixture conditions are corresponding with the WSJ0-2mix and WHAM! databases, which have been evaluated in Section 5.6 and Section 5.7, respectively. Thus, this section mainly focuses on the other two conditions.

We first conduct experiments on reverberant mixture condition to compare between speech separation and speaker extraction methods. From Table 5.13, we observe that the SpEx system achieves better performance than Conv-TasNet and BLSTM-TasNet speech separation methods. Compared with the Pre-Enh+BLSTM-TasNet system with a speech enhancement module as a pre-processing, the SpEx system still achieves 9.1% relative improvement in terms of SI-SDRi. Compared with the results between Table 5.11 and Table 5.13, we observe that the reverberation in the mixture also significantly degrades the performances of both speech separation and speaker extraction systems.

TABLE 5.14: SDRi (dB), SI-SDRi(dB) and PESQ in a comparative study on the WHAMR! dataset under the open condition, where the mixture is corrupted with both additive noise and reverberation. "Ref: Avg. 7.3s" means the duration of the reference speech in the test set is random with an average duration of 7.3s. "Ref:60s" indicates the duration of the reference speech is fixed as 60s in the test set. The duration of reference speech in SpEx is always random during training. For speech separation (SS) task, we report the results evaluated on the oracle-selected streams. For speaker extraction (SE) task, we report the results evaluated on the SpEx-extracted stream. The SDR and SI-SDR of the origianl mixture speech are -3.49dB and -6.1dB.

Task	Methods	Domain	SDRi	SI-SDRi	PESQ
SS	Conv-TanNet [128]	Time	-	8.3	-
	BLSTM-TasNet [128]		-	9.2	-
	Pre-Enh+BLSTM-TasNet+Post-Enh [128]		-	10.1	-
SE	SpEx (Ref:Avg. 7.3s)	Time	9.5	10.3	2.24
	SpEx (Ref:60s)		**10.1**	**11.1**	**2.30**

TABLE 5.15: SDRi (dB), SI-SDRi(dB) and PESQ in an universal study of the SpEx system, which is trained on 4 conditions and tested on each condition individually. "Ref: Avg. 7.3s" means the duration of the reference speech in the test set is random with an average duration of 7.3s. "Ref:60s" indicates the duration of the reference speech is fixed as 60s in the test set. The duration of reference speech in SpEx is always random during training.

Methods	Condition		SDRi	SI-SDRi	PESQ
	Training	Test			
SpEx (Ref:Avg. 7.3s)	4 Conditions	Clean Mixture	14.8	14.1	3.22
		Noisy Mixture	13.5	12.7	2.49
		Reverberant Mixture	8.8	9.8	2.74
		Noisy and Reverberant Mixture	10.1	10.8	2.31
SpEx (Ref:60s)	4 Conditions	clean mixture	15.9	15.4	3.32
		Noisy Mixture	14.3	13.8	2.57
		Reverberant Mixture	9.7	10.8	2.83
		Noisy and Reverberant Mixture	10.8	11.7	2.38

We further conduct the experiments on the mixture with additive noise and reverberation, as summarized in Table 5.14. Compared with Conv-TasNet and BLSTM-TasNet, the SpEx system achieves 33.7% and 20.7% relative improvements in terms of SI-SDRi. Although the performance of BLSTM-TasNet is further improved by applying a pre-processing and a post-processing speech enhancement modules, the SpEx system still achieves better performance without any additional processing.

Finally, we study an universal SpEx model that is trained on four mixture conditions (clean, noisy only, reverberant only, and noisy and reverberant). The performance of the universal SpEx system is further evaluated on each mixture condition, individually. The experimental results are summarized in Table 5.15. Compared with the results in Table 5.12, Table 5.13 and Table 5.14, we observe that the universal SpEx model trained on four conditions achieves better performance than

that trained on a single adversarial condition, i.e. noisy only condition, reverberant only condition, noisy and reverberant condition.

5.9 Conclusions

In this work, we propose an frequency-domain and an time-domain speaker extraction methods that emulate humans' ability of selective auditory attention. The speaker extraction method forms a top-down voluntary focus by using the reference speech of the target speaker without the need of the information about the number of speakers.

The SpEx network also overcomes the phase estimation issue in the frequency-domain speaker extraction. The improvements are attributed to the dilated convolutional encoder-decoder framework without decomposing the mixture audio into magnitude and phase spectrums, the multi-scale learning to capture different temporal resolutions, and the multi-task learning to jointly optimize the speaker encoder and speech extraction pipeline. Our experiments show that the SpEx network significantly outperforms the frequency-domain speaker extraction methods when we use a comparable number of parameters.

The ability of human to detect a particular signal from other interference speech or background noise is greatly degraded. The experimental results of the SpEx system are also made the same conclusions on these adversarial conditions. Although the performance of the SpEx system is degraded by noise and reverberation, the SpEx system always achieves better performance than other speech separation methods. Meanwhile, the problems of speech separation are addressed inherently by the speaker extraction technique.

In summary, the proposed SpEx network marks a significant step towards solving the cocktail party problem. It will potentially improve the performance of many down-stream speech processing applications, such as speaker verification and speaker diarization.

Chapter 6

Multi-talker Speaker Verification with Speaker Extraction

Nowadays, smart devices become more and more popular in our daily life. Speech is a nature way to interact with these smart devices. To make the smart devices secure, speaker verification becomes an important role to identify and only response to the owner. However, the performance of speaker verification degrades significantly when the speech is corrupted by interference speakers most of the time.

In this chapter, we propose a multi-talker speaker verification framework with speaker extraction to solve the problem that the test speech is corrupted by interference speakers. To the best of my knowledge, this is the first work to address the overlapped multi-talker speaker verification problem. Section 6.1 introduces the problem of speaker verification when the test utterance is corrupted by interference speaker most of the time. Section 6.2 describes the proposed multi-talker speaker verification approach with speaker extraction as a front-end. Experimental setups and results are described in Section 6.3 and 6.4. Section 6.5 concludes this work.

The work in this chapter has been published in [44].

6.1 Problem of Multi-talker Speaker Verification

The performance of speaker verification degrades significantly when the test speech is corrupted by interference speakers. When the speech segments of the target speaker and interference speaker are non-overlapped in a multi-talker speech, speaker diarization may work well for speaker verification [129–134]. Speaker diarization is effective to segment the speakers and exclude unwanted speech segments of the interference speakers even when the speakers in a multi-talker speech have a slight overlap [135, 136]. However, such speaker diarization system fails when multiple talkers speak simultaneously most of the time.

Since the multi-talker speech is mixed in temporal and spectral directions, one possible solution is to separate the multi-talker speech into different output streams with speech separation techniques. Each output stream only contains one individual speaker. Then, the speaker verification system compares each output stream with the enrolled speech to identify whether the output stream is the enrolled speaker. As we know, the speech separation system requires the number of speakers in the multi-talker speech to be known as a prior. However, the number of speakers is always unknown in most real-world applications.

As discussed in Chapter 4 and 5, speaker extraction techniques don't need to know the number of speakers in advance. In addition, speaker verification always knows the enrolled speech, which could be used as the reference speech of the target speaker in speaker extraction. Furthermore, speaker extraction only outputs the target speaker's voice if the target speaker is in the multi-talker test speech. The speaker verification system only needs to verify whether the extracted speech is the same speaker as the enrolled speech once. Thus, speaker extraction is the most suitable technique to address the performance degradation of speaker verification when the test speech is an overlapped multi-talker speech.

6.2 Multi-Talker Speaker Verification with Speaker Extraction

With an overlapped multi-talker speech as the input, speaker verification is only interested in whether the target speaker described by the enrolled speech is in

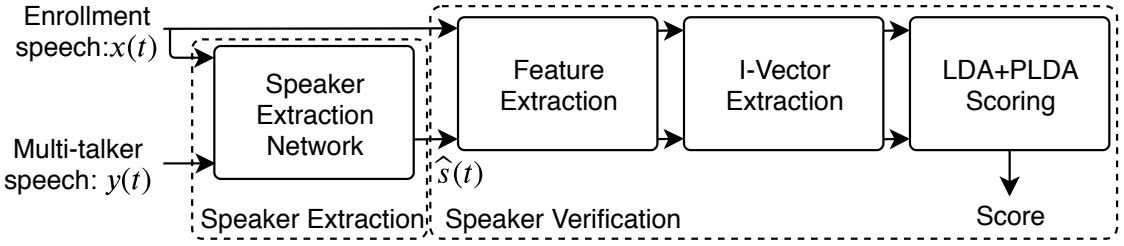


FIGURE 6.1: The flow chart of overlapped multi-talker speaker verification system. It consists of a speaker extraction module and a traditional speaker verification module. "$y(t)$" represents the overlapped multi-talker speech. "$x(t)$" represents the enrollment speech, which is also used as the reference speech in speaker extraction. "$\hat{s}(t)$" represents the extracted target speaker's speech from $y(t)$.

the multi-talker speech. The speaker verification system doesn't care how many interference speakers in the multi-talker speech. In this chapter, we propose an overlapped multi-talker speaker verification framework, which consists of a speaker extraction module and a traditional speaker verification module, as shown in Figure 6.1. We name the proposed pipeline as SE-SV. The speaker extraction module is used as a front-end to extract the target speaker's voice if the target speaker is in the multi-talker speech. The traditional speaker verification module could be any monaural speaker verification system with a single speaker's test speech and an enrolled speech as inputs, for example, i-vector/PLDA system [107, 137–139].

Specifically, the speaker extraction takes in the multi-talker speech $y(t)$ and a reference speech $x(t)$ of the target speaker, where the reference speech is also the enrolled speech of the following speaker verification system. If the target speaker is existed in the multi-talker speech, the target speaker's voice $\hat{s}(t)$ is extracted and used as the test speech of the speaker verification system. If the multi-talker speech doesn't contain the target speaker's voice, the extracted speech $\hat{s}(t)$ may be silence or random segments of the multi-talker speech. Then, the speaker verification system further verifies whether the extracted speech $\hat{s}(t)$ belongs to the target speaker. If the extracted speech belongs to the target speaker, it means that the multi-talker speech has the target speaker's command. Then, the speech application may authorize and response to the command.

This chapter evaluates the proposed frequency-domain and time-domain speaker extraction techniques in Chapter 4 and 5 for the overlapped multi-talker speaker verification. We compare the usage of three target speaker extraction methods as front-end and their interaction with the back-end speaker verification system: (1)

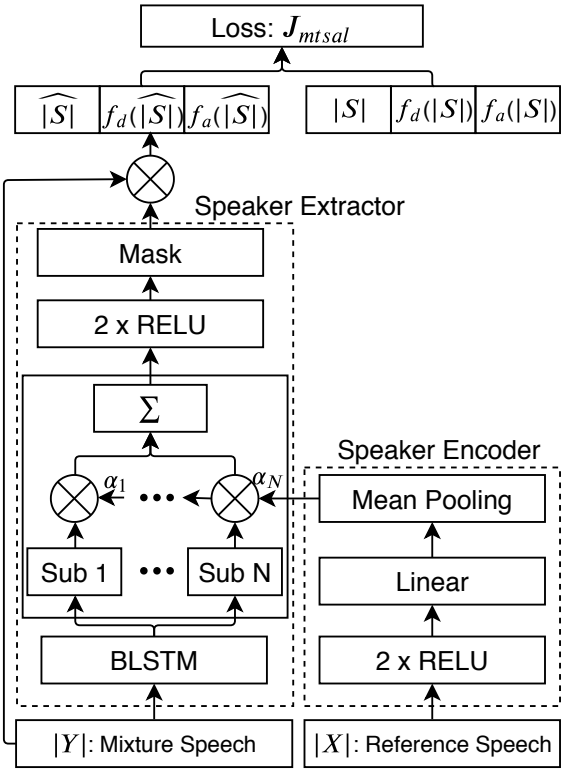


FIGURE 6.2: The architecture of the SBF-MTSAL method. "Sub" indicates the sub-layer. α represents the adaptation weights associated with the target speaker. "N" is the number of sub-layers, which is also equal to the dimension of the adaptation weights. $|\hat{S}|$ and $|S|$ are the extracted magnitude and the target clean magnitude, respectively. $f_d(\cdot)$ and $f_a(\cdot)$ are the delta and acceleration computation functions. During the inference, the calculation of the delta and acceleration is not necessary.

SpeakerBeam front-end with magnitude and temporal spectrum approximation loss (SBF-MTSAL), (2) SBF-MTSAL with a concatenation framework (SBF-MTSAL-Concat), and (3) Multi-scale time-domain speaker extraction (SpEx).

6.2.1 SBF-MTSAL

The SBF-MTSAL architecture is composed of a speaker extractor and a speaker encoder, as shown in Figure 6.2. The speaker extractor applies an adaptation layer to focus attention to the target speaker using the adaptation weights, which are associated with the target speaker in the speaker encoder. The adaptation layer is a context adaptive deep neural network (CADNN) structure [87].

Specifically, the speaker extractor takes the magnitude of the multi-talker speech ($|Y|$) as inputs. The BLSTM transforms the magnitude into representations including both the target speaker and the interference speakers. Then, the adaptation layer only lets pass the target speaker by weighting and summing the contributions of the sub-layers. The adaptation weights ($\alpha = [\alpha_1, ..., \alpha_N]$) are learned from a reference speech ($|X|$) of the target speaker. The reference speech is from the enrollment speech in the multi-talker speaker verification task. With the target speaker information encoded in the adaptation weights, the speaker extractor estimates a phase sensitive mask for the target speaker. Then, the extracted magnitude of the target speaker is obtained by element-wise multiplying the estimated phase sensitive mask and the magnitude of the multi-talker speech. In the inference stage, the extracted magnitude ($|\hat{S}|$) is used to reconstruct the signal together with the phase of the multi-talker speech through iSTFT.

To train the speaker extraction network, we compute a magnitude and temporal spectrum approximation loss, as defined in Eq. 4.6. In particular, the delta and acceleration of the extracted magnitude of the target speaker are used as the temporal spectrum in the objective function. Similarly, the magnitude and its dynamic information (i.e., delta and acceleration) are also obtained as the supervision. The MSEs between these corresponding magnitudes and their dynamic information are calculated to jointly optimize the speaker extraction network. After the speaker extractor and the speaker encoder are well trained, the speaker extraction network is used as the front-end of the multi-talker speaker verification system, as described in Figure 6.1.

6.2.2 SBF-MTSAL-Concat

As discussed in Chapter 4.3.2, the SBF-MTSAL-Concat approach also consists of a speaker extractor and a speaker encoder, as shown in Figure 4.3. Different from the SBF-MTSAL method, the speaker extractor and the speaker encoder are communicated through a concatenation structure. In particular, the speaker embeddings learned from the speaker encoder are concatenated to every frame of the representations, which are transformations of the magnitude of the multi-talker speech. At the same time, the speaker encoder is trained together with the speaker extractor by back-propagating the errors from the speaker extractor to the speaker

encoder. A stack of BLSTMs and non-linear ReLU layers are applied to estimate a phase sensitive mask using a magnitude and temporal spectrum approximation loss, as defined in Eq. 4.6.

In the SBF-MTSAL method, the speaker encoder learns the adaptation weights to make the speaker extractor form an attention to the target speaker. The adaptation weights are learned to associate with the speaker information with DNN. Different from the SBF-MTSAL method, the SBF-MTSAL-Concat approach learns a speaker embedding in the speaker encoder using a BLSTM. Unlike the DNN, the BLSTM has the ability to model the context of the reference speech of the target speaker. With the forward and backward paths, the BLSTM transforms the reference speech into better representations.

Same as the SBF-MTSAL method, the speaker extractor and the speaker encoder are jointly optimized using some simulated data in the SBF-MTSAL-Concat approach. After the networks are well trained, the speaker extraction is used as a front-end processing of the multi-talker speaker verification system to obtain the extracted speech.

6.2.3 SpEx

As discussed in Section 4.4, the multi-scale time-domain speaker extraction consists of four network components, a speaker encoder, a speech encoder, a speaker extractor, and a speech decoder, as shown in Figure 4.5. The speaker encoder encodes the reference speech $x(t)$ of the target speaker into a speaker embedding. The speaker embedding describes the speaker characteristic, which is an important information to make the speaker extraction focus on the the target speaker. The speech encoder transforms the time-domain multi-talker speech $y(t)$ into spectrum or spectrum-like feature representation. The speaker extractor estimates a mask that only let pass the target speaker's voice together with the speaker embedding. Finally, the speech decoder reconstructs the time-domain speech signal of the target speaker from the masked spectrum of the mixture speech.

Same as SBF-MTSAL-Concat, the speaker encoder applies a BLSTM to leverage the contextual information of the reference speech to obtain better speaker embedding. Different from the frequency-domain speaker extraction methods (SBF-MTSAL and SBF-MTSAL-Concat), the speaker encoder is jointly optimized with a cross-entropy loss and a SI-SDR loss through a multi-task learning. The cross-entropy loss is used for speaker classification. At the same time, the SI-SDR loss evaluates the speech quality of the speaker extraction. A weight is used to balance the contribution of these two losses in optimizing the speaker encoder.

The frequency-domain speaker extraction methods always apply a STFT and iSTFT as speech encoder and speech decoder to transform and reconstruct the time-domain signal, respectively. However, the phase estimation problem remains a challenge task in the frequency-domain methods. Different from SBF-MTSAL and SBF-MTSAL-Concat, SpEx transforms and reconstructs the time-domain signal with trainable speech encoder and decoder instead of STFT and iSTFT. Without decomposing the time-domain mixture signal into magnitude and phase spectra, SpEx inherently avoids the phase estimation problem. In addition, SpEx also adopts a multi-scale encoding and decoding scheme to capture complementary feature representation with various temporal resolutions.

Different from SBF-MTSAL and SBF-MTSAL-Concat methods with BLSTM, SpEx adopts temporal convolution network in the speaker extraction pipeline. The main reason is that the time-domain speech processing always uses a small window to encode the signal. It thus results in a huge number of frames, which causes gradient vanish problem to model plenty of frames in a BLSTM. The speaker extractor exploits a stacked depth-wise separable dilated CNN to estimate a filter mask for every scale of the encoded feature representations from the speech encoder. The stacked dilated structure captures the long-range dependency of the speech signal. Meanwhile, the depth-wise separable framework reduce the number of parameters for a manageable computation cost. Then, the estimated mask for each scale is used to extract the embedding spectrum of the target speaker. Different from frequency-domain methods using a MSE loss between the estimated and clean magnitude spectra, a multi-scale SI-SDR loss is calculated between the reconstructed and clean time-domain signals to optimize the four network components jointly.

6.3 Experimental Setup

6.3.1 Speaker Extraction Database

To train the speaker extraction systems, we simulated a two-speaker mixture database, which was same as the WSJ0-2mix-extr database[1] for two-speaker mixture experiments in Chapter 4 and 5. Specifically, we randomly selected two utterances of two different speakers from the "si_tr_s" set of the WSJ0 corpus with 50 male and 51 female speakers to simulate a training set and a development set at a sampling rate of 8kHz. The training set and the development set included $20,000$ and $5,000$ mixtures, respectively. Similarly, the test set included $3,000$ mixtures which were generated by randomly mixing the utterances selected from the "si_dt_05" and "si_et_05" sets of the WSJ0 corpus with 10 male and 8 female speakers in total.

In the simulation of two-speaker mixture, the first selected speaker was used as the target speaker, the other one was considered as the interference speaker. The first selected utterance from the original WSJ0 corpus was used as the target clean speech to supervise the network optimization. Another different utterance of the target speaker was randomly selected to be used as the reference speech to form the attention to this target speaker.

Considering the interference speech as noise, each two-speaker mixture was generated at a random SNR between 0dB and 5dB based on the rule of maximum duration. For example, if the duration of first utterance was 10 seconds and that of the second utterance was 5 seconds, the duration of the two-speaker mixture would be 10 seconds. Therefore, the overlapped percentage of the two-speaker mixture was 50%. The overlapped percentages of the two-speaker mixture in the training set, development set, and test set were shown in Figure 6.3. Most of two-speaker mixtures were highly overlapped with the average length of 8.5 seconds.

[1]The database simulation code is available at: `https://github.com/xuchenglin28/speaker_extraction`

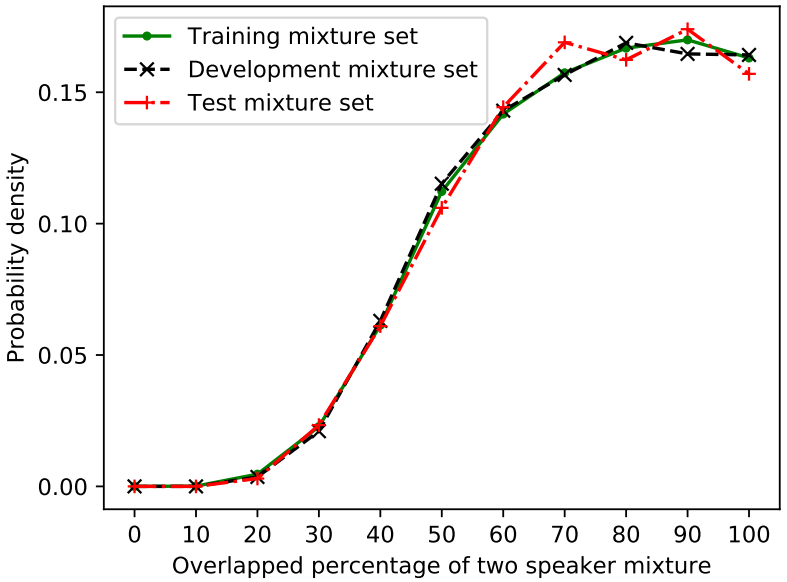


FIGURE 6.3: The overlapped percentages of the two-speaker mixture in the training set, development set, and test set.

6.3.2 Speaker Verification Database

As we known, the speaker extraction database (WSJ0-2mix-extr) was simulated by randomly selecting utterances from the WSJ0 corpus. These selected utterances from the WSJ0 corpus were considered as *Clean Dataset*, which was used for the speaker verification experiments. For the training set (20,000 mixtures) and development set (5,000 mixtures) in the WSJ0-2mix-extr database, there were 8,769 and 2,791 different utterances selected from the "si_tr_s" set of the WSJ0 corpus with 50 male and 51 female speakers in total. These 8,769 and 2,791 utterances were used as the training set, and development set of the *Clean Dataset*, respectively. The training set and development set were used together to train a speaker verification system for the clean condition.

For the test set (3,000 mixtures) in the WSJ0-2mix-extr database, 1,478 different utterances were selected to simulate the mixtures from the "si_dt_05" and "si_et_05" sets of the WSJ0 corpus, which had 10 male and 8 female speakers. These 1,478 utterances were used as the test utterances of the evaluation set in the *Clean Dataset*. Meanwhile, 1,875 different utterances were selected as the reference speeches in speaker extraction, which were usually used as the enrollment utterances of the evaluation set in speaker verification. Since the speakers in the evaluation set were different from those in the training set and development set, the evaluation set

was used to evaluate the speaker verification performance. These test utterances and enrollment utterances formed the *Clean evaluation set* with 48,000 non-target trials and 3,000 target trials. The utterances in the *Clean Dataset* had an average duration of 7 seconds.

To evaluate the speaker verification performance with a multi-talker speech, we formed a *Mixture evaluation set* with 48,000 non-target trials and 3,000 target trials. The mixture speech and reference speech of the target speaker in these 3,000 target trials were same as the test set in the WSJ0-2mix-ext database. The 48,000 non-target trials were generated by pairing the simulated 3,000 mixture speech with a random utterance from other 16 speakers. For each mixture speech, the utterance of other 16 speakers, which were different from the target speaker and interference speaker in this mixture speech, was used as the enrollment speech in the speaker verification non-target trials.

6.3.3 Speaker Extraction Setup

We exploited three proposed speaker extraction approaches for the multi-talker speaker verification. The frequency-domain SBF-MTSAL and SBF-MTSAL-Concat methods extracted the magnitude of the target speaker. Then, the time-domain signal was reconstructed with the extracted magnitude and the phase of the mixture through inverse STFT. The details of the STFT configuration, network configuration, and training scheme were described in Section 5.2.

The third speaker extraction method was SpEx in time-domain. The SpEx method applied data-driven convolution and de-convolution operations to encode and reconstruct the signal instead of STFT and inverse STFT. The details of the network configuration, and training scheme were described in Section 5.3.

6.3.4 Speaker Verification Setup

As discussed in Section 6.3.2, the *Mixture evaluation set* was used to evaluate the performance of the proposed multi-talker speaker verification framework. It included 48,000 non-target trials and 3,000 target trials. Each trial consisted of an enrollment utterance of a single speaker and a multi-talker test utterance (two

speakers in this work). The multi-talker speaker verification framework applied a speaker extraction as a front-end. Thus, the extracted speech from the speaker extraction was assumed to be a single speaker. To show the upper bound performance of the multi-talker speaker verification using the speaker extraction, we generated a corresponding *Clean evaluation set* using the original clean utterances from the WSJ0 corpus according to the 51,000 trials of the *Mixture evaluation set*[2]. Each test utterance of the *Clean evaluation set* was the first speaker's utterance used to generate the mixture speech of the *Mixture evaluation set*. The enrollment utterances and identity keys of the *Clean evaluation set* were same as those of the *Mixture evaluation set*.

The training and development sets of the *Clean Dataset*, called as *Clean(training&dev)*, were used to train universal background model (UBM), total variability matrix, linear discriminate analysis (LDA), and Probabilistic LDA (PLDA) models. This speaker verification system was named as *Clean SV*. Since the *Clean SV* system was trained with clean utterances, the extracted speech from the speaker extraction system might cause a mismatch when the extracted speech was sent to the *Clean SV* system. To address the mismatch problem, we further trained the speaker verification system using both the clean utterances from the training set of the *Clean Dataset* and the extracted utterances from the development set of the WSJ0-2mix-extr database. For a fair comparison with the *Clean SV* system, we firstly pooled 2,791 extracted utterances and 8,769 clean utterances (named as *Clean(training)+Ext1 set*) to train a *Hybrid SV1* system. To investigate how was the performance of the speaker verification with more data, we trained another *Hybrid SV2* system with 8,769 clean utterances and 5,000 extracted utterances. These utterances were formed as a *Clean(training)+Ext2 set*. When the speaker extraction system (i.e., SpEx) was used as the front-end of the multi-talker speaker verification system, the corresponding extracted utterances using this speaker extraction system were used to train the *Hybrid SV1* and *Hybrid SV2* systems.

In this work, all the speaker verification systems were trained with 19 MFCCs together with energy plus their first and second order derivatives of the speech regions. A cepstral mean normalization [124] with a window size of 3 seconds was applied to normalize the features. The 60-dimensional acoustic features were computed with a Hamming window of 25ms and a shift of 10ms. The silence was

[2]The trials and keys of the *Clean and Mixture evaluation sets* for speaker verification evaluation are available at: `https://github.com/xuchenglin28/speaker_extraction`

removed by an energy based voice activity detection method. The speaker verification system was based on a gender-independent UBM with 512 mixtures. The UBM and total variability matrix were estimated with 400 total factors, resulting in a 400-dimensional speaker embedding. The LDA and PLDA were estimated with 150 latent variables.

6.4 Experimental Results

6.4.1 SBF-MTSAL and SBF-MTSAL-Concat

To investigate the effect of overlapped test speech on speaker verification (SV) system, we perform the SV experiments on both *Mixture and Clean evaluation set* as described in Section 6.3.4. System 1 of Table 6.1 is the baseline system of SV with clean training data on mixture test set. System 15 of Table 6.1 shows the upper bound performance (also called as ideal performance) of SE-SV for overlapped multi-talker SV. Comparison between System 1 and 15 of Table 6.1 shows that the performance of SV system seriously degrades when the test speech is corrupted by interference speaker's speech most of the time.

Table 6.1 presents the performances of SV systems with and without target speaker extraction. System 1 of Table 6.1 is the baseline results of overlapped multi-talker SV. Systems 6 to 8 and Systems 9 to 11 of Table 6.1 show the performances of proposed SE-SV approach with SBF-MTSAL and SBF-MTSAL-Concat as front-ends on multi-talker SV. The comparison of performances among System 1 to 11 suggests the following findings: (1) the proposed SE-SV approach significantly improve the performance of multi-talker SV, specifically, the SE-SV with SBF-MTSAL-Concat (System 11) on the overlapped multi-talker speech can obtain around 64.4%, 27.7%, 18.1% relative reduction over the baseline (System 1) in terms of EER, DCF08, and DCF10, respectively; (2) SE-SV with SBF-MTSAL-Concat outperforms SE-SV with SBF-MTSAL in terms of both EER and DCFs; (3) pooling the clean training set and extracted speech data is effective to alleviate the effect caused by the mismatch problem between clean and extracted speech; (4) more extracted utterances for SV training could further improve the performance of SE-SV on the overlapped multi-talker speech; (5) comparing System 2 with 7, and 10, we observe that most of improvement on the overlapped multi-talker speech is

TABLE 6.1: Performance of speaker verification (SV) system with and without speaker extraction. "Training" represents the type of training data. "Eval" represents the type of evaluation test data. "TSE" represents whether or which target speaker extraction method is used. "Baseline" represents the zero-effort test case where SV system is trained with clean data and evaluated on mixture data. "Upper Bound" represents the case where clean speech data are used in both training and testing, which offers the upper bound performance of multi-talker SV system. "OSD-SV" represents the case where we replace the speaker extraction network in Figure 6.1 with an oracle speaker diarization (OSD) system. "DCF08" represents the minimum detection cost with $P_{Target} = 0.01$. "DCF10" represents the minimum detection cost with $P_{Target} = 0.001$. †, ‡ and * represent the extracted data by SBF-MTSAL, SBF-MTSAL-Concat, and SpEx, respectively. The details of experimental setup can be referred to section 6.3.4.

System No.	**Systems**	**Training**	**Eval**	**TSE**	**EER (%)**	**DCF08**	**DCF10**
1 (Baseline)	SV	Clean(training&dev)	Mixture	No	21.80	0.873	0.912
2	SV	Clean(training)+Ext1‡	Mixture	No	21.57	0.854	0.926
3	SV	Clean(training)+Ext2‡	Mixture	No	21.67	0.850	0.898
4	SV	Clean(training)+Ext1*	Mixture	No	21.70	0.860	0.929
5	SV	Clean(training)+Ext2*	Mixture	No	21.23	0.838	0.883
6	SE-SV	Clean(training&dev)	Mixture	SBF-MTSAL	10.87	0.766	0.867
7	SE-SV	Clean(training)+Ext1†	Mixture	SBF-MTSAL	8.50	0.677	0.797
8	SE-SV	Clean(training)+Ext2†	Mixture	SBF-MTSAL	8.30	0.643	0.777
9	SE-SV	Clean(training&dev)	Mixture	SBF-MTSAL-Concat	10.37	0.736	0.861
10	SE-SV	Clean(training)+Ext1‡	Mixture	SBF-MTSAL-Concat	7.93	0.640	0.747
11	SE-SV	Clean(training)+Ext2‡	Mixture	SBF-MTSAL-Concat	7.77	0.631	0.747
12	SE-SV	Clean(training&dev)	Mixture	SpEx	7.60	0.632	0.748
13	SE-SV	Clean(training)+Ext1*	Mixture	SpEx	6.63	0.556	0.724
14	SE-SV	Clean(training)+Ext2*	Mixture	SpEx	**6.00**	**0.551**	**0.683**
15 (Upper Bound)	SV	Clean(training&dev)	Clean	No	3.00	0.360	0.522
16	SV	Clean(training)+Ext1‡	Clean	No	3.07	0.366	0.526
17	SV	Clean(training)+Ext2‡	Clean	No	3.07	0.377	0.524
18	SV	Clean(training)+Ext1*	Clean	No	3.17	0.387	0.524
19	SV	Clean(training)+Ext2*	Clean	No	2.87	0.335	0.531
20	OSD-SV	Clean(training&dev)	Mixture	No	14.60	0.851	0.908

attributed to the speaker extraction front-end in the SE-SV. The same conclusion could be made by comparing System 3, 8, and 11.

6.4.2 SpEx

As we known in Section 5.4.2.4, SpEx always outperforms SBF-MTSAL and SBF-MTSAL-Concat in terms of objective and subjective evaluations of speech quality and intelligibility. Comparing the performance among System 6 to 14, we observe that the proposed SE-SV with SpEx outperforms the SE-SV systems with both SBF-MTSAL and SBF-MTSAL-Concat in terms of EER and DCFs. The SE-SV with SpEx (System 14) on the overlapped multi-talker speech obtains around $22.8\%, 14.5\%, 8.6\%$ relative reduction over the SE-SV with the best frequency-domain speaker extraction (System 11) in terms of EER, DCF08, and DCF10, respectively. In addition, the SE-SV with SpEx (System 14) achieves $72.5\%, 36.9\%, 25.1\%$ relative reduction over the baseline (System 1) on the overlapped multi-talker speech in terms of EER, DCF08, and DCF10, respectively.

6.4.3 Speaker Extraction vs. Oracle Speaker Diarization

To investigate whether speaker diarization is effective on the multi-talker speech, we perform the multi-talker SV experiment with oracle speaker diarization. The oracle speaker diarization is an upper bound system using the target speaker's clean speech to find the start and end time of the target speaker in the multi-talker speech. Specifically, we first apply the energy-based VAD on the clean speech that makes up the mixture speech, then generate the diarization labels for the mixture speech according to the VAD labels of clean speech. Since the utterance in WSJ0 corpus is clean, energy-based VAD works well to find the speech segments of the target speaker with the clean speech. We consider these diarization labels as oracle diarization labels. With the oracle speaker diarization, the average percentage of removed non-target speech frames in the mixture speeches is around 24%. System 20 in Table 6.1 shows the performance of the multi-talker SV system with the oracle speaker diarization, which also means the best performance achieved by speaker diarization for multi-talker SV. The comparison of performance among System 1, 8, 11, 14, and 20 suggests that our proposed SE-SV method always

significantly outperforms the multi-talker SV with the oracle speaker diarization in the overlapped multi-talker scenarios.

6.5 Conclusions

In this chapter, we propose the SE-SV approach to improve the performance of SV on overlapped multi-talker speech. By exploiting the proposed speaker extraction methods, SBF-MTSAL and SBF-MTSAL-Concat, and SpEx in Chapter 4 and 5 as front-ends, the performance of SV is significantly improved on the overlapped multi-talker speech. Comparing with the multi-talker SV with an oracle speaker diarization, we further observe that the proposed SE-SV achieves significant improvement.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

Speech separation or speaker extraction plays an important role to improve the performance of speech applications in multi-talker environment, for example, a cocktail party. The goal of this thesis is to improve the speech separation and speaker extraction technologies, which are used to solve the cocktail party problem. This thesis has contributed three novel approaches to advance the state-of-the-art speech separation and speaker extraction technologies for better speech quality and intelligibility. In addition, this thesis also conducted a systematic study examining the proposed frequency and time-domain speaker extraction methods as front-ends for multi-talker speaker verification. The works of this thesis are summarized as follows.

7.1.1 Multi-task Learning of Neural Networks for Temporal Continuity in Speech Separation

Most frequency-domain speech separation methods typically use spectrum features that only contain the spectral information within each short-time frame. However, the temporal structure of speech is not considered. This thesis has exploited both spectral and temporal information and proposed to use temporal objective functions and spectro-temporal features captured by a grid LSTM to solve the frame

leakage problem together with a multi-task learning framework. Inspired by the grid LSTM communicating information in two dimensions, spectro-temporal features are captured by a grid LSTM on top of the image-like spectrum in both spectral and temporal directions through inner communications. The spectro-temporal features contain both spectral and temporal information. In addition, the temporal objective function computes a loss to consider both spectral and temporal information using dynamic features, for example, SDC. Finally, the novel multi-task learning framework incorporates the speech separation task with a sub-task that predicts time-frequency attributes (silence, single and overlapped) of each time-frequency bin. The explicit attributes of each time-frequency bin in the mixture helps the mask estimation for each individual speaker. The improved speech separation system is called as SDC-G-MTL.

Experiments are carried out using the WSJ0-2mix database. It is observed that the performance of speech separation is improved step by step using the temporal objective function, spectro-temporal features and the multi-task learning comparing with the uPIT-BLSTM method. Specifically,SDC-G-MTL achieves relative improvements by 10.5%, 9.1%, and 6.3% in terms of GNSDR, SIR, and SAR, respectively, comparing with the uPIT-BLSTM baseline. Meanwhile, SDC-G-MTL achieves a relative error reduction of 26% in terms of FLER over the uPIT-BLSTM baseline. It indicates a significant reduction of frame leakage and an impovement of speech quality for temporal continuity. An A/B preference listening test of subjective evaluation shows a 54% of preference for the proposed SDC-G-MTL while a 11% preference for the uPIT-BLSTM baseline. However, the number of speakers required as a prior during training couldn't always be known in practice. In addition, global speaker permutation ambiguity problem remains across utterances.

7.1.2 Top-down Selective Auditory Attention with Speaker Extraction

To address the problems of unknown number of speakers and global speaker permutation ambiguity in speech separation, this thesis has further studies a frequency-domain and a time-domain speaker extraction methods. Speaker extraction mimics humans' ability of selective auditory attention by extracting a target speaker's voice from a multi-talker environment. Speaker extraction takes a reference speech of

the target speaker as a cue to form the top-down selective auditory attention. The target speaker is always characterized by the speaker embedding encoded with a speaker encoder. The proposed frequency-domain approach applies a simple concatenation framework with a BLSTM based speaker encoder. The speaker encoder improves the quality of speaker embedding by capturing the contextual information. In addition, the speaker encoder and speaker extractor are jointly optimized with a magnitude and temporal spectrum approximation loss for temporal continuity.

The frequency-domain speaker extraction always reconstructs the time-domain signal with the phase from the original mixture. Since there is a phase difference between the mixture and the clean single speaker, the usage of the phase from the mixture is not optimal. To address this phase problem, this thesis further proposes a novel end-to-end multi-scale time-domain speaker extraction network (SpEx). SpEx converts the mixture speech into multi-scale embedding coefficients instead of decomposing the speech signal into magnitude and phase spectra. Therefore, SpEx inherently avoids the phase problem. The multi-scale embedding coefficients have the benefits of capturing complementary information with different temporal resolutions. With the multi-scale embedding coefficients, SpEx conducts a multi-scale learning to extract the target speaker's voice. In addition, the multi-task learning framework enables SpEx to be optimized with a multi-scale SI-SDR loss for speaker extraction and a cross-entropy loss for speaker classification. The multi-task learning improves the quality of speaker embedding to well characterize the target speaker. As a result, the perceptual quality of the extracted target speaker's voice is improved.

Experiments were firstly conducted on the WSJ0-2mix-extr and WSJ0-3mix-extr database for speaker extraction with two-speaker and three-speaker mixture, respectively. Experimental results show that the proposed frequency-domain and time-domain methods achieve better performance than the baselines. Meanwhile, the time-domain method (SpEx) achieves 37.1%, 35.6% and 15.0% relative improvements over the proposed frequency-domain approach (SBF-MTSAL-Concat) in terms of SDR, SI-SDR and PESQ. To compare with the state-of-the-art speech separation methods, experiments are further conducted on well-known WSJ0-2mix, WHAM! and WHAMR! database. Experimental results show that SpEx always

achieves better performance than other speech separation methods in clean, noisy, reverberant, and noisy and reverberant two-speaker mixture conditions.

7.1.3 Multi-talker Speaker Verification with Speaker Extraction

Although a reference speech of the target speaker is required, speaker extraction is practical to the applications where only register speakers are need to be responded, for example, speaker verification. The performance of speaker verification degrades significantly when the test speech is an overlapped multi-talker speech. To the best of my knowledge, this thesis has proposed the first solution to solve the speaker verification problem when the test utterance is corrupted by interference speakers most of the time. The multi-talker speaker verification system using speaker extraction as a front-end is called as SE-SV. Three different speaker extraction methods in both time and frequency-domains are exploited to extract the speech of the attended speaker from the mixture. The attended speaker is characterized by the enrollment speech. Then, a traditional speaker verification system makes a decision to decline or accept whether the extracted speech and the enrollment speech are the same speaker.

Experiments were carried out using a simulated two-speaker mixture database with $48,000$ non-target trials and $3,000$ target trials. Comparing with the traditional speaker verification baseline test on *Mixture evaluation set*, the SE-SV with speaker extraction achieves significant improvement. Specifically, the SE-SV with SBF-MTSAL-Concat achieves around $64.4\%, 27.7\%, 18.1\%$ relative reduction over the baseline in terms of EER, DCF08, and DCF10, respectively; If SpEx is applied, $22.8\%, 14.5\%, 8.6\%$ relative reductions are further obtained in terms of EER, DCF08, and DCF10, respectively. Speaker extraction is an efficient technique to solve the multi-talker speaker verification problem.

7.2 Future Work

Speaker extraction is a potential technique to make the real-world applications possible in multi-talker environment. We plan to extend the current works in the

following three research directions.

Noise and reverberation: SpEx has shown significant improvement over its frequency-domain counterpart by avoid the phase estimation problem. SpEx has also shown better performance than other speech separation methods in different mixture condition. Since the real-world environment is complex, the performance of SpEx needs to be studied in a even complex environment, for example, non-overlapped noisy condition. In addition, the speech encoder and speech decoder of SpEx are data-driven, which may cause the performance degradation when the training condition is different from the test environment. To mitigate this situation, SpEx could be improved by forcing each filter in the speech encoder and decoder only responding to specific frequency, for example, the mechanism in SincNet [140]. In addition, the mismatch caused by different training and test conditions needs to be studied by exploiting some adaptation techniques.

Attention based information fusion: Speaker extraction requires the speaker information to be integrated into the extraction pipeline. The SBF-IBM method has proposed a contextual adaptation layer to integrate the speaker information as the weights to each sub-layer in the adaptation layer. Since the number of sub-layers is equal to the dimension of speaker embedding, it may result in a parameter explosion. A simple concatenation scheme is studied in this thesis in Chapter 4 and Chapter 5. Other techniques to fuse two inputs (speaker information and mixture speech) could be further studied, for example, attention based techniques [141].

Multiple channels: As we known, speech processing with multiple channels always achieves better performance than that of single channel, due to the additional spatial information. When the speech direction is estimated with the multi-channel inputs, the speech from that direction could be enhanced and others are suppressed. From our previous study of neural network based beamforming [105] for far-filed speech recognition, the multi-talker speech applications, such as, speech recognition, and speaker verification, could be further improved by exploiting multi-channel speaker extraction. The proposed speaker extraction methods could be extended to multiple channels by applying a minimum variance distortionless response (MVDR) beamformer [105].

Joint optimization of front-end processing and back-end application: To the best of my knowledge, this thesis has proposed the first solution to address the multi-talker speaker verification when the test speech is corrupted by interference speaker most of the time. However, the front-end speaker extraction and back-end speaker verification are independent from each other. A unified neural network framework could further improve the performance of overlapped multi-talker speaker verification by jointly optimizing the speaker extraction and speaker verification.

Bibliography

[1] DeLiang Wang. Deep learning reinvents the hearing aid. *IEEE spectrum*, 54 (3):32–37, 2017. 1, 62

[2] Jacek P Dmochowski, Zicheng Liu, and Philip A Chou. Blind source separation in a distributed microphone meeting environment for improved teleconferencing. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 89–92, 2008. 1

[3] Masahiro Fukui, Toshihiko Watanabe, and Minato Kanazawa. Sound source separation for plural passenger speech recognition in smart mobility system. *IEEE Transactions on Consumer Electronics*, 64(3):399–405, 2018. 1

[4] Shun'ichi Yamamoto, Kazuhiro Nakadai, Hiroshi Tsujino, Toshio Yokoyama, and Hiroshi G Okuno. Improvement of robot audition by interfacing sound source separation and automatic speech recognition with missing feature theory. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1517–1523, 2004. 1

[5] Cherry E. Colin. Some experiments on the recognition of speech, with one and with two ears. *The Journal of The Acoustical Society of America*, 25(5): 975–979, 1953. 2

[6] Stephan Getzmann, Julian Jasny, and Michael Falkenstein. Switching of auditory attention in "cocktail-party" listening: ERP evidence of cueing effects in younger and older adults. *Brain and cognition*, 111:1–12, 2017. 2

[7] Kevin T Hill and Lee M Miller. Auditory attentional control and selection during cocktail party listening. *Cerebral cortex*, 20(3):583–590, 2009. 2, 62

[8] Richard Lyon. A computational model of binaural localization and separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 8, pages 1148–1151, 1983. 2, 16

[9] Ray Meddis and Michael J Hewitt. Virtual pitch and phase sensitivity of a computer model of the auditory periphery. i: Pitch identification. *The Journal of the Acoustical Society of America*, 89(6):2866–2882, 1991. 17

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

[11] Michael L Seltzer, Jasha Droppo, and Alex Acero. A harmonic-model-based front end for robust speech recognition. In *Eighth European Conference on Speech Communication and Technology*, 2003. 17

[12] DeLiang Wang and Guy J Brown. *Computational auditory scene analysis: Principles, algorithms, and applications*. Wiley-IEEE press, 2006.

[13] Guoning Hu and DeLiang Wang. Auditory segmentation based on onset and offset analysis. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(2):396–405, 2007. 2, 16, 17

[14] Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004. 2, 16, 17

[15] Andrzej Cichocki, Rafal Zdunek, and Shun-ichi Amari. New algorithms for non-negative matrix factorization in applications to blind source separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 5, 2006.

[16] Mikkel N Schmidt and Rasmus Kongsgaard Olsson. Single-channel speech separation using sparse non-negative matrix factorization. In *Proceedings of INTERSPEECH*. ISCA, 2006.

[17] Paris Smaragdis. Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs. In *International Conference on Independent Component Analysis and Signal Separation*, pages 494–499. Springer, 2004. 19

[18] Paris Smaragdis. Convolutive speech bases and their application to supervised speech separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(1):1–12, 2007. 19

[19] Tuomas Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3):1066–1074, 2007. 19

[20] R Mitchell Parry and Irfan Essa. Incorporating phase information for source separation via spectrogram factorization. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages II–661, 2007.

[21] Paul D O'grady and Barak A Pearlmutter. Discovering convolutive speech phones using sparseness and non-negativity. In *International Conference on Independent Component Analysis and Signal Separation*, pages 520–527. Springer, 2007. 19

[22] Felix Weninger, Jonathan Le Roux, John R Hershey, and Shinji Watanabe. Discriminative nmf and its application to single-channel source separation. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014. 2, 16, 17, 19

[23] Tuomas Virtanen. Speech recognition using factorial hidden markov models for separation in the feature space. In *Ninth International Conference on Spoken Language Processing*, 2006. 2, 16, 20

[24] Trausti Kristjansson, John Hershey, Peder Olsen, Steven Rennie, and Ramesh Gopinath. Super-human multi-talker speech recognition: The IBM 2006 speech separation challenge system. In *Ninth International Conference on Spoken Language Processing*, 2006. 20, 21

[25] Michael Stark, Michael Wohlmayr, and Franz Pernkopf. Source-filter-based single-channel speech separation using pitch information. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(2):242–255, 2011. 2, 16, 20

[26] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 31–35. IEEE, 2016. 2, 11, 26, 28, 47, 57, 58, 60, 82, 85

[27] Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R Hershey. Single-channel multi-speaker separation using deep clustering. In *Proceedings of INTERSPEECH*, pages 545–549, 2016. 26, 27, 47, 57, 58, 60, 103

[28] Zhong-Qiu Wang, Jonathan Le Roux, and John R Hershey. Alternative objective functions for deep clustering. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 686–690, 2018. 2, 33, 60, 73, 103

[29] Zhuo Chen, Yi Luo, and Nima Mesgarani. Deep attractor network for single-microphone speaker separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 246–250, 2017. 3, 28, 57, 58, 60, 103

[30] Yi Luo, Zhuo Chen, and Nima Mesgarani. Speaker-independent speech separation with deep attractor network. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(4):787–796, 2018. 3

[31] Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 241–245, 2017. 3, 14, 29, 40, 41, 57, 58

[32] Morten Kolbæk, Dong Yu, Zheng-Hua Tan, and Jesper Jensen. Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(10):1901–1913, 2017. 3, 14, 30, 40, 41, 43, 47, 48, 50, 51, 53, 54, 55, 57, 58, 60, 103

[33] Yi Luo and Nima Mesgarani. Tasnet: time-domain audio separation network for real-time, single-channel speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 696–700, 2018. 3, 34, 35, 78

[34] Yi Luo and Nima Mesgarani. Real-time single-channel dereverberation and separation with time-domain audio separation network. In *Proceedings of INTERSPEECH*, pages 342–346, 2018. 35, 78, 103

[35] Yi Luo and Nima Mesgarani. Conv-tasnet: Surpassing ideal time-frequency magnitude masking for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(8):1256–1266, 2019. 3, 34, 35, 70, 73, 78, 79, 103

[36] Keisuke Kinoshita, Lukas Drude, Marc Delcroix, and Tomohiro Nakatani. Listening to each speaker one by one with recurrent selective hearing networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5064–5068, 2018. 3, 60

[37] Naoya Takahashi, Sudarsanam Parthasaarathy, Nabarun Goswami, and Yuki Mitsufuji. Recursive speech separation for unknown number of speakers. In *Proceedings of INTERSPEECH*, pages 1348–1352, 2019. 3

[38] Emine Merve Kaya and Mounya Elhilali. Modelling auditory attention. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 372(1714): 20160101, 2017. 3, 62, 71, 74, 75

[39] Chenglin Xu, Wei Rao, Xiong Xiao, Eng Siong Chng, and Haizhou Li. Single channel speech separation with constrained utterance level permutation invariant training using grid lstm. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6–10, 2018. 7, 39

[40] Chenglin Xu, Wei Rao, Eng Siong Chng, and Haizhou Li. A shifted delta coefficient objective for monaural speech separation using multi-task learning. In *Proceedings of INTERSPEECH*, pages 3479–3483, 2018. 7, 39

[41] Chenglin Xu, Wei Rao, Eng Siong Chng, and Haizhou Li. Optimization of speaker extraction neural network with magnitude and temporal spectrum approximation loss. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6990–6994, 2019. 7, 61, 81

[42] Chenglin Xu, Wei Rao, Eng Siong Chng, and Haizhou Li. Time-domain speaker extraction network. In *IEEE Automaitc Speech Recognition and Understanding Workshop*, 2019. 7

[43] Chenglin Xu, Wei Rao, Eng Siong Chng, and Haizhou Li. Spex: Multi-scale time domain speaker extraction network. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1370–1384, 2020. 7, 61, 81

[44] Wei Rao, Chenglin Xu, Eng Siong Chng, and Haizhou Li. Target speaker extraction for multi-talker speaker verification. In *Proceedings of INTERSPEECH*, pages 1273–1277, 2019. 7, 62, 109

[45] John Garofolo, D Graff, D Paul, and D Pallett. CSR-I (WSJ0) complete LDC93S6A. *Web Download. Philadelphia: Linguistic Data Consortium*, 1993. 11, 82

[46] Gordon Wichern, Joe Antognini, Michael Flynn, Licheng Richard Zhu, Emmett McQuinn, Dwight Crow, Ethan Manilow, and Jonathan Le Roux. WHAM!: Extending speech separation to noisy environments. In *Proceedings of INTERSPEECH*, 2019. 12, 104

[47] Robin Scheibler, Eric Bezzam, and Ivan Dokmanić. Pyroomacoustics: A python package for audio room simulation and array processing algorithms. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 351–355, 2018. 13

[48] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1462–1469, 2006. 14

[49] Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis. Joint optimization of masks and deep recurrent neural networks for monaural source separation. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 23(12):2136–2147, 2015. 14, 26

[50] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 749–752, 2001. 14

[51] Yi Hu and Philipos C Loizou. Evaluation of objective quality measures for speech enhancement. *IEEE Transactions on audio, speech, and language processing*, 16(1):229–238, 2007. 14

[52] Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994. 15

[53] Seungjin Choi, Andrzej Cichocki, Hyung-Min Park, and Soo-Young Lee. Blind source separation and independent component analysis: A review. *Neural Information Processing-Letters and Reviews*, 6(1):1–57, 2005. 16

[54] Pierre Comon and Christian Jutten. *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press, 2010. 16

[55] Wei Lu and Jagath C Rajapakse. Constrained independent component analysis. In *Advances in neural information processing systems*, pages 570–576, 2001. 16

[56] Wei Lu and Jagath C Rajapakse. ICA with reference. *Neurocomputing*, 69 (16-18):2244–2257, 2006. 16

[57] Gil-Jin Jang and Te-Won Lee. A maximum likelihood approach to single-channel source separation. *Journal of Machine Learning Research*, 4(Dec): 1365–1392, 2003. 16

[58] Mike E Davies and Christopher J James. Source separation using single channel ica. *Signal Processing*, 87(8):1819–1832, 2007. 16

[59] Yang Shao, Soundararajan Srinivasan, Zhaozhang Jin, and DeLiang Wang. A computational auditory scene analysis system for speech segregation and robust speech recognition. *Computer Speech & Language*, 24(1):77–93, 2010. 16, 17

[60] Sam T Roweis. One microphone source separation. In *Advances in neural information processing systems*, pages 793–799, 2001. 16, 20

[61] Manuel J Reyes-Gomez, Daniel PW Ellis, and Nebojsa Jojic. Multiband audio modeling for single-channel acoustic source separation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages V–641, 2004. 16, 20

[62] Albert S Bregman. *Auditory scene analysis: The perceptual organization of sound*. MIT press, 1994. 16

[63] Yuxuan Wang, Arun Narayanan, and DeLiang Wang. On training targets for supervised speech separation. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 22(12):1849–1858, 2014. 17, 24, 25, 74

[64] Jen-Tzung Chien. *Source Separation and Machine Learning*. Academic Press, 2018. 18

[65] Zoubin Ghahramani and Michael I Jordan. Factorial hidden markov models. In *Advances in Neural Information Processing Systems*, pages 472–478, 1996. 21

[66] John R Hershey, Steven J Rennie, Peder A Olsen, and Trausti T Kristjansson. Super-human multi-talker speech recognition: A graphical modeling approach. *Computer Speech & Language*, 24(1):45–66, 2010. 21

[67] John R Hershey and Peder A Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages IV–317, 2007. 21

[68] Ozgur Yilmaz and Scott Rickard. Blind separation of speech mixtures via time-frequency masking. *IEEE Transactions on signal processing*, 52(7): 1830–1847, 2004. 24

[69] Mohammad H Radfar and Richard M Dansereau. Single-channel speech separation using soft mask filtering. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(8):2299–2310, 2007.

[70] Aarthi M Reddy and Bhiksha Raj. Soft mask methods for single-channel speaker separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(6):1766–1776, 2007.

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

[72] Yipeng Li and DeLiang Wang. On the optimality of ideal binary time–frequency masks. *Speech Communication*, 51(3):230–239, 2009. 24, 74

[73] Arun Narayanan and DeLiang Wang. Ideal ratio mask estimation using deep neural networks for robust speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7092–7096, 2013. 24, 74

[74] Hakan Erdogan, John R Hershey, Shinji Watanabe, and Jonathan Le Roux. Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 708–712, 2015. 24, 68, 74

[75] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 30

[76] Chenglin Xu, Lei Xie, and Xiong Xiao. A bidirectional lstm approach with word embeddings for sentence boundary detection. *Journal of Signal Processing Systems*, pages 1–13, 2017. 30

[77] David Gunawan and Deep Sen. Iterative phase estimation for the synthesis of separated sources from single-channel mixtures. *IEEE Signal Processing Letters*, 17(5):421–424, 2010. 32, 33, 73

[78] Yi Luo, Zhuo Chen, John R Hershey, Jonathan Le Roux, and Nima Mesgarani. Deep clustering and conventional networks for music separation: Stronger together. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 61–65, 2017. 33, 60

[79] Zhong-Qiu Wang, Jonathan Le Roux, DeLiang Wang, and John R Hershey. End-to-end speech separation with unfolded iterative phase reconstruction. In *Proceedings of INTERSPEECH*, pages 2708–2712, 2018. 33, 73, 103

[80] Szu-Wei Fu, Ting-yao Hu, Yu Tsao, and Xugang Lu. Complex spectrogram enhancement by convolutional neural network with multi-metrics learning. In *IEEE 27th International Workshop on Machine Learning for Signal Processing*, pages 1–6, 2017. 34, 69, 73

[81] Donald S Williamson and DeLiang Wang. Time-frequency masking in the complex domain for speech dereverberation and denoising. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(7):1492–1501, 2017.

[82] Ke Tan and DeLiang Wang. Complex spectral mapping with a convolutional recurrent network for monaural speech enhancement. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6865–6869, 2019. 34, 69, 73

[83] Tara N Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4580–4584, 2015. 34

[84] Tara N Sainath and Bo Li. Modeling time-frequency patterns with lstm vs. convolutional architectures for lvcsr tasks. In *Proceedings of INTERSPEECH*, pages 813–817, 2016. 34, 43, 47, 52

[85] J. Le Roux, S. Wisdom, H. Erdogan, and J. R Hershey. SDR-half-baked or well done? In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 626–630, 2019. 36, 77

[86] M. Delcroix, K. Zmolikova, K. Kinoshita, A. Ogawa, and T. Nakatani. Single channel target speaker extraction and recognition with speaker beam. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5554–5558, 2018. 36, 60, 66, 69, 71, 79, 85, 89, 91, 96, 97, 99

[87] M. Delcroix, K. Kinoshita, C. Yu, A. Ogawa, T. Yoshioka, and T. Nakatani. Context adaptive deep neural networks for fast acoustic model adaptation in noisy conditions. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5270–5274, 2016. 36, 112

[88] Sadaoki Furui. Speaker-independent isolated word recognition using dynamic features of speech spectrum. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(1):52–59, 1986. 41, 42, 69

[89] Xiong Xiao, Shengkui Zhao, Duc Hoang Ha Nguyen, Xionghu Zhong, Douglas L Jones, Eng Siong Chng, and Haizhou Li. Speech dereverberation for enhancement and recognition using dynamic features constrained deep neural networks and feature adaptation. *EURASIP Journal on Advances in Signal Processing*, 2016(1):4, 2016. 41

[90] Bocchieri Bielefeld. Language identification using shifted delta cepstrum. In *Fourteenth Annual Speech Research Symposium*, 1994. 41, 42

[91] Pedro A Torres-Carrasquillo, Elliot Singer, Mary A Kohler, Richard J Greene, Douglas A Reynolds, and John R Deller Jr. Approaches to language identification using gaussian mixture models and shifted delta cepstral features. In *IEEE Workshop on Spoken Language Technology*, 2002. 41

[92] Kshitiz Kumar, Chanwoo Kim, and Richard M Stern. Delta-spectral cepstral coefficients for robust speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4784–4787, 2011. 41

[93] Haipeng Wang, Cheung-Chi Leung, Tan Lee, Bin Ma, and Haizhou Li. Shifted-delta mlp features for spoken language recognition. *IEEE Signal Processing Letters*, 20(1):15–18, 2012. 41

[94] Nal Kalchbrenner, Ivo Danihelka, and Alex Graves. Grid long short-term memory. *arXiv preprint arXiv:1507.01526*, 2015. 43, 47

[95] Shuo-Yiin Chang, Bo Li, Tara N Sainath, Gabor Simko, and Carolina Parada. Endpoint detection using grid long short-term memory networks for streaming speech recognition. In *Proceedings of INTERSPEECH*, pages 3812–3816, 2017. 43, 47

[96] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 48, 86

[97] Jinyu Li, Li Deng, Reinhold Haeb-Umbach, and Yifan Gong. *Robust Automatic Speech Recognition: A Bridge to Practical Applications*. Academic Press, 2015. 49, 62

[98] W. Rao and M. W. Mak. Boosting the performance of i-vector based speaker verification via utterance partitioning. *IEEE Trans. on Audio, Speech and Language Processing*, 21(5):1012 – 1022, 2013. 49

[99] David Talkin. A robust algorithm for pitch tracking (rapt). *Speech coding and synthesis*, 495:518, 1995. 57

[100] Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027, 2016. 58

[101] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009. 58

[102] Andrew RA Conway, Nelson Cowan, and Michael F Bunting. The cocktail party phenomenon revisited: The importance of working memory capacity. *Psychonomic bulletin & review*, 8(2):331–335, 2001. 61

[103] Donna Coch, Lisa D Sanders, and Helen J Neville. An event-related potential study of selective auditory attention in children and adults. *Journal of cognitive neuroscience*, 17(4):605–622, 2005. 61

[104] Shinji Watanabe, Marc Delcroix, Florian Metze, and John R Hershey. *New Era for Robust Speech Recognition: Exploiting Deep Learning*. Springer, 2017. 62

[105] Xiong Xiao, Chenglin Xu, Zhaofeng Zhang, Shengkui Zhao, Sining Sun, Shinji Watanabe, Longbiao Wang, Lei Xie, Douglas L Jones, Eng Siong Chng, and Haizhou Li. A study of learning based beamforming methods for speech recognition. In *CHiME 2016 workshop*, pages 26–31, 2016. 62, 129

[106] Gregory Sell, David Snyder, Alan McCree, Daniel Garcia-Romero, Jesús Villalba, Matthew Maciejewski, Vimal Manohar, Najim Dehak, Daniel Povey, Shinji Watanabe, et al. Diarization is hard: Some experiences and lessons learned for the JHU team in the Inaugural DIHARD Challenge. In *Proceedings of INTERSPEECH*, pages 2808–2812, 2018. 62

[107] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transaction on Audio, Speech, and Language Processing*, 19(4):788–798, 2010. 67, 71, 87, 111

[108] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny. Speaker adaptation of neural network acoustic models using i-vectors. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 55–59, 2013. 67

[109] Andrew Senior and Ignacio Lopez-Moreno. Improving DNN speaker independence with i-vector inputs. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 225–229, 2014.

[110] Penny Karanasou, Yongqiang Wang, Mark JF Gales, and Philip C Woodland. Adaptation of deep neural network acoustic models using factorised i-vectors. In *Proceedings of INTERSPEECH*, pages 2180–2184, 2014. 67

[111] Yi Luo and Nima Mesgarani. Tasnet: Surpassing ideal time-frequency masking for speech separation. *arXiv preprint arXiv:1809.07454v1*, 2018. 70, 73, 78, 79, 88, 103

[112] David Snyder, Pegah Ghahremani, Daniel Povey, Daniel Garcia-Romero, Yishay Carmiel, and Sanjeev Khudanpur. Deep neural network-based speaker embeddings for end-to-end speaker verification. In *IEEE Spoken Language Technology Workshop*, pages 165–170, 2016. 71

[113] Zili Huang, Shuai Wang, and Kai Yu. Angular softmax for short-duration text-independent speaker verification. In *Proceedings of INTERSPEECH*, pages 3623–3627, 2018. 71

[114] Vaninirappuputhenpurayil Gopalan Reju, Soo Ngee Koh, and Yann Soon. Convolution using discrete sine and cosine transforms. *IEEE Signal Processing Letters*, 14(7):445–448, 2007. 73

[115] Tara N Sainath, Ron J Weiss, Andrew Senior, Kevin W Wilson, and Oriol Vinyals. Learning the speech front-end with raw waveform cldnns. In *Proceedings of INTERSPEECH*, 2015. 73

[116] Szu-Wei Fu, Tao-Wei Wang, Yu Tsao, Xugang Lu, and Hisashi Kawai. End-to-end waveform utterance enhancement for direct evaluation metrics optimization by fully convolutional neural networks. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 26(9):1570–1584, 2018. 73

[117] Zhenyao Zhu, Jesse H Engel, and Awni Hannun. Learning multiscale features directly from waveforms. In *Proceedings of INTERSPEECH*, 2016. 73

[118] Doroteo T. Toledano, María Pilar Fernández-Gallego, and Alicia Lozano-Diez. Multi-resolution speech analysis for automatic speech recognition using deep neural networks: Experiments on timit. *PLOS ONE*, 13(10):1–24, 2018. 73, 76, 79, 92

[119] Donald Eric Broadbent. *Perception and communication*. Pergamon Press, 1958. 74

[120] Xiangbin Teng, Xing Tian, and David Poeppel. Testing multi-scale processing in the auditory system. *Scientific Reports*, 6:34390, 2016. 76

[121] Quan Wang, Hannah Muckenhirn, Kevin Wilson, Prashant Sridhar, Zelin Wu, John Hershey, Rif A Saurous, Ron J Weiss, Ye Jia, and Ignacio Lopez Moreno. Voicefilter: Targeted voice separation by speaker-conditioned spectrogram masking. In *Proceedings of INTERSPEECH*, pages 2728–2732, 2019. 79

[122] Kateřina Žmolíková, Marc Delcroix, Keisuke Kinoshita, Takuya Higuchi, Atsunori Ogawa, and Tomohiro Nakatani. Learning speaker representation for neural network based multichannel speaker extraction. In *IEEE Automatic Speech Recognition and Understanding Workshop*, pages 8–15, 2017. 79

[123] Marc Delcroix, Katerina Zmolikova, Tsubasa Ochiai, Keisuke Kinoshita, Shoko Araki, and Tomohiro Nakatani. Compact network for speakerbeam target speaker extraction. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6965–6969, 2019. 79

[124] Bishnu S Atal. Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *the Journal of the Acoustical Society of America*, 55(6):1304–1312, 1974. 87, 119

[125] Anne M Treisman. Selective attention in man. *British medical bulletin*, 1964. 97

[126] Lisa J Stifelman. The cocktail party effect in auditory interfaces: a study of simultaneous presentation. *Retrieved August*, 2:2011, 1994. 102

[127] Chenxing Li, Lei Zhu, Shuang Xu, Peng Gao, and Bo Xu. Cbldnn-based speaker-independent speech separation via generative adversarial training. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 711–715, 2018. 103

[128] Matthew Maciejewski, Gordon Wichern, Emmett McQuinn, and Jonathan Le Roux. Whamr!: Noisy and reverberant single-channel speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 696–700, 2020. 105, 106

[129] Xavier Anguera, Simon Bozonnet, Nicholas Evans, Corinne Fredouille, Gerald Friedland, and Oriol Vinyals. Speaker diarization: A review of recent research. *IEEE Transactions on Audio, Speech, and Language Processing*, 20 (2):356–370, 2012. 110

[130] Oleg Kudashev, Sergey Novoselov, Konstantin Simonchik, and Alexander Kozlov. A speaker recognition system for the SITW challenge. In *Proceedings of INTERSPEECH*, pages 833–837, San Francisco, California, Sep. 2016.

[131] Yi Liu, Yao Tian, Liang He, and Jia Liu. Investigating various diarization algorithms for speaker in the wild (SITW) speaker recognition challenge. In *Proceedings of INTERSPEECH*, pages 853–857, San Francisco, California, Sep. 2016.

[132] Ondrej Novotnỳ, Pavel Matejka, Oldrich Plchot, Ondrej Glembek, Lukás Burget, and Jan Cernockỳ. Analysis of speaker recognition systems in realistic scenarios of the sitw 2016 challenge. In *Proceedings of INTERSPEECH*, pages 828–832, 2016.

[133] Houman Ghaemmaghami, Md Hafizur Rahman, Ivan Himawan, David Dean, Ahilan Kanagasundaram, Sridha Sridharan, and Clinton Fookes. Speakers in the wild (SITW): The QUT speaker recognition system. In *Proceedings of INTERSPEECH*, pages 838–842, San Francisco, California, Sep. 2016.

[134] David Snyder, Daniel Garcia-Romero, Gregory Sell, Alan McCree, Daniel Povey, and Sanjeev Khudanpur. Speaker recognition for multi-speaker conversations using x-vectors. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, Brighton, UK, May 2019. 110

[135] Delphine Charlet, Claude Barras, and Jean-Sylvain Liénard. Impact of overlapping speech detection on speaker diarization for broadcast news and debates. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7707–7711, 2013. 110

[136] Sree Harsha Yella and Hervé Bourlard. Overlapping speech detection using long-term conversational features for speaker diarization in meeting room conversations. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(12):1688–1700, 2014. 110

[137] Patrick Kenny. Bayesian speaker verification with heavy-tailed priors. In *Proceedings of Odyssey: Speaker and Language Recognition Workshop*, volume 14, 2010. 111

[138] Simon JD Prince and James H Elder. Probabilistic linear discriminant analysis for inferences about identity. In *IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.

[139] Daniel Garcia-Romero and Carol Y Espy-Wilson. Analysis of i-vector length normalization in speaker recognition systems. In *Proceedings of INTERSPEECH*, pages 249–252, 2011. 111

[140] Mirco Ravanelli and Yoshua Bengio. Speaker recognition from raw waveform with sincnet. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 1021–1028, 2018. 129

[141] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 129