Speech Enhancement Using Spectrogram Feature Fusion for Noise Robust Speech Recognition

Hao Shi

Graduate School of Informatics
Kyoto University

Abstract

Automatic speech recognition (ASR) under noisy scenarios is challenging. *Speech enhancement* (SE) is an effective front-end technology for reducing the impact of noise on ASR. Compared with the SE methods based on conventional signal processing, deep learning methods have shown more effective performance. Based on the powerful structure, appropriate learning targets and input-output features are designed for training supervised SE systems. Many SE methods introduce useful information into the network by diverse learning targets and input-output features. However, few studies explored methods that fully utilize the complementary information between different representations within single input speech. This thesis study focuses on effectively extracting complementary representations within single speech audio and incorporating them into neural networks to improve SE and ASR performance. Specifically, the complementary information between multiple learning targets, multiple bandwidth input features, waveform-spectrogram hybrid domain features, and pretrained-finetuned features are investigated for improving the robustness of SE and ASR.

Mapping and masking are two primary learning targets in frequency-domain supervised SE. Although there is complementarity between them, one of them has been conventionally used as the front-end. In Chapter 3, we first analyze how they complement each other. To improve the human hearing experience, we propose subband-based spectrogram fusion (SBSF), which combines the spectrogram of low-frequency and high-frequency estimated by different SE models to utilize complementarity. Experimental results on the Voice-Bank Demand (VB-D) dataset show that enhancing different sub-bands with different learning targets improves the human hearing experience. To improve the robustness of ASR, we propose a spectrogram fusion (SF)–based end-to-end (E2E) robust ASR system, in which the mapping-based and masking-based SE methods are used as the front-end simultaneously. Experimental results on the simulated noisy Aishell-1 dataset show that the proposed method improves ASR, especially under the low signal-to-noise ratio.

Noise severely affects the speech structure. With the degraded spectrogram, the neural network may fail to capture the detailed speech component. In Chapter 4, we first propose the multi-masked spectrogram to allow the network to detect speech boundary information. These feature maps make the speech boundary information obvious. Experimental results on the VB-D dataset show that the proposed method is effective for spectral detail recovery and enhances SE performance with a proper decomposition number. The spectrogram can be divided into wideband and narrowband with different frame lengths. Although they have different spectral characteristics, SE systems conventionally utilize a single bandwidth spectrogram. We propose an SE system that simultaneously utilizes multiple bandwidth information, which is fused in the encoder. Experimental results on the VB-D dataset show

that larger bandwidth differences provide better auxiliary information, and using multiple bandwidth input features improves the human hearing experience.

While waveform-domain SE has been extensively investigated in recent years and achieves state-of-the-art performance in many datasets, spectrogram-based SE tends to show robust and stable enhancement behavior. In Chapter 5, we propose a waveform-spectrogram hybrid method (WaveSpecEnc) to improve the robustness of waveform-domain SE. WaveSpecEnc refines the corresponding temporal feature map by spectrogram encoding in each encoder layer. Incorporating spectral information improves the human hearing experience. Furthermore, we improve it for robust ASR by further utilizing spectrogram encoding information (WaveSpecEnc+) to both SE front-end and ASR back-end. Experimental results using the CHiME-4 dataset show that the proposed method consistently improves ASR performance in real evaluation sets, outperforming other methods, such as DEMUCS and Conv-Tasnet. Refining in the shallow encoder layers is very effective, and the effect is confirmed even with a large ASR model using WavLM.

Adapting an ASR system to unseen noise environments is crucial. In Chapter 6, we first thoroughly investigate adapter-based ASR adaptation. Self-supervised learning (SSL)-based pretrained models have significantly improved ASR performance. As the feature extraction (FE) module is also well-trained with a large amount of training data, freezing the FE during finetuning for downstream ASR tasks is common. However, when there is a severe mismatch between the simulated noisy data for pretraining and real noisy data, finetuning the FE with the real noisy data should be done without losing the pretrained information. We propose a dual-path FE adaptation to address this problem. It combines the frozen pretrained FE and finetuned-adapted FE paths with convolutional fusion layers. Moreover, adapters are inserted into the Transformer encoder. Experimental results on the CHiME–4 dataset show that there are some complementarities between the pretrained and finetuned FE paths. Finetuning the proposed FE with adapters in the encoder is more effective for adapting to new noises.

In Chapter 7, we compare all proposed methods in this thesis. The diffusion model, which is recently proposed as a probabilistic method, is also chosen for comparison. Specifically, we compare the frequency-domain and waveform-domain methods; we also compare the deterministic and probabilistic methods. We conducted experiments using the VB-D dataset. The complementary information from both input features and output learning targets improves the SE performance. On the other hand, the waveform-domain methods perform much better than the frequency-domain methods that only process the magnitude of the spectrogram. They show comparable performance to the method processing complex spectrograms. The waveform-spectrogram hybrid method proposed in Chapter 5 outperforms all other methods. Moreover, it is found that with the same deep neural network structure, the diffusion and deterministic models show comparable performance in improving the human hearing experience.

Chapter 8 concludes the thesis and a brief look at future work.

Acknowledgements

This dissertation was accomplished at the Speech and Audio Processing Laboratory, Graduate School of Informatics, Kyoto University. I want to express my profound appreciation to the people who helped me in many ways.

My most sincere thanks go first to my supervisor, Prof. Tatsuya Kawahara. Due to the COVID-19 pandemic, I could not come to Japan for a long time after enrolling. Despite this, during our long-term remote communication, he offered me precious advice and ample support, instilling confidence to complete my Ph.D. journey. Initially, I was introverted and sometimes unsure how to share my thoughts. With his encouragement, I gradually became more outgoing and courageous in expressing my ideas. Additionally, I am deeply grateful for his dedication and for repeatedly sacrificing much of his valuable time to revise my papers for submission. Furthermore, he has given me many valuable suggestions and assistance in my daily life, helping me adapt to and fall in love with life in Kyoto. His earnest attitude towards scientific research has profoundly influenced me, inspiring me to strive to work on the front line of research for the rest of my life.

I also express my special thanks and appreciation to Prof. Longbiao Wang and Prof. Jianwu Dang for the opportunities and help they gave me from the beginning of my master career. During the COVID-19 pandemic, they kindly provided me with an excellent research environment at Tianjin University.

I also express my special thanks and appreciation to Kazuki Shimada-san, Shusuke Takahashi-san, and Yuki Mitsufuji-san for my first internship at Sony. I thank them for allowing me to study in such a wonderful group. I also appreciate the help from Takashi Shibuya-san and Zhi Zhong-san during my internship at Sony. And I thank them for their valuable comments on completing the experiments and their continued help after my internship. I also express my special thanks and appreciation to Naoyuki Kamo-san, Tomohiro Nakatani-san, Shoko Araki-san, and Marc Delcroix-san for holding my second internship at NTT. Due to the COVID-19 pandemic, all my welcome and farewell parties before my internship were online. That was my first time attending a party offline. I thank them for all the help during and after my internship and for continuing to give me valuable comments after my internship to write the paper.

Furthermore, I express my special thanks and appreciation to the members of my dissertation committee: Prof. Ko Nishino and Prof. Yuichi Nakamura for their time in carefully checking this dissertation and providing advice during my presentation.

I also thank the members of the Speech and Audio Processing Lab. I want to thank Yuan Gao and Yahui Fu for the help they have given me in my daily life and for the wonderful time we spent together. I am grateful for the comments and support from Dr. Masato Mimura and Dr. Shinsuke Sakai. I am also grateful for the comments from Prof. Kazuyoshi Yoshii. I appreciate the help from Assistant Professor Koji Inoue. I want to thank Ms. Mayumi Abe for helping me handle many issues. I also appreciate the help from other members of the lab.

Finally, I want to extend my deepest gratitude to my parents. Their unconditional love and support have been the cornerstone of my journey. They have always encouraged me to follow my dreams and have been my steadfast support at every step of my growth. Their understanding and patience have empowered me to explore and realize my ambitions in my own way.

Table of Contents

List of Figures **xi**

List of Tables **xv**

1 Introduction **1**
1.1 Background 1
1.2 Task Formulation 2
1.2.1 Automatic Speech Recognition (ASR) 2
1.2.2 Speech Enhancement (SE) 2
1.3 Problems of Interests 3
1.3.1 Speech Distortion and Loss Caused by SE 3
1.3.2 Waveform-domain SE is Unstable 3
1.3.3 ASR Back-end is Not Robust to New Noise Conditions 4
1.4 Approaches 4
1.4.1 Fusion of Spectrogram Features with Different Learning Targets for SE 5
1.4.2 Fusion of Multi-masked and Multi-resolution Spectrogram for SE . 5
1.4.3 Fusion of Spectrogram Features to Waveform-domain SE 6
1.4.4 Fusion of Different Feature Extraction Modules for Adapter-based ASR 6
1.5 Thesis Outline 7

2 Preliminaries **9**
2.1 Speech Enhancement (SE) 9
2.1.1 Single-channel SE Methods 10
2.1.2 Multi-channel SE Methods 14
2.2 End-to-end (E2E) Automatic Speech Recognition (ASR) 15
2.2.1 Connectionist temporal classification (CTC) 16
2.2.2 RNN Transducer 16

2.2.3 Attention-based Encoder-Decoder Model . . . 17
2.2.4 Encoder Architecture . . . 17
2.3 Self-supervised Learning for ASR . . . 18
2.3.1 Wav2Vec 2.0 . . . 19
2.3.2 HuBERT . . . 19
2.3.3 WavLM . . . 19
2.4 Adapter . . . 19

3 Fusion of Spectrogram Features with Different Learning Targets for SE 21
3.1 Introduction . . . 21
3.2 Analysis of Complementarity Between Mapping and Masking . . . 22
3.2.1 Analysis on Complementarity . . . 22
3.2.2 Analysis on Dynamic Ranges of mapping-based System . . . 24
3.3 Spectrograms Fusion (SF) . . . 25
3.3.1 Subband-based Spectrogram Fusion (SBSF) . . . 25
3.3.2 Minimum Difference Masks-based SF (MDMs-SF) for Noise Robust ASR . . . 28
3.4 Experiments . . . 29
3.4.1 Datasets . . . 29
3.4.2 Model Settings . . . 30
3.4.3 Evaluation Metrics . . . 32
3.4.4 Results and Analysis of SBSF . . . 33
3.4.5 Results and Analysis of MDMs-SF E2E ASR . . . 37
3.5 Conclusion . . . 37

4 Fusion of Multi-masked and Multi-resolution Spectrogram for SE 39
4.1 Introduction . . . 39
4.2 Multi-masked and Multi-resolution Spectrograms Fusion . . . 41
4.2.1 Multi-masked Spectrograms Fusion (MM-SF) . . . 41
4.2.2 Multi-resolution Spectrograms Fusion . . . 42
4.3 Experiments . . . 46
4.3.1 Feature Extraction . . . 46
4.3.2 Baselines . . . 46
4.3.3 Neural Network Structure . . . 47
4.3.4 Results of MM-SF . . . 48
4.3.5 Results of Multi-resolution Input System . . . 50
4.4 Conclusion . . . 53

5 Fusion of Spectrogram Features into Waveform-domain SE 55
5.1 Introduction . . . 55
5.2 Waveform-spectrogram Hybrid System . . . 56
5.2.1 WaveSpecEnc SE Front-end . . . 56
5.2.2 WaveSpecEnc+ for Robust ASR . . . 58
5.3 Experiments . . . 60
5.3.1 Evaluations of Pre-trained Speech Enhancement Front-end . . . 60
5.3.2 Evaluations of SE-based Adaptation for Noise-mismatched ASR Back-end . . . 65
5.3.3 Evaluations of Finetuning Both SE Front-end and ASR Back-end . 70
5.4 Conclusion . . . 74

6 Fusion of Different Feature Extraction Modules for Adapter-based ASR 75
6.1 Introduction . . . 75
6.2 Exploration of Adapter for Noise Robust ASR . . . 77
6.3 Dual-path Adaptation for Feature Extraction Module . . . 78
6.4 Experiments . . . 80
6.4.1 Dataset . . . 80
6.4.2 Experimental Settings . . . 81
6.4.3 Results and Analysis of Adapter Exploration . . . 83
6.4.4 Results and Analysis of Feature Extraction Adaptation . . . 88
6.4.5 Evaluations with noisy speech-trained HuBERT . . . 91
6.4.6 Evaluations with noisy speech-trained WavLM . . . 91
6.5 Conclusion . . . 92

7 Comparision of Different SE Methods 93

8 Conclusions 97
8.1 Contributions . . . 97
8.2 Future Work . . . 98

References 101

List of Publications 113

List of Figures

1.1 Thesis Outline 8

2.1 Structure of CRN 12
2.2 Structure of adapter. 20

3.1 Square loss ratio of mapping (Eq. (2.6)) and masking (Eq. (2.7)) in different frequencies (0 – 8,000 Hz), which are calculated with Eq. (3.1): the lower, the better. 23
3.2 Square loss of mapping & masking in different frequencies (0 – 8,000 Hz) on Voice Bank training set. 24
3.3 The flowchart of subband-based optimization: the subband enhanced spectrogram will be used to replace the corresponding information of full-band enhanced spectrogram. 26
3.4 Overview of robust ASR systems. 27
3.5 Square loss ratio (the lower, the better) of DM_F $\rightarrow$ DM_H, DM_L + DM_H, and DM_L + SA_H at different frequencies (257-dimensional linear spectrogram) on Voice Bank training set, which were calculated with Eq. (3.1). . . 31
3.6 Performance of evaluation measures (PESQ, SIG, OVRL, BAK) of different enhancement systems in SNRs (-5, 0, 5, 10, and 15 db) conditions: -5 db was an unseen condition, and the noisy conditions were unseen. The horizontal axis represents SNRs, and the vertical axis represents the value of the evaluation metric. 34

4.1 Spectrogram examples extracted with different window lengths: (a) 32ms narrowband spectrogram; (b) 16ms wideband spectrogram; (c) 8ms wideband spectrogram. 40
4.2 (a) Flowchart of proposed multi-masked spectrograms fusion (MM-SF). (b) one example of multi-masked spectrograms extraction. 41
4.3 The flowchart of the proposed multi-resolution spectrograms fusion systems. 43

4.4 Structure of Linear Block. 44
4.5 16ms and 8ms features aligned with 32ms features for framing 44
4.6 Diagram of the frame concatenation. 45
4.7 Samples of decomposed spectrograms. 47
4.8 Measures on different decomposition numbers: Red line represents the "MM-SF"; Blue line represents the "CRN-stack"; Black line represents the "CRN". 48
4.9 Spectrograms of different enhancement systems: (a) clean speech; (b) CRN enhanced speech; (c) CRN-stack enhanced speech; (4) Decomposition enhanced speech. 49
4.10 Selected feature maps of baseline (CRN) and proposed decomposition method. 50
4.11 Spectrograms of different SE systems. 53

5.1 (a) Encoder layer structure of the proposed waveform-spectrogram Hybrid system (WaveSpecEnc); (b) feature dimensions processed by different blocks in the proposed encoder layer: $\boldsymbol{y_t}$ represents the waveform-domain input from the previous WaveSpecEnc encoder layer or original waveform; $\boldsymbol{y_f}$ represents the spectral inputs from the previous frequency block or the magnitude of the spectrogram; $\widetilde{\boldsymbol{y_f}}$ represent the spectral output (to the frequency block in the next encoder layer); $\widetilde{\boldsymbol{y_t}}$ represents the final output (to the next encoder or LSTM layer). 57
5.2 Flowchart of different robust ASR systems: (a) ASR system with WaveSpecEnc front-end; (b) WaveSpecEnc+-based ASR system. 59
5.3 (Seen) Relatively improvement of SIG / OVRL / BAK values (↑) compared with non-enhanced signals (Table 5.1) in real development and evaluation sets. All noise conditions are SEEN to the model. 61
5.4 (Unseen) Relatively improvement of SIG / OVRL / BAK values (↑) compared with non-enhanced signals (Table 5.1) in real development and evaluation sets. The test noise conditions are UNSEEN to the model. 61
5.5 (Seen) dMOS values (↑) in simulated and real sets. All noise conditions are SEEN to the model. 62
5.6 (Unseen) dMOS values (↑) in simulated and real sets. The test noise conditions are UNSEEN to the model. 62
5.7 Enhanced magnitude spectrograms of the pre-trained SE front-end. The clip is a real noisy speech under PED noise condition: (a) Noisy, (b) Bi-LSTM enhanced, (c) DEMUCS enhanced, and (d) WaveSpecEnc enhanced. 64

5.8 Enhanced magnitude spectrograms of SE front-end after joint training. The clip is a real noisy speech under PED noise condition: (a) Noisy, (b) Bi-LSTM enhanced, (c) DEMUCS enhanced, (d) WaveSpecEnc enhanced, (e) WaveSpecEnc+ enhanced. 69

6.1 Insert position of adapter for ASR back-end. 78
6.2 Neural network structure of (a) baseline feature extraction module; (b) proposed dual-path adaptation of feature extraction module (Dual-FE-Conv). . 79
6.3 Flowchart of (a) adapter-based adaptation for Transformer encoder; (b) adapter-based adaptation for both FE module and Transformer encoder. . . 80

List of Tables

3.1 Performance of Different SE Systems on Voice-bank Test Set. 32
3.2 Performance of Different SE Systems on Reverb Chanllenge 2014 far room test set. 32
3.3 Performance of Different SE Systems on Reverb Chanllenge 2014 near room test set. 33
3.4 Performance of Different SE Systems on Different Noisy Conditions (Unseen, synthesized, clean speech from Voice Bank dataset, noisy from non-speech 100). 33
3.5 Performance of SE in the test set 1. 35
3.6 Performance of SE in the test set 2. 35
3.7 CER (%) of different E2E systems with test set 1: the SNRs of the test set are known; the noise of the test set is unknown. 36
3.8 CER (%) of different E2E systems with test set 2: both the SNRs and the noise of the test set are unknown. 36

4.1 Results of different enhancement systems. 47
4.2 Results of different enhancement systems: 8ms (16ms, 32ms) feat. represents that 8ms (16ms, 32ms) feature as input and output feature; 8ms (16ms) aux. represents that the auxiliary feature is 8ms (16ms); 8, 16ms aux. represents that the auxiliary features are both 8ms and 16ms spectrograms. 51
4.3 Input dimension (32ms, 16ms, 8ms) of Linear Block in different encoder layers: we use the output dimension of Conv Block (n) $\times$ the number of framing m. 51

5.1 (Unprocessed noisy data) Evalution metrics on real development and evaluation sets. 60

5.2 (Seen) Word Error Rate (%, ↓) in real development and evaluation sets. All noise conditions are seen to the model. FT represents whether the front-end has been finetuned. The back-end is not finetuned in this experiment. 66
5.3 (Unseen) Word Error Rate (%, ↓) in real development and evaluation sets. The test noise conditions are unseen to the model. Compared with the seen results in Table 5.2, the relative decrease percentage of WER under the unseen testing (Decrease). FT represents whether the front-end has been finetuned. The back-end is not finetuned in this experiment. 67
5.4 Number of Conformer Layers with Spectrogram Encoding (Num.). All noise conditions are seen to the model. 68
5.5 Word Error Rate (%, ↓) in real development and evaluation sets. All noise conditions are seen to the model. LM denotes whether to use the external language model. 71
5.6 Word Error Rate (%, ↓) in real development and evaluation sets. All noise conditions are seen to the model. WavLM is adopted as the acoustic model. 72
5.7 Comparison between different single channel automatic speech recognition systems (Word Error Rate, %, ↓). 73

6.1 Performance of baseline pretrained ASR model. 83
6.2 Effect of placing the adapter into different encoder layers on development sets (**trained using the entire CHiME–4 training dataset**). 84
6.3 Effect of placing the adapter into different encoder layers on evaluation sets (**trained using the entire CHiME–4 training dataset**). 84
6.4 Effect of different embedding dimensions in adapter (**trained using the entire CHiME–4 training dataset**). 85
6.5 Effect of different training sets for adapter-based adaptation on development sets. "**Held**" represents whether the specific noise conditions were excluded during model training; when utilizing the held-out approach, both the training and testing sets utilize a single noise type condition. "**Real**" represents whether the real noisy data is used during the training process. "**Simu.**" represents whether the simulated noisy data is used during training. "**Utt.**" represents how many utterances (channels 1 to 6 of the same utterance are considered single utterances) are used during the training process. ♣ represents the number of all utterances in the corresponding noisy condition (this is due to the slightly different amounts of simulated data for the four noise conditions). ★ represents that 100 sentences are selected from four noise conditions to constitute a training set. 86

6.6 Effect of different training sets for adapter-based adaptation on evaluation sets. "**Held**" represents whether the specific noise conditions were excluded during model training; when utilizing the held-out approach, both the training and testing sets utilize a single noise type condition. "**Real**" represents whether the real noisy data is used during the training process. "**Simu.**" represents whether the simulated noisy data is used during training. "**Utt.**" represents how many utterances (channels 1 to 6 of the same utterance are considered single utterances) are used during the training process. ♣ represents the number of all utterances in the corresponding noisy condition (this is due to the slightly different amounts of simulated data for the four noise conditions). ★ represents that 100 sentences are selected from four noise conditions to constitute a training set. 87
6.7 Effect of adapter for different SE-based robust ASR systems (**trained using the entire CHiME–4 training dataset**). 88
6.8 Evaluation with HuBERT finetuned with LibriSpeech–960 on development sets: "FE" represents the feature extraction module; "Enc" representes the Transformer encoder; "FT" means finetuning all parameters; "Ada" means the use of adapters. 89
6.9 Evaluation with HuBERT finetuned with LibriSpeech–960 on evaluation sets: "FE" represents the feature extraction module; "Enc" representes the Transformer encoder; "FT" means finetuning all parameters; "Ada" means the use of adapters. 89
6.10 Evaluation with HuBERT finetuned with LibriSpeech–960 and MUSAN noises on development sets. 90
6.11 Evaluation with HuBERT finetuned with LibriSpeech–960 and MUSAN noises on evaluation sets. 90
6.12 Evaluation with WavLM finetuned with LibriSpeech–960 and MUSAN noises on development sets. 91
6.13 Evaluation with WavLM finetuned with LibriSpeech–960 and MUSAN noises on evaluation sets. 91

7.1 Comparison of different proposed systems. "Dete." represents the deterministic methods; "Prob." represents the probabilistic methods. 94

Chapter 1

Introduction

1.1 Background

Intelligent speech applications such as smartphone assistants, smart speakers, and car navigation systems have brought convenience to human beings. However, there is often noise in the application scenarios, which is also received when the microphone picks up the speech signal, significantly affecting speech application performance. Due to randomness of noise, even the same noise source affects the speech signal differently according to the different scenarios and microphone distances. Compared with stationary noise, non-stationary noise is more challenging to process. In addition, the combination of different noise sources also brings the uncertainties.

Automatic speech recognition (ASR) is an essential part of human-computer interaction. The quality of ASR directly affects the subsequent quality of interaction. Due to the influence of noise, ASR in noisy scenarios often cannot achieve satisfactory performance. Research on noise robust ASR has been conducted for decades. In addition to collecting and simulating noisy speech with various noise conditions, decoupling noise from noisy speech is also achieved with the speech enhancement (SE) front-end. Although SE can restore most of the speech signal, it also leads to speech distortion and loss, which are fatal for ASR. It is necessary for SE to reduce the drawbacks mentioned above. With the development of deep learning, many neural network structures are designed and constantly improve SE performance. Since neural network structures have flexibility, appropriate features and learning targets are also crucial. They may have complementarity, which has not been exposed in previous studies. Besides, speech has different feature representations: frequency-domain spectrogram and time-domain waveform.

1.2 Task Formulation

Here, we formulate the main tasks addressed in this thesis: automatic speech recognition (ASR) and speech enhancement (SE).

1.2.1 Automatic Speech Recognition (ASR)

Automatic speech recognition (ASR) [1] aims to transcribe the linguistic content of the input speech signal. The ASR have been studied for several decades [1, 2]. It was initially studied based on the pattern matching theory. However, it is gradually replaced by statistical models due to its performance and robustness. The Hidden Markov Model (HMM) was the most widely used by combining with Gaussian Mixture Models (GMM) as the acoustic model in traditional ASR. HMMs were used to model the temporal variability, while GMMs handled the acoustic variations. Despite HMM-GMM systems showing strong acoustic modeling ability, it is limited by the piece-wise combination of simple Gaussian distributions. A significant advancement in ASR came with the integration of Deep Neural Networks (DNN) with HMMs. The DNN-HMM model uses DNN for acoustic modeling, significantly improving the system's capabilities, especially its ability to handle diverse speech patterns.

However, the DNN-HMM model needs to be combined with many components, such as the language model and pronunciation dictionary. Some of these modules require expert knowledge and are optimized independently. To replace the complex system, the end-to-end (E2E) ASR system integrates multiple modules into one model to achieve E2E recognition. Connectionist Temporal Classification (CTC), Attention-based Encoder-Decoder (AED), and Recurrent neural network Transducer (RNN-T) are the most popular techniques for E2E ASR. Although E2E ASR has many advantages and has achieved state-of-the-art performance on many benchmark datasets, some drawbacks still exist. The main issue is that it requires extensive and diverse training data. Moreover, it is also unstable to unfamiliar speech patterns. With the introduction of self-supervised learning (SSL), the performance of E2E ASR has been further improved. Currently, strategies based on SSL pre-training using massive unlabeled data and supervised fine-tuning using in-domain data have shown effectiveness.

1.2.2 Speech Enhancement (SE)

Speech enhancement (SE) aims to improve the quality and intelligibility of speech signals. It is useful in numerous applications, such as hearing aids. Moreover, it is also used as the front-end module for ASR. The traditional SE methods, such as Spectral Subtraction and Wiener

Filtering, are based on mathematical signal processing. There is a trade-offs between noise reduction and speech signal quality. Some methods effectively process stationary noises but often fail to process non-stationary noises. Besides, music noise is often introduced with the processing of the traditional SE methods. With the development of deep learning, supervised SE methods achieve better performance than traditional SE methods, thus attracting more attention. With powerful nonlinear modeling capabilities of deep neural networks, the supervised SE methods often contain few or no mathematical assumptions.

Supervised SE has gone through several stages of development. The model structure of the early supervised SE methods was simple. Phase information in speech signals is difficult to predict, thus only the signal's magnitude part is estimated. Then, more complex network structures have been proposed to process phase information as well. Estimating the real and imaginary parts of the Short-Time Fourier Transform or directly the waveform are two mainstream methods involving phase processing. Even though these models have already strong modeling capabilities and the performance is often better than the magnitude-only methods, some works still point out that they need to be more robust against the randomness of phase information. In addition, SE training strategies oriented for ASR are also essential.

1.3 Problems of Interests

1.3.1 Speech Distortion and Loss Caused by SE

SE can improve the quality of speech signals from noisy speech signals. However, SE also brings signal distortion and loss, which significantly impact, especially in applications where speech intelligibility and quality are critical. For example, in telecommunications, distortion can cause listener misunderstanding or fatigue. In hearing aids, it can significantly impact the wearer's ability to understand speech, especially in noisy environments. It has much larger impact on SE systems for downstream speech applications than human hearing experiences (normal listeners) because the human brain has strong signal-fixing abilities. Thus, the distortion causes ASR errors and system performance degradation. Hence, developing and implementing SE with the minimization of distortions is critical. It is needed to balance the trade-off between noise reduction and preserving natural speech quality.

1.3.2 Waveform-domain SE is Unstable

Deep learning-based SE [3, 4] can be classified into frequency-domain [3, 4] and waveform-domain models [5, 6]. Frequency-domain SE extracts frequency features from the waveform-domain speech signals. The short-time Fourier transform (STFT) is the most commonly

used. The magnitude of spectrogram [4] is a common frequency-domain feature, but it ignores the phase information and limits the model performance. To address the problem, real and imaginary parts of STFT, also called the complex-domain feature [7], which contains both magnitude and phase information, have been adopted by many SE systems in recent years [8, 9]. Different from frequency-domain SE, waveform-domain SE [5, 6, 10] adopts speech waveform as input and output features. The magnitude and phase information is included in the waveform. With intensive studies, waveform-domain SE systems achieve state-of-the-art performance in many datasets [5, 6, 11]. However, it is often pointed out that the frequency-domain SE systems have more stable enhancement performance than waveform-domain SE systems [12] because of the instability of the phase information [13].

1.3.3 ASR Back-end is Not Robust to New Noise Conditions

While ASR systems have achieved significant advancements, the robustness is still challenging, especially in adapting to new unseen noise conditions. The noises may mask critical speech features or introduce distortions the ASR model has not learned to handle. For example, background chatter in a crowded place is not present in the training data of the ASR system. This limitation is critical for ASR systems to handle changing and dynamic environments. The core reason is primarily caused by the design and training of ASR systems. Most ASR systems are trained on large speech corpus. Although some noise augmentation techniques are involved, they often fail to cover the data distribution of the real speech signal. The lack of robustness to new noise conditions significantly affects the practical deployment of ASR systems. For example, the inaccurately recognized commands in a noisy environment in a speech assistant system degrades the user experience. Therefore, the robustness is critical for improving user experience and broader acceptance of ASR technology.

1.4 Approaches

This thesis study explore complementary representations within single speech audio and incorporates them into neural networks to improve SE and ASR performance. Moreover, the ASR back-end adaptation with limited data is also studied.

1.4.1 Fusion of Spectrogram Features with Different Learning Targets for SE

We reduce speech distortion and loss by fusing complementary enhanced spectral features. Mapping and masking are two widely used learning targets for frequency-domain SE. Although some studies found that these two learning targets have some complementarities, few studies analyzed their characteristics.

First, we investigate the complementarity between these two learning targets based on their performance at each frequency bin. The mapping-based and masking-based SE systems perform well in the low-frequency and high-frequency parts, respectively. Based on the observation, we propose sub-band spectrogram fusion (SBSF) to combine the complementary enhanced sub-bands. We combine the spectrogram of low-frequency and high-frequency bands, which are estimated by different learning targets. We also combine the full-band SE and sub-band SE modules.

Furthermore, we also propose a spectrogram fusion (SF)-based E2E robust ASR system, in which the mapping-based and masking-based SE are used as the front-end simultaneously. We adopt SF to combine the advantages of mapping-based and masking-based SE systems. SF and ASR modules are connected in an E2E manner, and joint training is conducted to finetune the front-end and the back-end. The fusion of two SEs is beneficial for ASR, especially under low signal-to-noise ratios.

1.4.2 Fusion of Multi-masked and Multi-resolution Spectrogram for SE

We reduce the speech distortion and loss by fusing compelmentary input spectral features. The spectrogram can be divided into wideband and narrowband according to the framing length. We extract different spectral representations from single-resolution spectrogram and multi-resolution spectrograms.

We first propose the multi-masked spectrograms fusion (MM-SF) to highlight the speech components from the single-resolution spectrogram. We extract the strong speech part and ignore others to make the speech component boundary obvious. The speech feature maps are determined based on the output mask by a trained masking-based SE system. Stacking feature maps of strong speech components enables the input features to provide sufficient speech boundary information. The proposed MM-SF method is effective for spectral detail recovery.

Although narrowband and wideband spectrograms have different spectral characteristics, SE systems conventionally utilize single narrow bandwidth spectrograms. We propose an SE system that simultaneously utilizes multiple bandwidth spectral information, more specifi-

cally, augments the wider bandwidth spectrograms as auxiliary information. Spectrograms of different bandwidths are processed separately by multiple convolution blocks and fused in the encoder. The spectrogram, which differs more from the main enhancement spectrogram, provides better auxiliary information.

1.4.3 Fusion of Spectrogram Features to Waveform-domain SE

In order to improve the robustness of the waveform-domain SE system, we propose a waveform-spectrogram hybrid system (WaveSpecEnc). The proposed system complements waveform-domain DEMUCS [5] with the magnitude of spectrogram information. The waveform-spectrogram information fusion is done in the encoder. In each encoder layer, temporal and spectral information is first extracted by convolution processing at the utterance level. Then, the temporal feature maps are segmented and aligned with the spectral feature maps. The aligned spectral information is used to refine the segmental temporal information. The artificial noise is alleviated by introducing spectrogram-domain information.

Furthermore, we improve the WaveSpecEnc (WaveSpecEnc+) by augmenting the encoding information of the ASR back-end with spectral information extracted in the SE module. The enhanced spectral feature maps in the last layer in the WaveSpecEnc encoder are used to supplement the filter-bank (FBank) encoding in the ASR back-end. We aim to extract discriminative information from the enhanced spectral feature maps, which helps improve filterbank encoding performance in the ASR encoder. Compared to WaveSpecEnc, which only integrates spectrogram encoding information into the front-end's temporal information, WaveSpecEnc+ integrates spectrogram encoding information into both the front-end and the ASR back-end to enhance the performance of ASR.

1.4.4 Fusion of Different Feature Extraction Modules for Adapter-based ASR

Compared with the SE front-end, the ASR back-end has much more parameters, and thus the necessary amount of the training data for the ASR back-end is usually far larger than that of the SE front-end. Moreover, in many practical applications, it is often not allowed to finetune the ASR back-end but only possible to tune the SE front-end. We first investigate an effective adaptation way to finetune the SE front-end only by freezing the ASR back-end when encountering a new noise scene. ASR performance can be improved by finetuning the SE front-end by propagating the ASR loss [14].

Despite the widespread application of adapters across various ASR tasks, limited investigation has been conducted towards noise-robust ASR. We explore adapter-based noise-robust

ASR, considering a range of viewpoints. Our primary focus is investigating the adapter insertion points, the data employed for training, and the synergy with SE front-end models. We also use the adapters to adapt SSL pre-trained ASR. Furthermore, we propose a dual-path adaptation of the feature extraction (FE) module of SSL models to address the mismatch between the pretraining with simulated noisy data and the evaluation on real noisy data. The proposed method utilizes the complementarity between the pre-trained and the finetuned FE module paths.

1.5 Thesis Outline

The organization of this thesis is as follows. Chapter 2 gives a brief review of the SE and ASR. Chapter 3 addresses fusing of spectrogram features with different learning targets. Chapter 4 addresses fusing of multi-masked and multi-resolution spectrogram. Chapter 5 improving robustness of the waveform-domain SE system and adapting the ASR back-end with SE-based adaptation. Chapter 6 addresses adaptation of ASR back-end with adapter-based adaptation and adaptation of the FE module within the SSL pre-trained model. Chapter 7 compares all proposed SE methods in this thesis. Chapter 8 concludes this thesis and gives future directions in SE and adaptation for the ASR back-end. Figure 1.1 depicts the organization.

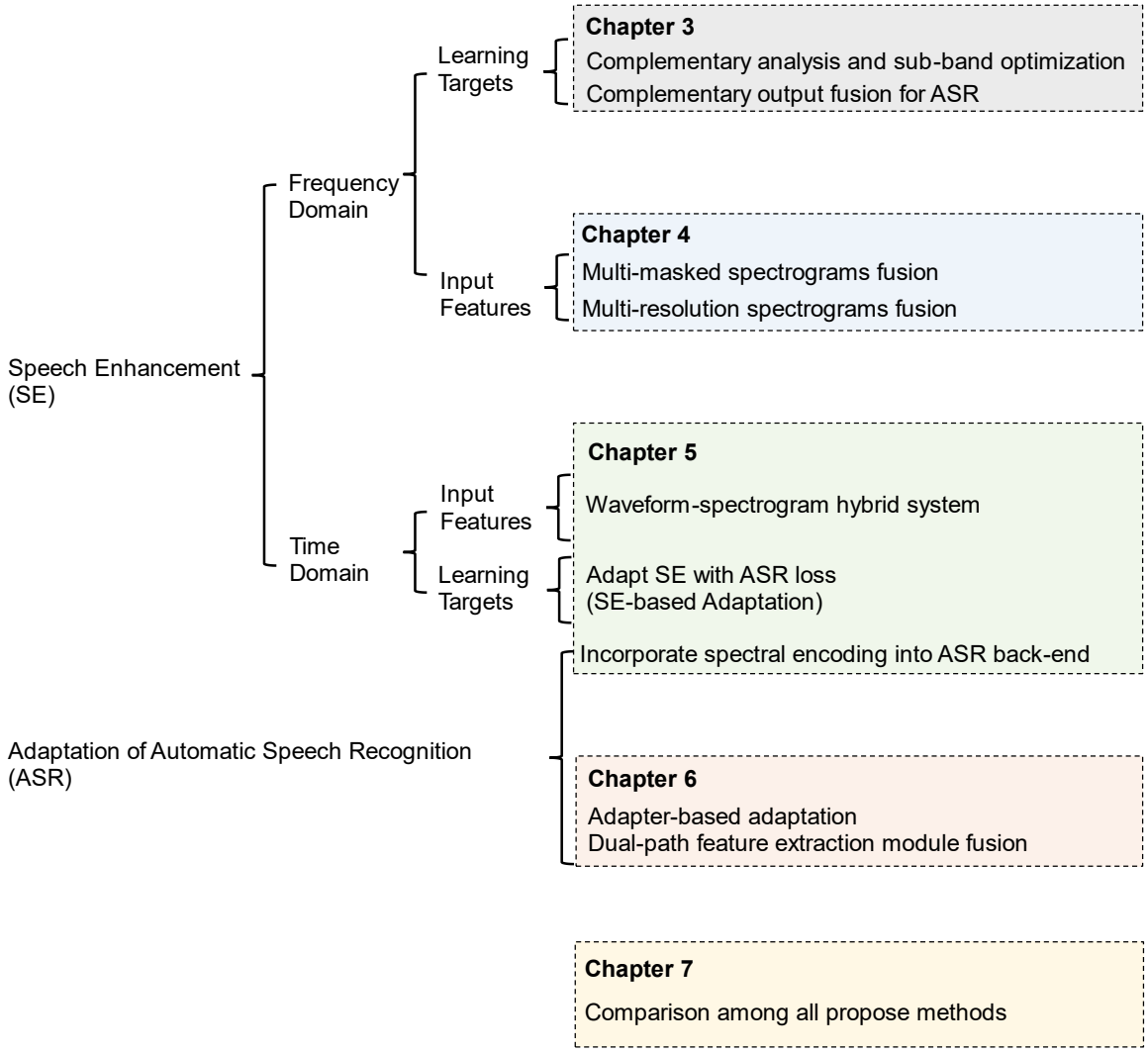


Fig. 1.1 Thesis Outline

Chapter 2

Preliminaries

This chapter presents a literature review of speech enhancement (SE) and automatic speech recognition (ASR). Section 2.1 introduces SE methods. Section 2.2 introduces end-to-end ASR systems. Section 2.3 introduces self-supervised learning-based speech processing methods.

2.1 Speech Enhancement (SE)

A noisy speech signal y can be expressed as:

$$y = x * r + n \tag{2.1}$$

where x represents a clean speech signal in the waveform domain, $*$ represents the convolution process, r represents a room impulse response, and n represents an additive noise. The SE aims to recover x (output) from y (input). Waveform-domain features x and y have their corresponding frequency-domain representations. The spectrogram is a common frequency-domain feature. It is extracted via the Short-Time Fourier Transform (STFT):

$$Y = |\text{STFT}(y)|, \quad X = |\text{STFT}(x)| \tag{2.2}$$

$\text{STFT}(\cdot)$ denotes the Short-Time Fourier Transform. $|\cdot|$ denotes the modulus. Y and X represent the magnitude spectrogram of the noisy and clean speech. The corresponding frequency-domain representation of the equation (2.1) can be expressed as:

$$Y = X \odot R + N \tag{2.3}$$

where R and N represent the magnitude of room impluse ratio and noise. $\odot$ denotes point-wise matrix multiplication.

2.1.1 Single-channel SE Methods

Traditional Methods

Spectral Subtraction (SS) is one of the earliest and simplest techniques to reduce noise in noisy speech. It operates in the frequency domain, where the magnitude spectrogram of the noisy speech is by subtracting an estimate of the noise spectrogram N. It assumes that noise is stationary and can be estimated from non-speech segments. As a result, it is prone to artifacts such as "musical noise" in when noise estimation is inaccurate or when dealing with non-stationary noise.

Cepstral Mean Normalization (CMN) is mainly used for channel normalization and reducing the impact of environmental distortions, including mild reverberation. Reverberation occurs when sound waves reflect off surfaces and combine with the direct path sound. It results in a smeared or blurred spectral representation, making enhancing speech from its reflections challenging. Eqn. (2.3) can be rewritten by taking the logarithm and assuming $N = 0$:

$$\log(Y) = \log(X) + \log(R) \tag{2.4}$$

Then, each term is converted into ceptral coefficients. CMN effectively minimizes the long-term ceptrum variations by estimating the mean of ceptral coefficients corresponding $\log(R)$ and subtracting it. In speech dereverberation, CMN is often used as a preprocessing step. While CMN can reduce the impact of mild to moderate reverberation, it may not be sufficient for severe reverberation beyond the analysis time frame. In such cases, CMN is typically combined with other dereverberation techniques to achieve better results.

Wiener filtering is also a widely-used frequency-domain SE method to minimize the impact of noise by leveraging the statistical properties of both the signal and noise. It estimates the clean speech signal by applying a frequency-dependent filter derived from the ratio of the power spectral densities of the clean speech $\widetilde{S}$ and noise $\widetilde{N}$. Specifically, the Wiener filter H:

$$H = \frac{\widetilde{S}^2}{\widetilde{S}^2 + \widetilde{N}^2}, \quad \widetilde{X} = H \odot Y. \tag{2.5}$$

The enhanced speech signal $\widetilde{X}$ is obtained by filtering the noisy speech signal Y with H. Wiener filtering is simple but effectively reduces noise while preserving the essential characteristics of the speech.

Supervised Methods

Traditional SE models have mathematical assumptions and often fail when dealing with non-stationary noise. Supervised SE [15–17] based on deep neural networks allows the model to fit any mathematical model without or with few mathematical assumptions. Therefore, supervised SE methods receive more and more attention and achieve better performance than traditional methods. According to the input and output features, SE can be classified into frequency-domain and waveform-domain methods.

The **frequency-domain SE** adopts the frequency-domain feature as the input and the output features.

Mapping [18, 19, 13] directly enhanced spectrogram by the strong nonlinear capability of neural networks [13]. The loss function for directly mapping (DM) is as follows:

$$\mathscr{L}_{DM}(X, X_{\mathrm{DM}}) = \frac{1}{TF} \sum_{t,f=1}^{T,F} (\widetilde{X}(t,f) - X(t,f))^2, \tag{2.6}$$

where t and f represent the time frame and frequency bin, respectively. X is the ground-truth. T represents the total number of frames in a speech sample. F represents the total frequency bins. $\widetilde{X}$ is the output magnitude spectrogram.

The masking approach is proposed according to computational auditory scene analysis [20]. The earliest ideal binary mask [21] was designed to classify T–F bins of speech and non-speech signals, and the ideal ratio mask [22] indicates which T-F bins are dominated by speech. Masking-based SE uses deep neural networks to obtain a mask $\widetilde{M}$ between speech and noise. This mask is applied to the observed noisy signal to extract the speech signal. The loss function of SA is as follows:

$$\mathscr{L}_{SA}(X, X_{\mathrm{SA}}) = \frac{1}{TF} \sum_{t,f=1}^{T,F} (\widetilde{M}(t,f) \odot Y(t,f) - X(t,f))^2, \tag{2.7}$$

where $\widetilde{M}$ is the estimated mask, and $\odot$ denotes point-wise matrix multiplication.

The SE performance with different network structures varies greatly. The deep autoencoder [19] and deep neural network (DNN) [13], convolutional neural network [23], and recurrent neural network [24] are examples of early network structures for SE. Moreover, the combination of different types of networks [25] and some complex structures [26–28]—for example, the U-NET structure [29] and the generative adversarial network [30]—have powerful performance. The following models are one of the extended methods and used in later Chapters.

Convolutional Recurrent Neural Network [10] (CRN, shown in Fig. 2.1) performs well in frequency-domain SE as a baseline system. It contains an encoder, LSTM layers, and a decoder:

$$\widetilde{M} = \boldsymbol{D}(\boldsymbol{L}(\boldsymbol{E}(Y))) \tag{2.8}$$

$\boldsymbol{E}$ is the encoder of CRN, which contains several Conv Blocks. $\boldsymbol{L}$ contains several LSTM layers. $\widetilde{M}$ is the output of decoder. $\boldsymbol{D}$ is the decoder of CRN, which contains several DeConv

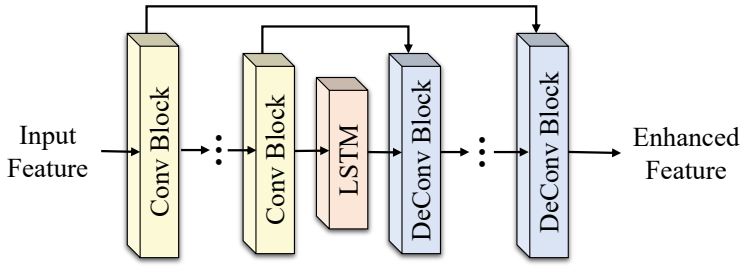


Fig. 2.1 Structure of CRN

Blocks.

The masking-based CRN system is as following:

$$\widetilde{X} = \widetilde{M} \odot Y \tag{2.9}$$

where $\widetilde{X}$ is the final enhanced spectrogram. When training the network, we use the SA [31, 32]. The loss function of training is the same as Eq. (2.7).

The **two stage model** contains two SE modules. Enhanced spectrogram $\widetilde{X}$ can be obtained from the first stage. Then, another SE module is employed to get the final enhanced feature:

$$\hat{X} = \mathscr{N}(\widetilde{X}), \tag{2.10}$$

or

$$\hat{X} = \mathscr{N}(\mathbf{Y}, \widetilde{X}), \tag{2.11}$$

where $|Y|$ is the noisy spectrogram. $\hat{X}$ is the second stage enhanced spectrogram. $\mathscr{N}$ represents the neural network of SE model. The enhanced spectrogram $\widetilde{X}$ can be directly input to the second stage with Eq.(2.10). Another way is to input the enhanced and noisy spectrograms to the neural network simultaneously with Eq. (2.11), which ensures that the noise spectrogram compensates for the information lost in the enhanced spectrogram. Both of them get the final enhanced spectrogram by Eq. (2.9).

For **waveform-domain SE**, the waveform y and x are the input and output feature of the neural network. The mean absolute error (MAE) between the enhanced waveform $\widetilde{x}$ and the original signal x is the common loss function for training waveform-domain SE:

$$\mathscr{L}_{Waveform-mae} = \frac{1}{T}\sum_{t=1}^{T} |\widetilde{x}(t) - x(t)|, \tag{2.12}$$

where T is the time points in the waveform. x is the ground-truth.

DEMUCS [5] is a powerful waveform-domain SE system. It is based on the U-Net structure. It contains an encoder, a decoder, and two long short-term memory (LSTM) [33] layers between them. Each **"Time Block" (DEMUCS encoder layer)** contains two "Conv_1d" layers:

$$GLU(Conv1d((ReLU(Conv1d(\cdot))))). \tag{2.13}$$

The first "Conv_1d" layer is followed by the "ReLU" activation function, while the second "Conv_1d" layer is followed by the "GLU" activation function. Each **"DEMUCS Decoder Layer"** contains one "Conv_1d" layer and one "DeConv_1d" layer:

$$ReLU(ConvTranspose1d(GLU(Conv1d(\cdot))))). \tag{2.14}$$

The "Conv_1d" layer is followed by the "GLU" activation function, while the "DeConv_1d" layer is followed by the "ReLU" activation function. The kernel size of the encoder and decoder layers is 8.

DEMUCS [5] also adopts upsampling [34] and downsampling [34] processing to the original input and enhanced output waveform. For the loss function, it adopts waveform-domain MAE loss (Eq. (2.12)) with multi-resolution frequency loss (Eq. (2.16)).

$$\mathscr{L}_{demucs} = \alpha \mathscr{L}_{Waveform-mae} + (1-\alpha)\sum_{r=1}^{R}(\mathscr{L}_{stft}(r)) \tag{2.15}$$

$$\mathscr{L}_{stft} = \frac{1}{TF}\sum_{t,f=1}^{T,F}(\frac{||STFT(\widetilde{x})| - |STFT(x)||}{|STFT(x)|} + |\log|STFT(\widetilde{x})| - \log|STFT(x)||). \tag{2.16}$$

where α is the hyperparameter for combining these two training losses. R represents the multi-resolution number, and r represents the specific resolution among {32ms, 64ms, 128ms} used in STFT.

Data mismatch is a big issue of supervised SE [35]. Conventionally, supervised SE systems are trained with simulated training data. Using real noisy data for training is difficult because a clean speech waveform is needed for ground truth. However, the data distribution between real and simulated noisy speech often differs significantly. Moreover, the noise conditions are also crucial. SE systems tend to degrade in the presence of unseen noise. It is necessary to evaluate the robustness of SE systems under real data and unseen noise conditions.

2.1.2 Multi-channel SE Methods

Traditional Methods

Multi-channel SE systems uses several microphones input $y_1, y_2, \ldots, y_n$. Thus, n observations are obtained. Non-blind condition and blind condition are two major approaches to microphone array processing, differing in whether or not the system has prior knowledge of the environment. Beamforming and multiple signal classification are two main classes of the non-blind condition method, which requires the direction of arrival (DOA) information. The spatial property is determined by an steering vector.

The beamformer aims to extract signals of a particular direction from observations. It requires the steering vectors for each microphone. According to the steering vectors, the beamformer design a filter that only passes the signal from a particular direction. Various methods, such as Delay-and-Sum and Minimum Variance Distortionless Response (MVDR), have been proposed for filter estimation. The Multiple signal classification (MUSIC) is an efficient method for estimating and distinguishing multiple signal sources coming from different directions. It constructs the covariance matrix of the received signal and performing eigenvalue decomposition to separate the signal subspace and noise subspace. Utilizing the analysis of subspaces, the MUSIC constructs a spatial spectrum function whose peak position corresponds to the blind direction of the signal source.

Independent component/vector analysis (ICA / IVA), multi-channel nonnegative matrix factorization (NMF), and nonlinear time-frequency masking are common methods for the blind condition.

ICA is based on the assumption of statistical independence of signal sources and uses mathematical methods to separate the observed mixed signals into independent components. As an extension of ICA, IVA considers the interdependence between different signals, making signal separation in reverberant environments more effective. NMF decomposes a non-negative data matrix into the product of two or more non-negative matrices. In multi-channel signal processing, NMF can separate and identify different sound sources in mixed audio signals. Its non-negative nature makes it ideal for processing natural signals. Time-frequency masking is a signal processing method based on the time-frequency domain used to extract or enhance target signals from mixed signals. It effectively suppresses undesired signal components while preserving or enhancing target components by masking the time-frequency representation. This method is particularly effective when processing signals with complex time-frequency characteristics, such as speech or music signals. Nonlinear time-frequency masking is widely used in speech enhancement, noise suppression, and

preprocessing before speech recognition, especially when the environment is noisy or the signal quality is low.

Supervised Methods

Multi-channel traditional methods can be extended to multi-channel supervised methods.

Deep Learning Beamforming [36, 31] leverages deep neural networks to optimize beamforming techniques for multi-channel speech signals. Learning from the microphone array's spatial characteristics enhances the target speech while suppressing noise from other directions. Deep neural networks, such as CNNs and LSTM, are used to design and refine the beamforming filters, significantly improving the clarity of the desired speech signal and overall noise reduction compared to traditional methods.

Deep ICA [37] utilizes deep learning to perform independent component analysis on multi-channel speech signals. It employs deep neural networks, such as autoencoders or generative adversarial networks (GANs), to model and separate the mixed sources into their original, independent components. By learning complex statistical properties of the noisy signals, it effectively separates and extracts the target speech from noise and other interfering sources, enhancing speech clarity in complex environments.

Supervised multi-channel processing can be done end-to-end by leveraging the powerful ability of deep networks. End-to-end SE involves using deep learning models to directly map noisy multi-channel input signals to enhanced speech outputs. This method simplifies the multi-channel processing by training a single deep neural network to handle feature extraction and enhancement in one integrated model. Optimizing the entire enhancement process end-to-end improves overall performance and reduces the error propagation in traditional pipeline methods. However, they are prone to overfit the training condition and dataset.

2.2 End-to-end (E2E) Automatic Speech Recognition (ASR)

With the development of deep learning, end-to-end (E2E) models have been investigated. The acoustic and language models are integrated in the E2E automatic speech recognition (ASR) models. Connectionist temporal classification (CTC), Recurrent Neural Network (RNN) Transducer, and Attention-based Encoder-Decoder models are the three most widely-used E2E methods.

2.2.1 Connectionist temporal classification (CTC)

Connectionist Temporal Classification (CTC) enables the model to align variable-length input speech sequences without predefined segmentation. This capability simplifies the training process. CTC requires the length of output labels to be smaller than the input sequence length, and introduces "blank" labels to make the output sequences have the same length as the input speech sequence. The architecture has only an encoder module and an additional linear layer with softmax to compute the CTC loss. The encoder converts the acoustic features into high-level representations. During training, CTC calculates the probabilities of all possible alignments and optimizes the model to maximize the probability of the ground-truth output. CTC runs real-time during decoding. We denote the input sequence as x and the output label sequence as l. The probability of l given x is denoted as $P(l|x)$. This probability is computed by summing over all possible alignments π, which can be mapped to l:

$$P(l|x) = \sum_{\pi \in \mathscr{A}(l)} P(\pi|x), \tag{2.17}$$

where $\mathscr{A}(l)$ represents the set of all valid alignments of l. The CTC loss is computed with the negative log-likelihood of the correct label sequence:

$$\mathscr{L}_{CTC} = -\log P(l|x). \tag{2.18}$$

CTC assumes that frames are independent. With this assumption, $P(\pi|x)$ can be decomposed into the product of the frames posterior:

$$P(\pi|x) = \prod_{t=1}^{T} P(\pi_t|x) \tag{2.19}$$

where T is the length of the speech sequence.

2.2.2 RNN Transducer

RNN Transducer (RNN-T) extends CTC by conditioning the output on the previously estimated tokens and the speech sequence until the current time step $P(l_u|x_{1:t}, l_{1:u-1})$. In this manner, the RNN-T solves the conditional independence assumption of CTC. Furthermore, it maintains the natural streaming capability.

RNN-T consists of an encoder module, a prediction module, and a joint module. The function of the encoder module is the same as the CTC-based E2E ASR, extracts the high-level representations h_t^{enc} from the input acoustic features. The prediction module estimates the high-level representation h_u^{pre} from the previous estimated token y_{u-1}. The joint module is a feed-forward layer that combines the h_t^{enc} and h_u^{pre}:

$$z_{t,u} = \theta(Qh_t^{enc} + Vh_u^{pre} + b_z), \tag{2.20}$$

where Q and V are weight matrices, b_z is the bias vector, and the θ is the non-linear function. The output layer precesses $z_{t,u}$:

$$h_{t,u} = W_y z_{t,u} + b_y, \tag{2.21}$$

where W_y denotes a weight matrix, and b_y denotes a bias vector. The probability of each output token k is:

$$P(y_u = k|x_{1:t}, y_{1:u-1}) = \text{softmax}(h^k_{t,u}). \tag{2.22}$$

The loss function of RNN-T is also to minimize $-\log P(l|x)$:

$$P(l|x) = \sum_{\pi \in \mathscr{A}(l)} P(\pi|x). \tag{2.23}$$

where $A(l)$ represents the set of all valid alignments of l.

2.2.3 Attention-based Encoder-Decoder Model

The Attention-based Encoder-Decoder (AED) model is another E2E ASR model. AED consists of an encoder module, an attention module, and a decoder module. The encoder module has the same function as the encoder in CTC and RNN-T models. The attention module computes the attention weights based on the decoder's current state and the encoder's output sequence. It determines the most relevant parts of the input sequence at each decoding step to compute the context vector. The decoder takes its previous estimated labels and the context vector to estimate the next label. This process is repeated at each decoding step, with the recalculation of the attention weights. The loss function of AED ($\mathscr{L}_{Attention}$) is to the maximum likelihood of $logP(l|x)$. In order to encourage monotonic alignment between the input speech features and the output label sequence, the AED model is often optimized together with the CTC loss in a multi-task manner:

$$\mathscr{L}_{Hybrid} = \beta \mathscr{L}_{Attention} + (1-\beta)\mathscr{L}_{CTC}, \tag{2.24}$$

where β is the hyperparameter. Although the attention-based encoder-decoder model performs better than CTC-based and RNN Transducer models, it is unsuitable for streaming.

2.2.4 Encoder Architecture

The encoder is vital for E2E ASR. It transforms the input speech features into high-level feature representations.

Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) was the most widely-used encoder for the early stage of E2E ASR. The Bi-directional LSTM-based encoder has stronger modeling capabilities than

the simple LSTM-based encoder because it uses the entire utterance information to generate the output. However, it cannot be used for straming ASR.

Transformer

Transformer-based [38–40] encoder introduces the self-attention mechanism to capture long-term information dependencies. The self-attention is used to compute the attention distribution of the input speech feature sequence with the dot-product similarity:

$$\begin{aligned} \alpha_{t,\tau} &= \frac{\exp(\beta(\mathbf{W}_q x_t)^T(\mathbf{W}_k x_\tau))}{\sum_{\tau'}\exp(\beta(\mathbf{W}_q x_t)^T(\mathbf{W}_k x_{\tau'}))} \\ &= softmax(\beta \mathbf{q_t^T k}_\tau), \end{aligned} \tag{2.25}$$

where the $\mathbf{W}_k$, $\mathbf{W}_q$, and $\mathbf{W}_v$ are linear transformation matrices of key k, query q, and value v. $\beta = \frac{1}{\sqrt{d}}$ is the scaling factor, and d is the dimension of the feature vector for attention head. Then, the attention weights combine the value vector:

$$\mathrm{z_t} = \sum_\tau \alpha_{t\tau}\mathbf{W_v}x_\tau = \sum_\tau \alpha_{t\tau} v_\tau. \tag{2.26}$$

Furthermore, the Multi-Head Self-Attention (MHSA) adopts multiple self-attentions on the input feature sequence and concatenates the multiple outputs.

Conformer

The Conformer enhances the Transformer with Convolutional Neural Networks (CNNs) [41, 42], enabling more efficient capture of local contexts. As a result, the Conformer has advantages in handling both the context-level and local feature-level information. Furthermore, it uses relative positional encodings instead of absolute ones, leading to better generalization and nuanced sequence-order processing. Besides, the Conformer has a macaron-style network structure consisting of a half-step feed-forward layer, followed by a convolution layer, and another half-step feed-forward layer, which makes the model's capacity to process sequential data more effectively.

2.3 Self-supervised Learning for ASR

With introduction of self-supervised learning (SSL), E2E ASR has significantly been improved. SSL [43, 44] enables the model to learn representations from massive unlabeled data.

2.3.1 Wav2Vec 2.0

Wav2Vec 2.0 [45] employs the self-supervised learning framework. It contains a CNN-based feature extraction (FE) module, a quantization module, and a contextualized Transformer. The training process of the Wav2Vec 2.0 follows the pretraining and the finetuning. During the pretraining stage, the model randomly masks portions of quantized features and tries to predict these masked parts with the unmasked parts. This pre-trained approach is similar to BERT's masked language model training with contrastive loss. During the finetuning stage, the quantization module will be removed, and the FE and Transformer modules will be finetuned using labeled data with the CTC loss or other objectives [46].

2.3.2 HuBERT

HuBERT [47] is also a self-supervised model designed for ASR. It also contains the FE module and Transformer-based encoder. HuBERT uses a clustering algorithm for the features extracted by the FE module to create pseudo labels; HuBERT then learns to predict these pseudo labels from masked segments of the extracted features. The audio feature is re-clustered using the representations learned in the first stage to create a new set of pseudo labels. The model is then re-trained on this refined labeling. This iterative process can be repeated several times. Once pretrained, the model can be finetuned on a labeled ASR dataset with the CTC loss or AED models [46].

2.3.3 WavLM

WavLM [48] contains the FE module and Transformer-based encoder. Different from the Wav2Vec 2.0 and HuBERT, the gated attention mechanism is adopted in WavLM. For pretraining, WavLM follows HuBERT's pretraining process and uses the pseudo labels created by the feature clustering. The most significant difference between HuBERT and WavLM is that WavLM uses the simulated noisy data during pretraining. As a result, WavLM has more robust performance than HuBERT, especially in noisy conditions.

2.4 Adapter

An adapter was initially proposed [49] to enhance transfer learning in natural language processing tasks, particularly when faced with limited task-specific data. Its strength is task customization without significantly altering the model's structure. This versatility makes it suitable for various contexts, including accent recognition, emotion detection [50, 51], and

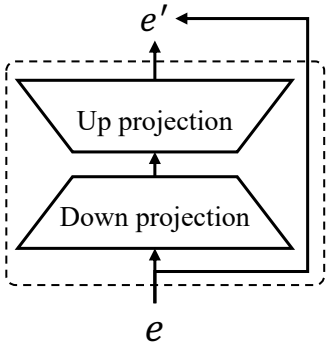


Fig. 2.2 Structure of adapter.

noise-robust ASR. The design of an adapter depends on task requirements and the model architecture [52]. The simple yet effective neural network structure of an adapter consists of two dense layers. It takes input from a chosen layer in the pretrained model's output. The primary function of the first dense layer is to capture the initial transformations of the input features from the pretrained model. The second layer builds on the transformed features from the first dense layer and captures intricate task-specific patterns and representations. The adaptation process is depicted as follows:

$$e' = e + \text{adapter}(e) \tag{2.27}$$

Here, e represents the selected layer's output, and e' represents the adapted feature. The structure of the adapter is shown in Fig. 2.2. We employ the LoRA [53] structure where the central dimension is smaller than both the input and output dimensions.

Chapter 3

Fusion of Spectrogram Features with Different Learning Targets for SE

3.1 Introduction

While frequency-domain SE systems can be improved in many ways, two types of learning targets are widely used: masking and mapping [31]. Masking targets [21, 22, 54, 55] describe the time—frequency relationships of clean speech to background interference, whereas mapping targets [18, 19, 13] correspond to the spectral representations of clean speech [31]. The motivation for mapping targets is that the features can be estimated directly by the strong nonlinear capability of neural networks [13]. The masking targets are proposed in accordance with computational auditory scene analysis [20]. The earliest ideal binary mask [21] was designed to classify T–F bins of speech signals and non-speech signals, and the ideal ratio mask [22] indicates which T-F bins are dominated by speech. It is found that the two types of learning targets have some complementarities [56, 24]. However, few relevant studies analyzed their characteristics, and explored using their complementarity to get better SE performance.

In this work, we address the aforementioned issues. We use direct mapping (DM) [13] and signal approximation (SA) [57, 58] as mapping and masking targets, respectively. First, we investigate the complementarity between these two learning targets based on their performance at each frequency bin. We find that the mapping-based and masking-based SE systems tend to perform well in the low-frequency and high-frequency parts, respectively. In addition, the recovery of the mapping-based SE system at high and low frequencies is very different, while the recovery of the masking-based SE system at each frequency is more stable.

On the basis of their complementarity, we investigate methods that combine them. Specifically, we propose subband-based spectrogram fusion (SBSF). First, we combine the spectrogram of low-frequency and high-frequency bands, which are estimated by different methods. Next, we combine the full-band SE and subband SE models. The subband-enhanced spectrogram is used to replace the corresponding subset information in the full-band-enhanced spectrogram. The major difference between the proposed method and previous works [59–61] is that our method divides the full-band spectrogram into subbands from the perspective of the complementarity between the mapping-based and masking-based SE systems. In this work, we divide the full-band spectrogram into low and high subbands considering the loss and endeavor to apply different SE models. The reason for the poor performance of mapping at high frequencies is that the loss is mainly concentrated in the low-frequency part during network training. Thus, the subband optimization is used to optimize the poorly predicted part of the full-band spectrogram. Furthermore, we investigate the effective combination of different learning targets.

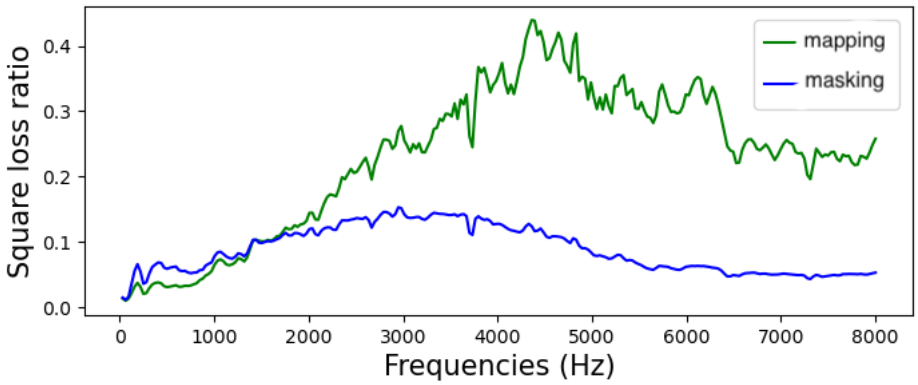
Some studies have shown the complementarities [62, 24] between the mapping-based and masking-based SE systems, but only one of them is still used as a front-end system in ASR [63, 64]. To investigate whether the complementary between the mapping and masking systems is beneficial for ASR, we propose a minimum difference masks-based spectrograms fusion [65, 4] (MDMs-SF) E2E robust ASR system. The mapping-based and masking-based SE are combined for the front-end. They are connected to ASR in an E2E manner. Joint training [66–68] is adopted to finetune the front-end and back-end.

3.2 Analysis of Complementarity Between Mapping and Masking

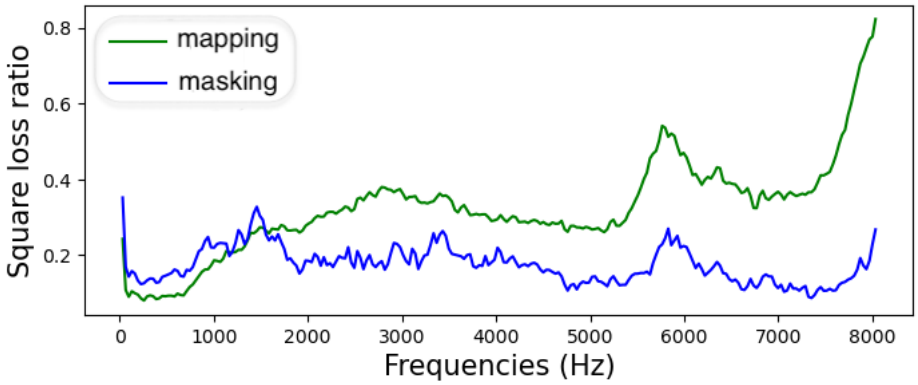
3.2.1 Analysis on Complementarity

We investigate the complementarity of mapping and masking approaches by measuring the SE performance on different frequency bins. To measure the recovery performance of different frequencies between the enhancement signal and unprocessed noisy signal, we compare the loss at different frequencies between the enhanced and unprocessed features. We define the square loss ratio as follows:

$$\text{Ratio}_F = \frac{\mathscr{L}_{square}}{\mathscr{L}_{original}} = \frac{\sum_{t=1}^{T} (|\widetilde{X}(t)| - |X(t)|)^2}{\sum_{t=1}^{T} (|Y(t)| - |X(t)|)^2}, \tag{3.1}$$



(a) The results on Voice Bank training set



(b) The results on REVERB challenge training set

Fig. 3.1 Square loss ratio of mapping (Eq. (2.6)) and masking (Eq. (2.7)) in different frequencies (0 – 8,000 Hz), which are calculated with Eq. (3.1): the lower, the better.

where $|\widetilde{X}|$ and $|Y|$ are the enhanced and noisy input speech magnitude spectrogram, respectively. We only sum $\mathscr{L}_{square}$ and $\mathscr{L}_{original}$ along the time axis, so a $(1,F)$ dimensional vector can be obtained according to Eq. (3.1). The square loss ratio shows the recovery of the enhanced spectrogram at different frequencies compared to the input (noisy) spectrogram.

We use all training set (Voice Bank of 10k utterances and REVERB Challenge of 8k utterances, which will be described in Section 3.4.1) to compute the square loss ratio. The DM system was trained with Eq. (2.6), and the SA system was trained with Eq. (2.7). We used a 257-dimensional spectrum as input and output.

Fig. 3.1 shows the square loss ratio compared to the input noisy spectrogram. We can see different trends between the two models. The curves of the "DM" and "SA" are clearly demarcated around 1,400 Hz. The mapping-based spectrogram had better recovery in the low-frequency part but worse recovery in the middle and high-frequency parts compared with those of the masking-based spectrogram. The cut-off point of 1,400 Hz is consistent for

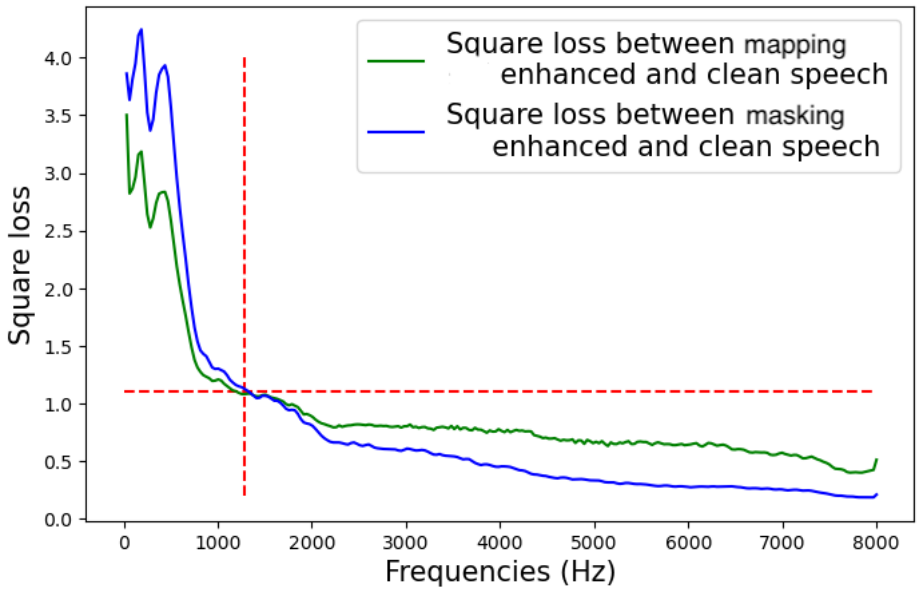


Fig. 3.2 Square loss of mapping & masking in different frequencies (0 – 8,000 Hz) on Voice Bank training set.

the two datasets. With the masking-based spectrogram, the recovery of each frequency was uniform and stable.

3.2.2 Analysis on Dynamic Ranges of mapping-based System

Fig. 3.2 shows the square loss of the mapping and masking approaches: $\mathscr{L}_{\mathscr{D}\mathscr{M}}$ and $\mathscr{L}_{\mathscr{I}\mathscr{A}}$. The loss of the low-frequency part was significantly larger than that of the high-frequency part. The 40th point (about 1,400 Hz) is marked with a red dashed line in accordance with the analysis in the previous section. When the frequency was lower than 1,400 Hz, the loss increased significantly, while it was stable for the frequency higher than 1,400 Hz. Thus, the dynamic range of loss differs for low-frequency and high-frequency regions in the DM system. This suggests that the main loss comes from the low-frequency part of the spectrogram, which may affect the recovery of the high-frequency part. Although the output of the masking-based network is not strictly distributed between 0 and 1, it still limits the output of the network and makes the difference between high and low frequencies smaller, which can alleviate the aforementioned problems. In the linear spectrogram, the energy difference between high and low frequencies is 50–90dB. Though the energy distributed at high and low frequencies is very different, there is some correlation between the high and low frequencies of the spectrogram. Therefore, the mapping-based SE should use the full-band

information to help the recovery of the low-frequency signal, while the high-frequency SE can be separately designed. This is not the case in the masking-based SE.

3.3 Spectrograms Fusion (SF)

3.3.1 Subband-based Spectrogram Fusion (SBSF)

In Chapter 3.2.1, we observed that the mapping-based SE system had better recovery in low frequencies, while the masking-based SE system had better recovery in high frequencies. Moreover, the mapping-based SE system can be divided into two dynamic ranges according to the square loss. It suggests the complementary between the mapping-based and masking-based SE systems. Combining these two analyses, we divide the whole spectrogram into two parts around 1,400 Hz and investigate the effective combination of these two learning targets. We call the high-frequency enhancement and low-frequency enhancement HEnh and LEnh, respectively. Both HEnh and LEnh use the full-band spectrogram as input feature to predict the sub-band output feature. We use the 257-dimensional linear spectrogram as an input feature. LEnh predicts 1-th to 40-th low-frequency bins of the spectrogram, and HEnh predicts 41-th to 257-th high-frequency bins. We investigate the following issues in the SBSF:

1) Is it effective to enhance different subbands with different learning target?

- **DM_L + SA_H**: directly concatenates DM-based LEnh and SA-based HEnh subband spectrograms. Considering the loss ratio, we select the mapping-based SE to enhance the low frequencies of the spectrogram, and the masking-based SE for the high-frequency spectrogram.

2) Is it effective to process different subbands by mapping separately?

- **DM_L + DM_H**: directly concatenates DM-based LEnh and DM-based HEnh subband spectrograms. In Section II–C, we reason that the poor high-frequency recovery of the mapping SE was because high energy in the low frequencies prevents effective training in the high-frequency regions due to the different dynamic ranges of the mapping targets. Thus, we design a separate DM-based method for the high-frequency region.

3) Is it more effective to combine the full-band and subband SE than combing two subband-based SE?

For DM_L + DM_H, we use subband SE to deal with different dynamic ranges of mapping. However, it ignores the global information, and may cause incoherence in the spectrogram. Thus, we also design a full-band and subband hybrid enhancement methods. Three

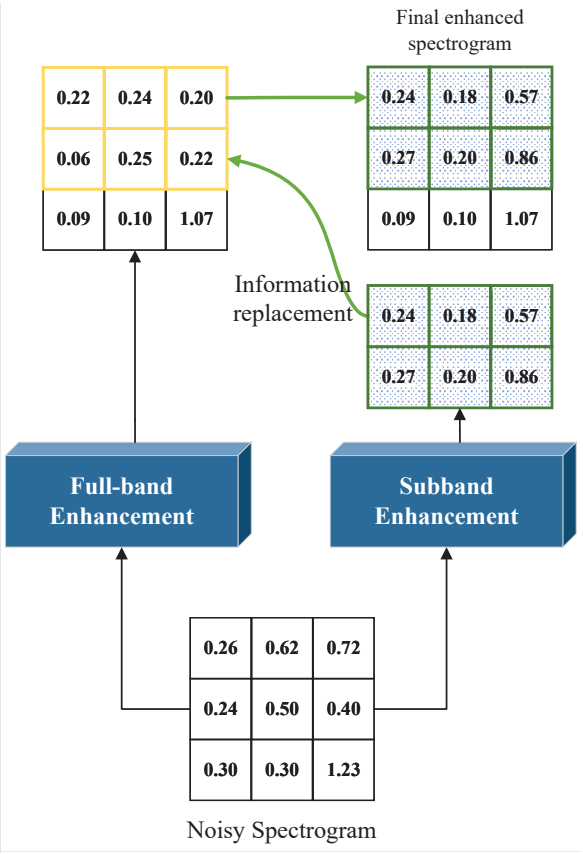


Fig. 3.3 The flowchart of subband-based optimization: the subband enhanced spectrogram will be used to replace the corresponding information of full-band enhanced spectrogram.

steps are used: (1) full-band SE, which computes a full-band enhanced spectrogram; (2) subband SE, which obtains a subband enhanced spectrogram; and (3) information replacement, which uses the subband enhanced information to replace the corresponding information in the full-band spectrogram. Fig. 3.3 shows the flowchart of the proposed method. Compared with DM_L + DM_H, we propose the following method:

- **DM_F** $\rightarrow$ **DM_H**: DM-based full-band enhancement with the DM-based HEnh replacement.

Furthermore, we inverstigate other full-band and subband hybrid combinations. Specifically, we design the following methods:

- **DM_F** $\rightarrow$ **DM_L**: DM-based full-band enhancement with the DM-based LEnh replacement.
- **DM_F** $\rightarrow$ **SA_H**: DM-based full-band enhancement with the SA-based HEnh replacement.

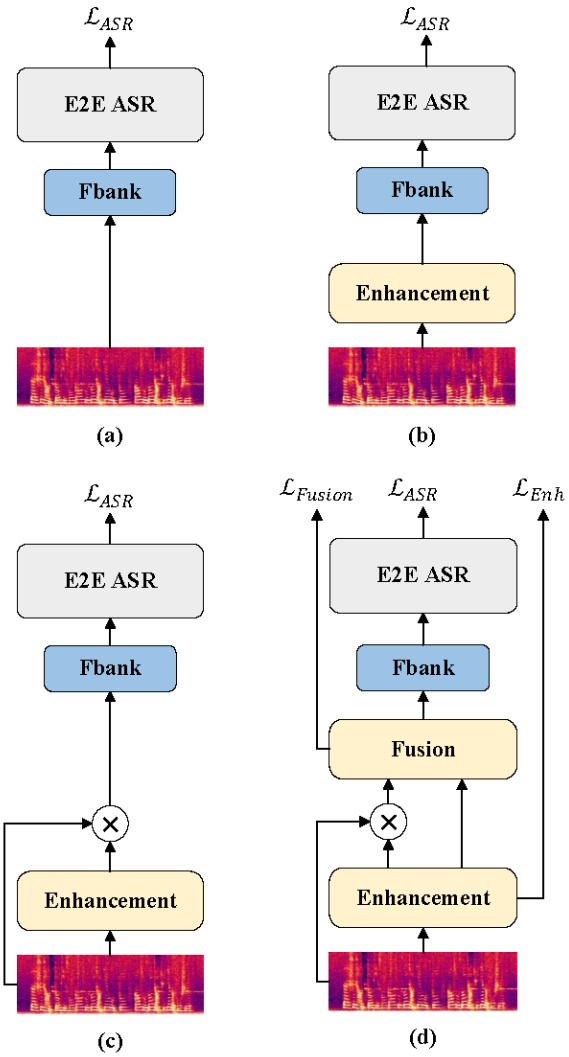


Fig. 3.4 Overview of robust ASR systems.

- **SA_F** $\rightarrow$ **DM_L**: SA-based full-band enhancement with the DM-based LEnh replacement.
- **SA_F** $\rightarrow$ **SA_H**: SA-based full-band enhancement with the SA-based HEnh replacement.
- **SA_F** $\rightarrow$ **DM_H**: SA-based full-band enhancement with the DM-based HEnh replacement.

3.3.2 Minimum Difference Masks-based SF (MDMs-SF) for Noise Robust ASR

Our proposed MDMs-SF E2E robust ASR method is composed of three modules: an enhancement module, a fusion module, and a recognition module. A flowchart of the proposed approach is shown in Fig. 3.4–(d). Compared with the mapping-only (Fig. 3.4–(b)) or masking-only (Fig. 3.4–(c)) SE front-end for ASR, the proposed MDMs-SF ASR system utilizes both the mapping and masking systems simultaneously.

Enhancement module

To obtain mapping- and masking-based spectrograms simultaneously, we train the enhancement module in a multi-target learning manner:

$$\mathscr{L}_{Enh} = \alpha \mathscr{L}_{DM} + (1-\alpha)\mathscr{L}_{SA} \tag{3.2}$$

Here, α is a hyperparameter for adjusting the loss from the two outputs.

Fusion module

SF [4] is an effective approach for exploiting the complementarities between mapping- and masking-based SE systems. It fuses the T-F bins from the mapping and masking spectrograms that are closest to the true labels to a single spectrogram. We use a deep neural network to estimate the minimum difference masks (MDMs) between $\widetilde{X}$ and $\ddot{X}$ [4]. The labels to train the MDM estimator are obtained from the enhanced and clean spectrogram. By comparing the Euclidean distance between the two spectrograms and the clean spectrogram, we set the corresponding MDM with a closer distance to 1, otherwise 0. Thus, each enhanced magnitude spectrogram has a corresponding MDM for each T-F bin, which gives a smaller absolute distance from the target magnitude spectrograms. In this paper, $\widetilde{MDM}$ and $M\ddot{D}M$ are used to extract the better parts of $\widetilde{X}$ and $\ddot{X}$, respectively.

Because the spectrogram is continuous, the MDMs in the testing stage are real values in (0, 1). The loss is defined as follows:

$$\mathscr{L}_{Fusion} = \frac{1}{TF}\sum_{i} ||\widetilde{MDM_i} - MDM_i||_F^2 \tag{3.3}$$

After predicting the MDMs, nonlinear selection and fusion processing are conducted to obtain the fusion speech magnitude spectrogram:

$$\widehat{X} = \widetilde{MDM} \odot \widetilde{X} + M\ddot{D}M \odot \ddot{X} \tag{3.4}$$

Recognition module

A speech Transformer [69] with self-attention [70] is used for the E2E ASR component. Except for the different input features, the structure of the model is not changed. We use The log Mel-filterbank (LMFB) as the input feature of the recognition module. We can easily obtain LMFB from the enhanced features.

Joint training

We propose a robust E2E ASR system that transforms noisy speech signals into text using a single network. The SE networks, the SF network, and ASR based on speech transformer are implemented with a single neural network. The parameters are updated by the stochastic gradient descent. SE, SF and ASR network are finetuned with joint training. The loss function of the joint training is defined as follows:

$$\mathscr{L}_{Joint} = \beta \mathscr{L}_{ASR} + (1-\beta)\mathscr{L}_{Enh} + \gamma \mathscr{L}_{Fusion} \tag{3.5}$$

The hyperparameter β, γ control the loss between $\mathscr{L}_{ASR}$, $\mathscr{L}_{Fusion}$ and $\mathscr{L}_{Enh}$.

3.4 Experiments

3.4.1 Datasets

The Voice Bank and REVERB Challenge datasets were used to evaluate SBSF under additive noise and reverberation conditions, respectively. The synthesized noisy Aishell dataset was used to evaluate the MDMs-SF E2E ASR.

Voice Bank

For the training set, we selected 26 speakers from the Voice Bank corpus [71]—13 male and 13 female—from the same accent region (England). Approximately 400 sentences are available from each speaker. The training set contains 10, 340 sentences. For validation set, we selected another 2 speakers from the Voice Bank corpus [71]—1 male and 1 female—from

the same accent region (England). The the validation set contains 1,232 sentences. Two artificially generated (speech-shaped noise and babble) and eight real noise recordings from the Demand database [72, 73] were used to synthesize the training and validation sets. The signal-to-noise ratio (SNR) values used for training were 15, 10, 5, and 0 dB. Two other speakers from England in the same corpus, a male and a female, and five other noises from the Demand database were used to create the test set. The SNR values used for testing were 17.5, 12.5, 7.5, and 2.5 dB. All data were sampled at 16 kHz.

REVERB Challenge [74]

The challenge uses utterances spoken by a single stationary distant-talking speaker with 1-channel (1ch), 2-channel (2ch) or 8-channel (8ch) microphone arrays in reverberant meeting rooms. In this paper, we use only single-channel data of channel-1 to train the model. The training set contains 7,861 utterances. We used the development set for model selection.

Synthesized Noisy Aishell

We used the Aishell ASR corpus [75] and the PNL 100 Nonspeech dataset [1] to synthesize the experimental dataset for ASR. For the training set, we randomly selected 70 kinds of noise and randomly synthesized the training set from the Aishell corpus with SNR values of 0, 5, 10, 15, and 20. We did not use the development set to tune or select the system. On the other hand, we used the development and the test set from the Aishell corpus to synthesize the test sets. For test set 1, we randomly selected 15 kinds of noise, different from those in the training set, and the whole development set from the Aishell corpus with same SNR values as training set. For test set 2, we used the remaining 15 kinds of noise and randomly synthesized the test set from the Aishell corpus according to SNR values of -5, 2.5, 7.5, 12.5, and 17.5. In summary, test set 1 contained some unknown noise, while all kinds of noise of test set 2 was unknown.

3.4.2 Model Settings

SBSF

All networks were implemented using TensorFlow. The model parameters were randomly initialized. The implementation of all networks was based on two-layer bidirectional long short-term memory neural networks (Bi-LSTM). The number of nodes in each hidden layer was 1024. For SA and DM, the input was a 257-dimensional spectrogram, and the enhanced

[1] http://web.cse.ohio-state.edu/pnl/corpus/HuNonspeech/HuCorpus.html

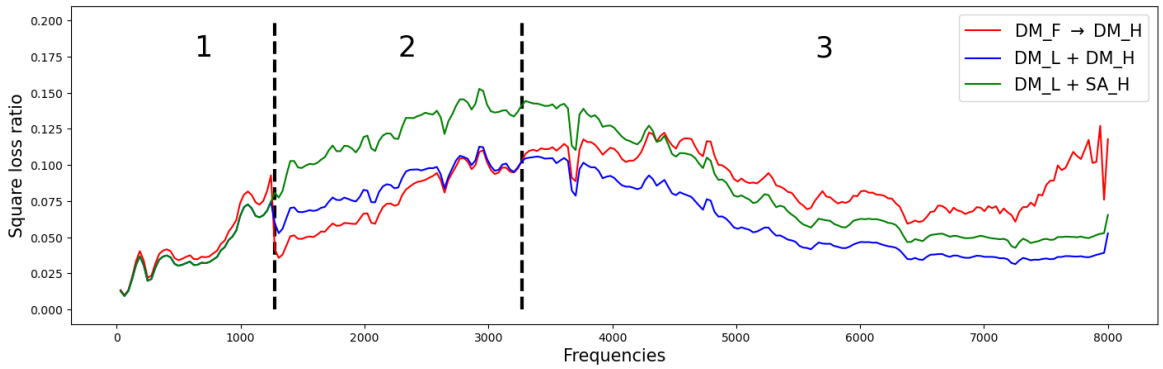


Fig. 3.5 Square loss ratio (the lower, the better) of DM_F → DM_H, DM_L + DM_H, and DM_L + SA_H at different frequencies (257-dimensional linear spectrogram) on Voice Bank training set, which were calculated with Eq. (3.1).

spectrogram output also had 257 dimensions. The activation of hidden layers for SA and DM was ReLU. For the activation function of the output layer, ReLU was chosen for SA and a linear function was used for DM. In addition, we estimated the 217-dimensional high-frequency and 40-dimensional low-frequency spectrograms for the HEnh and LEnh, respectively.

MDMs-SF E2E ASR

The enhancement module has three bidirectional long short-term memory (BLSTM) hidden layers, each having 512 nodes. The input and output were both 257-dimensional magnitude spectrograms. We used a short-time Fourier transform with a 32-ms Hamming window and a 16-ms window shift to obtain the 257-dimensional magnitude spectrograms for feature extraction. The Fbank was 80-dimensional. This module was trained in a multi-target learning manner to obtain mapping- and masking-based SE systems. The fusion module has three fully connected layers, with each hidden layer having 512 nodes. The input can be noisy, mapping-based, and masking-based magnitude spectrograms. Moreover, the output has two MDMs and two enhanced spectrograms. For the recognition module, we used the speech transformer with self-attention, under the same settings as described in [76]. Specifically, we used six self-attention blocks as encoders and six self-attention blocks as the prediction network. For each module, we performed pre-training. For pre-training enhancement module and fusion module, we used the data described in Section 3.1. For pre-training the recognition module, we used the clean data. The hyperparameter α was set to 0.5, the hyperparameter β was set to 1, and the hyperparameter γ was set to 0, meaning the ASR loss is primarily used. All model training ran for 60 epochs.

Table 3.1 Performance of Different SE Systems on Voice-bank Test Set.

Systems	**SIG**	**BAK**	**OVRL**	**PESQ**
Noisy	3.35	2.44	2.63	1.97
DM	3.85	2.55	3.23	2.60
SA	3.65	2.49	3.07	2.51
DM $\rightarrow$ DM	3.89	2.55	3.25	2.60
SA $\rightarrow$ DM	3.89	2.54	3.23	2.56
DM_L + SA_H	3.76	3.04	3.18	2.61
DM_L + DM_H	4.06	3.11	3.38	2.70
DM_F $\rightarrow$ DM_H	**4.09**	**3.12**	**3.42**	**2.74**
DM_F $\rightarrow$ DM_L	4.02	2.59	3.35	2.69
DM_F $\rightarrow$ SA_H	3.94	3.05	3.27	2.63
SA_F $\rightarrow$ DM_H	3.76	3.05	3.18	2.60
SA_F $\rightarrow$ DM_L	3.87	2.50	3.21	2.58
SA_F $\rightarrow$ SA_H	3.71	3.00	3.11	2.52

Table 3.2 Performance of Different SE Systems on Reverb Chanllenge 2014 far room test set.

Systems	**Far room 1**		**Far room 2**		**Far room 3**	
	PESQ	**STOI**	**PESQ**	**STOI**	**PESQ**	**STOI**
Noisy	2.59	84.69%	1.99	78.20%	1.87	71.31%
SA	2.91	89.81%	2.33	84.74%	2.26	83.22%
DM	2.74	88.74%	2.45	85.17%	2.36	84.21%
DM_F $\rightarrow$ DM_H	2.91	89.73%	**2.56**	**86.99%**	**2.48**	**86.10%**

3.4.3 Evaluation Metrics

We used several composite measures to evaluate different SE systems. Composite objective measures are obtained by linearly combining existing objective measures: C_{sig} for a five-point scale of signal distortion (SIG) [77]; C_{bak} for a five-point scale of background intrusiveness (BAK) [77]; C_{ovl} for the overall quality (OVRL, [1=bad, 2=poor, 3=fair, 4=good, 5=excellent]) [77]. The three composite measures obtained from log likelihood ratio (LLR) [77], the perceptual evaluation of speech quality (PESQ) [78], segmental SNR (segSNR) [77], and weighted-slope spectral (WSS) [79] distance:

$$C_{sig} = 3.093 - 1.029 * LLR + 0.603 * PESQ - 0.009 * WSS \tag{3.6}$$

$$C_{bak} = 1.634 + 0.478 * PESQ - 0.007 * WSS + 0.063 * segSNR \tag{3.7}$$

Table 3.3 Performance of Different SE Systems on Reverb Chanllenge 2014 near room test set.

Systems	Near room 1		Near room 2		Near room 3	
	PESQ	**STOI**	**PESQ**	**STOI**	**PESQ**	**STOI**
Noisy	3.11	95.18%	2.39	92.32%	2.27	89.38%
SA	**3.40**	95.95%	2.78	93.87%	2.59	90.99%
DM	2.94	93.35%	2.78	92.50%	2.70	91.05%
DM_F $\rightarrow$ DM_H	3.18	95.05%	**2.95**	**94.19%**	**2.83**	**92.90%**

Table 3.4 Performance of Different SE Systems on Different Noisy Conditions (Unseen, synthesized, clean speech from Voice Bank dataset, noisy from non-speech 100).

Systems	Crowd Noise				Machine Noise			
	SIG	**BAK**	**OVRL**	**PESQ**	**SIG**	**BAK**	**OVRL**	**PESQ**
Noisy	1.45	1.74	1.27	1.18	1.85	1.87	1.47	1.19
DM	1.97	2.03	1.62	1.35	2.19	2.02	1.75	**1.38**
SA	1.66	2.14	1.42	1.30	1.85	**2.14**	1.53	1.33
DM_F $\rightarrow$ DM_H	**2.12**	2.12	**1.74**	**1.44**	2.15	2.02	1.72	1.37
Systems	**Alarm and Siren**				**Wind**			
	SIG	**BAK**	**OVRL**	**PESQ**	**SIG**	**BAK**	**OVRL**	**PESQ**
Noisy	1.31	1.50	1.15	1.13	2.65	1.77	1.90	1.35
DM	2.38	2.15	1.90	1.51	3.00	2.22	2.36	1.85
SA	1.63	2.17	1.44	1.39	2.53	2.27	2.10	1.77
DM_F $\rightarrow$ DM_H	**2.62**	**2.24**	**2.08**	**1.64**	**3.21**	2.29	**2.51**	1.91

$$C_{ovl} = 1.594 + 0.805 * PESQ - 0.512 * LLR - 0.007 * WSS \tag{3.8}$$

We also adopted the Short-Time Objective Intelligibility (STOI) [80] as evaluation metrics. For all metrics, higher values indicate better performance. For the ASR back-end, we used the character error rate (CER) as an evaluation metric.

3.4.4 Results and Analysis of SBSF

Table 3.1 shows the performance of the different SE systems. Generally, "DM" performed better than "SA" in this study. The simple two-stage methods (DM$\rightarrow$DM and SA$\rightarrow$DM) did not get much improvement from the baseline DM.

We find that the mapping-based SE performed better below 1,400 Hz, while the masking-based SE system was better above 1,400 Hz. However, we find that a simple combination of

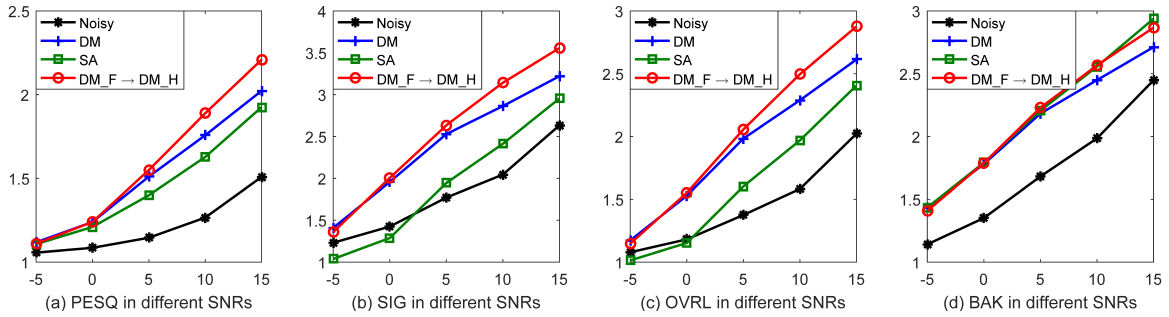


Fig. 3.6 Performance of evaluation measures (PESQ, SIG, OVRL, BAK) of different enhancement systems in SNRs (-5, 0, 5, 10, and 15 db) conditions: -5 db was an unseen condition, and the noisy conditions were unseen. The horizontal axis represents SNRs, and the vertical axis represents the value of the evaluation metric.

the two, "DM_L + SA_H," did not improve so much. This is because even in some high-frequency regions, mapping often produced better performance. However, "DM_L + DM_H" had a relatively large improvement. This shows that subband enhancement considering dynamic ranges is more beneficial for the mapping method.

Furthermore, the full-band and subband hybrid approaches showed better performance than directly concatenating subband spectrograms. Nevertheless, the experimental results also show that the masking-based subband optimization was worse than the mapping-based subband optimization. Table 3.2 and Table 3.3 show the results of different methods on the REVERB Challenge test set. "DM_F $\rightarrow$ DM_H" further improved the performance of the mapping-based system.

Fig. 3.5 shows the square loss ratio of "DM_F $\rightarrow$ DM_H", "DM_L + DM_H", and "DM_L + SA_H". We divided the full-band frequency into three parts. Part 1 had no significant differences among these methods for low-frequency recovery. We call Part 2 middle-frequency and Part 3 high-frequency. "DM_L + SA_H" had poor recovery in middle frequencies. Although the recovery of "DM_L + SA_H" in other regions was not much different from other methods, a large degradation was observed in PESQ, which illustrates the importance of Part 2 recovery.

We investigated the performance of different models at different frequencies according to the square loss ratio. The curves of the "DM_F $\rightarrow$ DM_H" and "DM_L + DM_H" are clearly demarcated around 3,200 Hz in the middle and high frequencies. "DM_F $\rightarrow$ DM_H" worked well for middle frequencies (Part 2) but not for high frequencies (Part 3). This suggests that about 3,200 Hz would be another cut-off point for dividing the frequencies into two dynamic ranges.

Table 3.5 Performance of SE in the test set 1.

Systems	**Test set 1**			
	SIG	**BAK**	**OVRL**	**PESQ**
Original noisy	2.80	**2.64**	2.12	1.50
DM_separate	3.46	1.82	2.73	2.04
SA_separate	3.09	1.66	2.43	1.83
Fusion_separate	**3.57**	1.83	**2.80**	**2.07**
DM_Joint	1.24	1.19	1.13	1.15
SA_Joint	2.74	1.49	2.07	1.48
Fusion_Joint_DM	1.53	1.25	1.21	1.07
Fusion_Joint_SA	2.63	1.46	2.03	1.52
Fusion_Joint	1.05	1.09	1.02	1.08

Table 3.6 Performance of SE in the test set 2.

Systems	**Test set 2**			
	SIG	**BAK**	**OVRL**	**PESQ**
Original noisy	2.37	**2.28**	1.83	1.41
DM_separate	2.81	1.60	2.19	1.65
SA_separate	2.55	1.50	2.00	1.55
Fusion_separate	**2.87**	1.58	**2.22**	**1.65**
DM_Joint	1.12	1.21	1.08	1.17
SA_Joint	2.29	1.41	1.79	1.40
Fusion_Joint_DM	1.50	1.28	1.21	1.10
Fusion_Joint_SA	2.12	1.33	1.69	1.41
Fusion_Joint	1.01	1.08	1.01	1.12

We synthesized another test set to evaluate the SE systems on different signal-to-noise ratios (SNRs) and noise conditions. We used all clean speech in the test set from the Voice Bank dataset and chose four noisy conditions: crowd, machine, alarm, siren, and wind. These noise samples were selected from a dataset with 100 non-speech audio clips[2]. The SNRs we chose were -5, 0, 5, 10, and 15 db. Table 3.4 shows the performance of enhancement systems under multiple noisy conditions. Except for the machine noise, the improvement of our method was consistent. Fig. 3.6 shows the performance of different SE systems for multiple SNRs. "SA" did not perform well under low SNRs and was even lower than noisy speech on SIG and OVRL. The performance of other enhancement methods at low SNRs was

[2]http://web.cse.ohio-state.edu/pnl/corpus/HuNonspeech/HuCorpus.html

Table 3.7 CER (%) of different E2E systems with test set 1: the SNRs of the test set are known; the noise of the test set is unknown.

β	System	CER (%)					
		20dB	15dB	10dB	5dB	0dB	AVG.
-	Noisy	12.5	14.6	17.3	24.6	39.7	21.8
	DM_separate	14.6	19.4	29.3	51.7	81.1	39.3
	SA_separate	20.0	33.2	53.3	83.0	111.3	60.3
	Fusion_separate	13.3	18.7	29.7	55.1	89.8	41.5
0.5	DM_Joint	11.7	13.0	16.4	25.5	43.6	22.1
	SA_Joint	10.2	11.6	13.9	20.2	36.4	18.5
	Fusion_Joint	9.8	11.6	15.4	26.8	49.9	22.8
1	DM_Joint	11.1	12.3	15.6	22.8	40.2	20.5
	SA_Joint	9.8	11.4	13.8	20.2	36.1	18.3
	Fusion_Joint	**9.8**	**11.0**	**12.9**	**19.0**	**33.5**	**17.3**

Table 3.8 CER (%) of different E2E systems with test set 2: both the SNRs and the noise of the test set are unknown.

β	System	CER (%)					
		17.5dB	12.5dB	7.5dB	2.5dB	-5dB	AVG.
-	Noisy	20.3	24.5	30.5	45.4	76.7	40.0
	DM_separate	25.1	35.8	58.1	91.5	125.4	68.4
	SA_separate	31.1	48.8	78.6	113.5	157.4	87.4
	Fusion_separate	23.8	36.3	61.6	97.5	139.4	73.1
0.5	DM_Joint	16.5	20.7	31.0	54.9	102.6	45.9
	SA_Joint	14.2	17.1	23.8	38.0	83.3	35.8
	Fusion_Joint	14.9	19.9	33.3	57.9	106.6	47.4
1	DM_Joint	16.4	19.6	27.9	45.7	96.4	41.9
	SA_Joint	14.0	**16.7**	24.0	37.7	75.9	34.2
	Fusion_Joint	**13.8**	16.8	**23.0**	**36.8**	**73.9**	**33.3**

very similar. All methods showed better performance with the increase in SNR. Compared with other methods, the performance of "DM_H $\rightarrow$ DM_F" increased significantly as the SNR improved.

3.4.5 Results and Analysis of MDMs-SF E2E ASR

Impact of MDMs-SF ASR without joint training

From Table 3.5 to Table 3.8, although "Fusion_separate" achieves the best performance in SE tasks, "DM_separate" achieves better performance on ASR. The ASR back-ends of "DM_separate", "SA_separate", and "Fusion_separate" are all trained on clean data. Comparing the results of Table 3.7 and Table 3.8, high SNR is beneficial to the "Fusion_separate". However, when the SNR is low, the poor recognition performance of "SA_separate" affects the "Fusion_separate". The performance of the fusion system is between the two fused systems.

Impact of SF-based ASR with joint training

Joint training has different effects on "Fusion_Joint", "DM_Joint" and "SA_Joint". During joint training with β=1.0, the system did not introduce enhanced loss. In "Fusion_Joint", the fusion module fuses "Fusion_Joint_DM" and "Fusion_Joint_SA". The loss in speech signal "Fusion Joint Mapping" becomes serious, as it does no longer work for enhancement, but focuses on feature extraction for recogition. The ASR performance of the systems obtained by using joint training is greatly improved. We explored whether it is necessary to introduce an enhancement loss ($\beta = 0.5$) during joint training. The results indicate that it does not improve the ASR performance, but instead dramatically degrades it. We compared the two parts of the loss and found that the enhancement loss is larger than the ASR loss, which may affect the convergence of the ASR part. With β set to 1, the recognition rate constantly improved as the SNR improves. "Fusion_Joint" gives improved results in almost all cases, especially when the SNR is low. In the case of 0 dB SNR, "Fusion_Joint" gives a relative improvement of more than 7% compared with "SA_Joint", and close to 17% compared with "DM_Joint". This shows that leveraging the complementarities between mapping- and masking-based SE systems is effective for robust ASR, especially when the noise is large (i.e., low SNR). When the noise and SNR are both unknown, the performance of "Fusion_Joint" is improved compared with "DM_Joint" and "SA_Joint", though the improvement is not large. This may be because "Fusion_Joint" benefits from better enhancement systems. Improving robustness of deep-learning-based SE for unseen noise is important.

3.5 Conclusion

In this chapter, we have fused the enhanced spectrograms to improve the SE and ASR. We first investigated the complementary between the mapping-based and masking-based

frequency-domain SE systems. We found that the mapping-based SE system performs well at low frequencies, while the masking-based SE system performs smoothly for full-band information. Then, we proposed the SBSF to enhance the poor performance sub-band of the full-band spectrogram. It shows that refining the poor performance sub-band helps SE. Finally, we utilized the complementary between these two learning targets and proposed MDMs-based SF for noise ASR. The experimental results show that the complementary between different SE systems helps SE and ASR performance.

Chapter 4

Fusion of Multi-masked and Multi-resolution Spectrogram for SE

4.1 Introduction

Spectrogram is a common feature for frequency-domain SE [81, 82, 24]. For noisy speech, the noise will destroy the speech spectrogram structure [3], especially the texture information. In addition, due to the influence of noise, it is difficult to see the shape of many important structures such as formants in noisy spectrograms. This brings difficulties to SE. Some two-stage systems utilize the first stage to obtain enhanced spectrogram, which are then fed into the second stage as input features. Although this is helpful for network learning, the enhanced spectrogram obtained in the first stage often has great defects in details and information retention, which makes it difficult to obtain a greater improvement through the second stage.

In this work, we address the above problem by highlighting speech components. We extract the strong speech part and ignore others, so as to make the boundary of the speech component obvious. Strong speech part is determined based on the output mask by a trained masking-based SE system. The mask value shows the proportion of speech components present. We extract the spectral information corresponding to the position where the mask is larger than a certain threshold to form a feature map. The stacking of feature maps of strong speech components enables the input features to provide sufficient speech boundary information, making the neural networks more sensitive to the input features.

Furthermore, according to the resolution of the spectrogram, which is controlled by the length of framing time, it can be divided into wideband and narrowband [83]. The two kinds of spectrograms are much different and have their own characteristics [83]. Fig. 4.1 shows

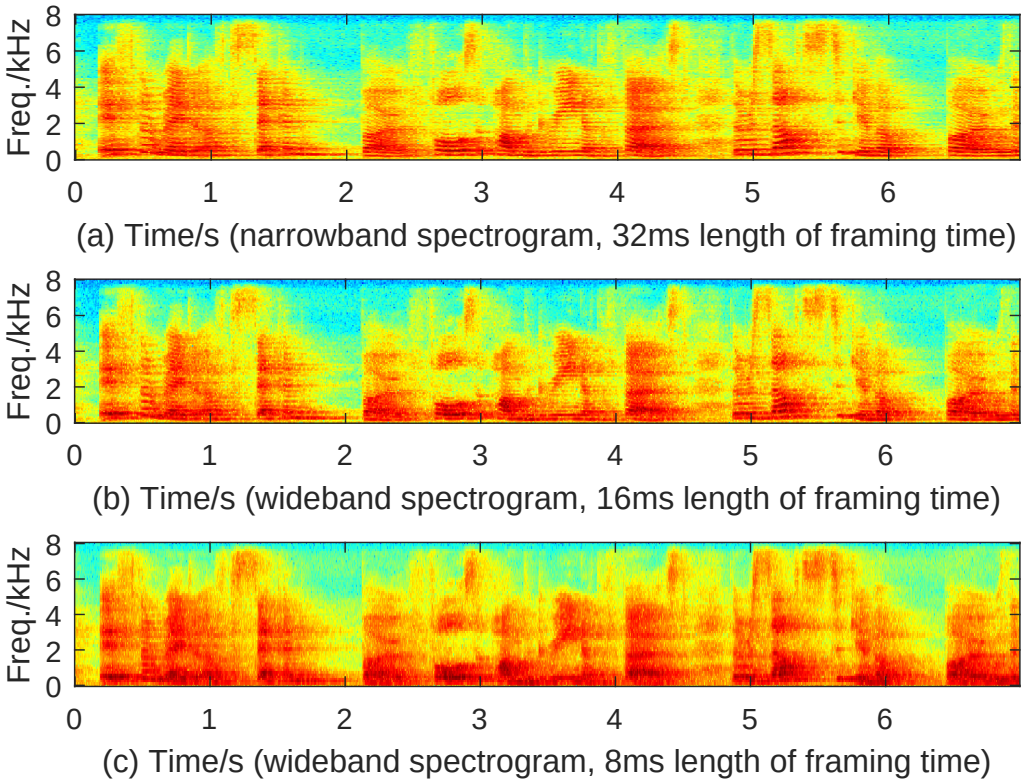


Fig. 4.1 Spectrogram examples extracted with different window lengths: (a) 32ms narrowband spectrogram; (b) 16ms wideband spectrogram; (c) 8ms wideband spectrogram.

the spectrograms extracted by 8ms, 16ms, and 32ms length of framing time. Because of the short time period of each frame, wideband spectrograms have better time resolutions and can capture the rapid amplitude changes [84]. In the wideband spectrograms, the formant information of speech can be clearly seen, but the harmonic frequencies cannot be seen [84]. On the other hand, the narrowband spectrograms have longer frame lengths. It is too long to capture the rapid changes in amplitude [84], but have better spectral resolutions. It is easy to see the position of the harmonics in the narrowband spectrograms, but difficult to spot the position of the formant [84].

Although there is information complementarity between spectrograms with different bandwidths, the current SE system conventionally uses spectrograms extracted by a single window length as input and output. Some related works use convolutional neural network to extract multi-scale features [85–92] instead of multiple bandwidth spectrogram inputs.

In this work, we design a multiple input SE system by incorporating 8ms and 16ms bandwidth spectrogram to the 32ms bandwidth spectrogram enhancement system. Spectrograms of different bandwidths are processed by multiple convolution blocks separately, and they are fused in the encoder. Two fusion positions are tried. More specifically, different bandwidth

spectrograms are fused only in the last encoder layer (MI-F) or layer by layer (MI-L). We propose to use Linear Blocks to fuse different information. For MI-F, one Linear Block is only added to the last encoder layer; for MI-L, Linear Blocks are added after each encoder layer.

4.2 Multi-masked and Multi-resolution Spectrograms Fusion

4.2.1 Multi-masked Spectrograms Fusion (MM-SF)

Spectrogram is a widely used feature to SE. However, the noise greatly deteriorates the structure of speech components in the spectrogram, especially when the signal-noise ratio is small. This will greatly affect the processing of the deep neural network with noisy spectrograms. We design multi-masked spectrograms fusion (MM-SF) to extract input features to highlight speech components from the noisy features.

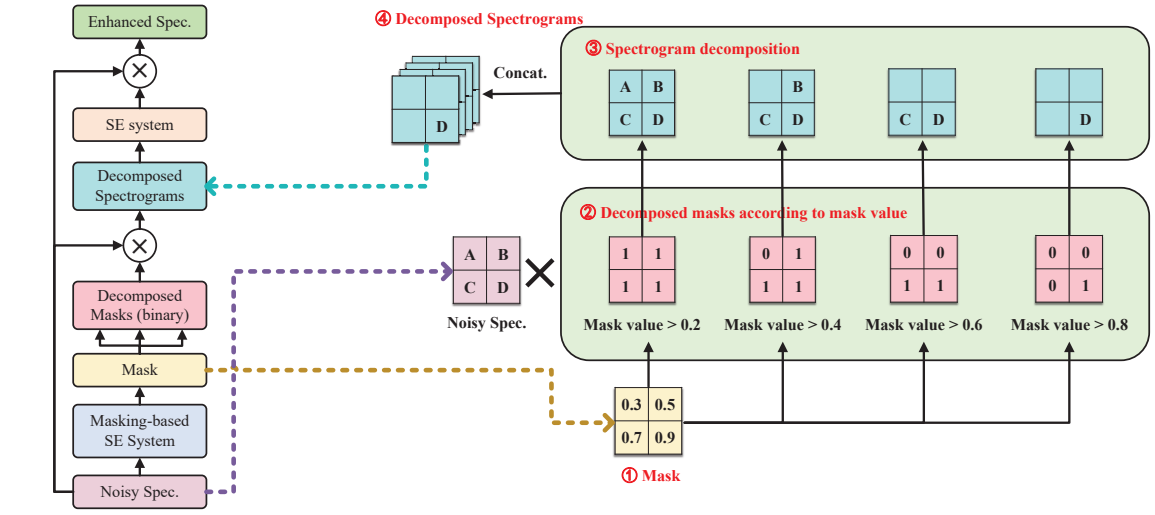


Fig. 4.2 (a) Flowchart of proposed multi-masked spectrograms fusion (MM-SF). (b) one example of multi-masked spectrograms extraction.

Fig. 4.2–(a) shows the flowchart of our proposed method. The proposed method has two stages. Masking-based SE is chosen to estimate a mask for the first stage. Then the mask is used to extract multi-masked spectrogram. The multi-masked feature $\mathbf{D}$ is as input feature to the second stage:

$$\widehat{\mathbf{M}} = \mathscr{N}(\mathbf{D}), \tag{4.1}$$

where $\widehat{\mathbf{M}}$ is the estimated mask of second stage. It should be emphasized that the enhanced spectrogram is not included as input feature in the second stage. Both two stages adopt the structure of CRN [93].

We extract the multi-masked spectrogram according to the value of $\widehat{\mathbf{M}}$. A mask value shows the proportion of speech components in noisy speech. Since most of the values of mask are in $(0,1)$, we divide $(0,1)$ into n equidistant intervals. The mask value greater than the lower bound of the interval is used to form a new feature map. The speech components with larger mask values have a high probability of being prominent. Our purpose is to only retain strong speech components in the decomposed spectrograms, ignoring other information. Thus, clear edge connection can be formed, so as to highlight the shape of the speech components.

This multi-masked spectrogram extraction can be divided into two steps: mask estimation and multi-masked spectrogram extraction. Each T-F bin in i–th masked spectrogram is:

$$\widehat{M}_i^{t,f} = \begin{cases} 0, |\widetilde{\mathbf{M}}|_i^{t,f} < \mathbf{b}_i^{t,f}, \\ 1, |\widetilde{\mathbf{M}}|_i^{t,f} > \mathbf{b}_i^{t,f}, \end{cases} \tag{4.2}$$

where $\mathbf{b}_i$ denotes the lower bound of the i–th interval and $\widetilde{\mathbf{M}}_i$ denotes the extraction mask. Thus, we can obtain $\mathbf{N}$ masks.

We use the masks to extract the information of the corresponding position in the noisy spectrogram:

$$\widehat{\mathbf{X}}_i = \widetilde{\mathbf{M}}_i \odot \mathbf{Y}, \tag{4.3}$$

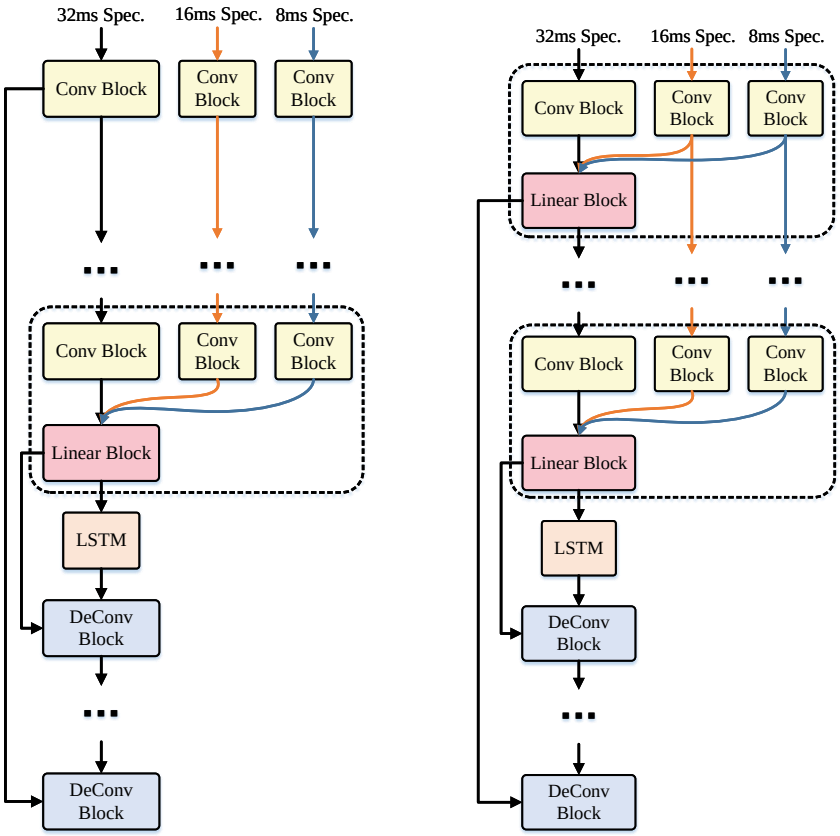
where $\widehat{\mathbf{X}}_i$ denotes the i–th masked spectrogram. For a feature map with a large lower bound of the mask interval, the speech information is more obvious. Finally, we concatenate the obtained feature maps to get a multiple-channel feature:

$$|\widehat{\mathbf{X}}| = \text{concat}(\widehat{\mathbf{X}}_i), i \in [1, N], \tag{4.4}$$

We use the multi-masked spectrograms as the input of the second stage network. It should be noted that we only use mask decomposition to obtain binary matrices instead of using the mask to enhance the spectrogram.

4.2.2 Multi-resolution Spectrograms Fusion

In this work, we utilize supplementary information of different bandwidth spectrograms. The proposed method inputs multi-bandwidth spectrograms simultaneously.



(a) Final Fusion (MI-F) (b) Layer-by-layer Fusion (MI-L)

Fig. 4.3 The flowchart of the proposed multi-resolution spectrograms fusion systems.

Structure of Neural Network

The flowcharts of the proposed methods are shown in Fig. 4.3–(a) and Fig. 4.3–(b). Both of Multi-input Final Fusion (**MI-F**) and Multi-input Layer-by-layer Fusion (**MI-L**) have an encoder, LSTM layers and a decoder. The network structure in front of the LSTM layers comprises the encoder. We use a Linear Block (shown in Fig. 4.4 to fuse the information of multiple bandwidth spectrograms:

$$h = \boldsymbol{LB}(fm_{32}, fm_{16}, fm_8), \tag{4.5}$$

where the fm_{32}, fm_{16}, fm_8 are feature maps of 32ms, 16ms, and 8ms bandwidth spectrograms respectively. $\boldsymbol{LB}$ represents the Linear Block, and h is the output of Linear Block. h and fm_{32} have the same feature dimension, which is realized by the linear layer of the Linear Block. For **MI-F**, Linear Block is only added to the last layer of the encoder. For **MI-L**, Linear Blocks are used to fuse the multiple bandwidth information after each Conv Block

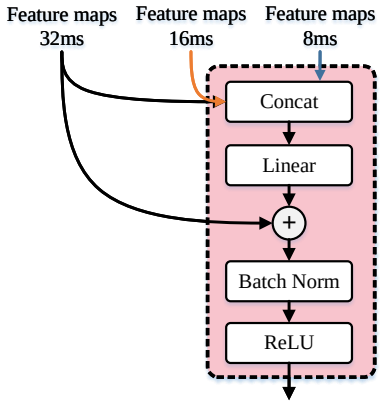


Fig. 4.4 Structure of Linear Block.

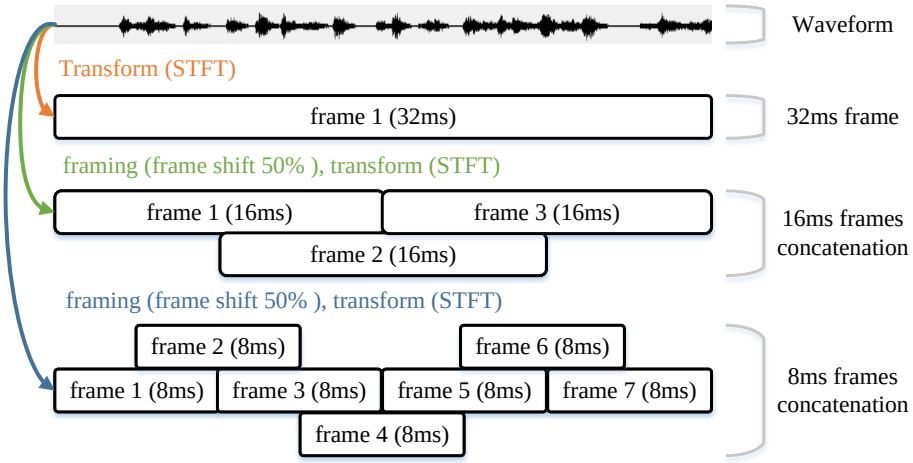


Fig. 4.5 16ms and 8ms features aligned with 32ms features for framing

in the encoder. The residual connection is used between the corresponding encoder layer and the decoder layer. For layers without a Linear Block, we directly input the output of the Conv Block into the corresponding layer of the decoder. When there is a Linear Block, we input the output of the Linear Block to the corresponding layer of the decoder. The proposed network can be expressed as follows:

$$\widetilde{M} = \mathcal{N}_{MI-F}(fm_{32}, fm_{16}, fm_8), \tag{4.6}$$

or

$$\widetilde{M} = \mathcal{N}_{MI-L}(fm_{32}, fm_{16}, fm_8), \tag{4.7}$$

where $\mathcal{N}_{MI-F}$ and $\mathcal{N}_{MI-L}$ are networks of proposed MI-F and MI-L methods. The final enhanced spectrogram can be obtained by Eq. (2.9).

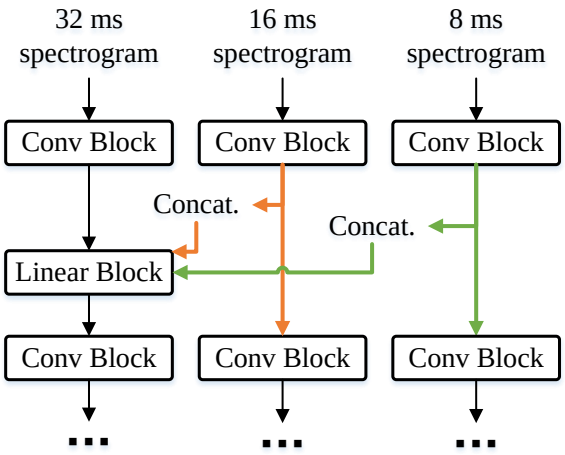


Fig. 4.6 Diagram of the frame concatenation.

Processing of Input Features

Spectrograms extracted with different time periods have different information in the same time frame. With different lengths of framing time and frame shift, the frame number and information of each frame are also different. In order to ensure that the corresponding frames of different bandwidth spectrograms are aligned when input to the network, we concatenate adjacent frames of 16ms and 8ms spectrograms. This process is applied after the Conv Block and before the Linear layer. In this work, the frame shift was 50%. One frame of 32ms spectrogram corresponds to adjacent 3 frames of 16ms spectrogram; one frame of 32ms spectrogram corresponds to adjacent 7 frames of 8ms spectrogram. In addition, to align the frames, the start and end time of the 32ms frame must be the same as that of 16ms/8ms after framing. This means that the i-th 32ms frame corresponds to the framing centered on the $2i$-th 16ms frame and the corresponding framing centered on the $4i$-th 8ms frame. The corresponding relationship is shown in Fig. 4.5. The diagram of the frame concatenation is shown in Fig. 4.6.

Training of the Network

The network takes SA masking as a learning target which calculates the loss with Eq. (2.7). The output of the network is the mask m for the 32ms spectrogram, which is used for enhancement in Eq. (2.9)

4.3 Experiments

We used the VB dataset [1], which is detailed described in the Chapter 3.4.1.

4.3.1 Feature Extraction

Feature extraction for MM-SF

We used the following parameters: window length was 512; hop length was 256; short-time fourier transform points was 512. We used the magnitude of the spectrogram as both input and output of this experiment.

Feature extraction for Multi-resolution spectrograms input system

We used the following parameters to extract 32ms spectrogram: window length was 512; hop length was 256; short-time Fourier transform points was 512. For 16ms/8ms spectrograms, these hyperparameters were set to 256/128, 128/64, 256/128. We used the magnitude of the spectrogram as both input and output of the experiments.

4.3.2 Baselines

We tested three baseline methods for MM-SF:
* CRN: the network was trained with Eq. (2.7); the input feature is noisy spectrogram; the input size was $1 \times 257 \times F$.
* CRN-stack: a two-stage method; it contains two CRNs, the input of the first CRN is noisy spectrogram, the input of the second CRN is the enhanced output from the first CRN.
* CRN-stack-w-noisy: the input of the second CRN is the concatenation of noisy and enhanced spectrogram; the other structures are same with "CRN-stack".

For the baselines to evaluate MI-L and MI-F, we tried **32ms**, **16ms**, and **8ms** spectrogram as input features for CRN. With different input feature dimensions, the dimensions of multiple bandwidth spectrograms will also have different dimensions after the convolutional processing, which will affect the number of nodes in the LSTM layers. For the input of 32ms spectrogram 1,792 LSTM layer nodes were used; 768 nodes for 16ms spectrogram and 256 nodes for 8ms spectrogram. All models had two LSTM layers. In order to make a fair comparison by considering the effect of the Linear Block, we add Linear Blocks after each Conv Block for the 32ms spectrogram baseline (**+ linear**).

[1] https://datashare.ed.ac.uk/handle/10283/2791

Table 4.1 Results of different enhancement systems.

System	SIG	BAK	OVRL	PESQ
noisy (input)	3.35	2.44	2.63	1.97
CRN	3.51	2.98	3.02	2.56
CRN-stack	3.60	3.04	3.10	2.62
CRN-stack-w-noisy	3.83	3.07	3.23	2.64
MM-SF	**4.02**	**3.10**	**3.37**	**2.72**

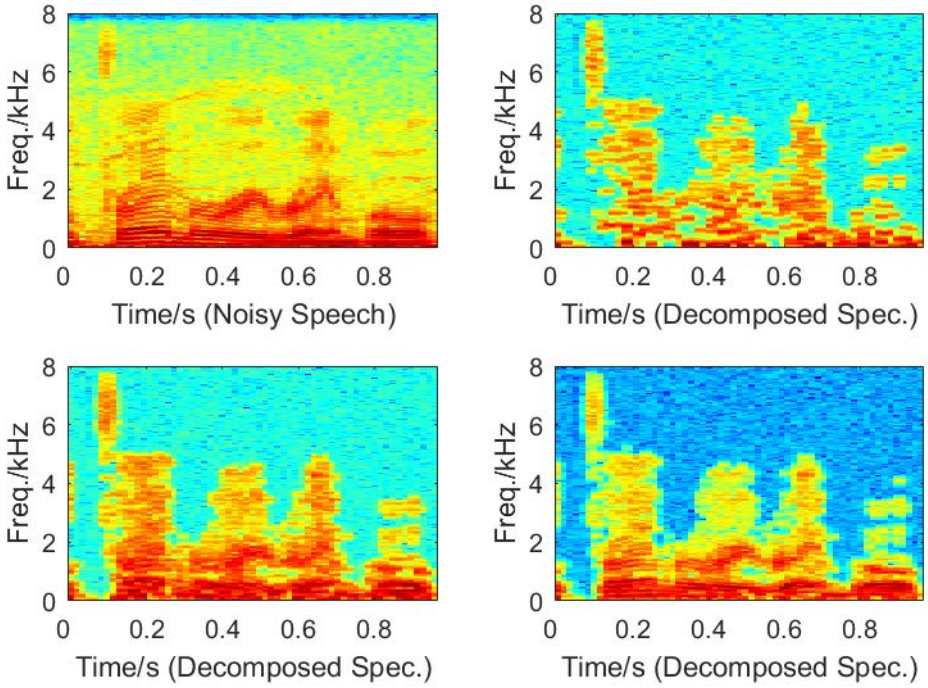
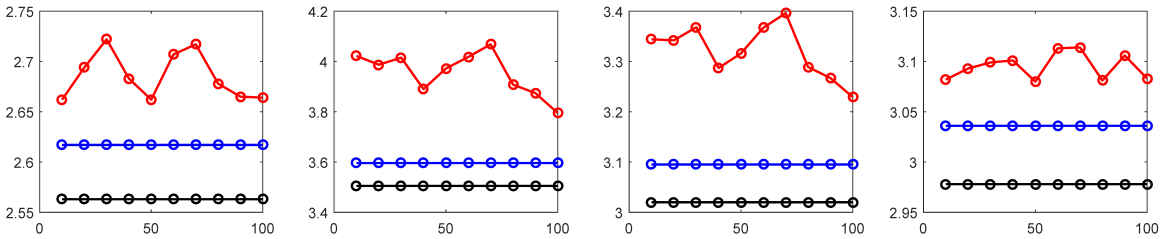


Fig. 4.7 Samples of decomposed spectrograms.

4.3.3 Neural Network Structure

We used the convolutional recurrent neural network (CRN) [93] in these experiments. In all experiments, except for the input dimensions, the network structures were the same. They have 5 encoder layers and 5 decoder layers. The parameters of the convolutional layer in the Conv Block are as follows: kernel size of (3,2), stride of (2, 1) and padding of (0, 1). The parameters of the deconvolutional layer in the DeConv Block are as follows: kernel size of (3,2), stride of (2, 1) and padding of (0, 0) except that (1, 0) was used for the 4th layer; the activation function of the last layer is ReLU, and the other layers are ELU. The numbers of feature maps in the encoder was $1 \rightarrow 16 \rightarrow 32 \rightarrow 64 \rightarrow 128 \rightarrow 256$, and the numbers of feature map in the decoder were $512 \rightarrow 256 \rightarrow 128 \rightarrow 64 \rightarrow 32 \rightarrow 1$. The LSTM had two layers, each layer had 1,792 nodes. We used n to represent the input feature dimensions.



(a) PESQ for different decomposition numbers (b) CSIG for different decomposition numbers (c) COVL for different decomposition numbers (d) CBAK for different decomposition numbers

Fig. 4.8 Measures on different decomposition numbers: Red line represents the "MM-SF"; Blue line represents the "CRN-stack"; Black line represents the "CRN".

4.3.4 Results of MM-SF

Performance of Different SE Systems

Table 4.1 shows the results of different enhancement systems. It is difficult to get the improved results simply by stacking the network. Although the "CRN-stack" had double number of parameters, the improvement of performance was small. Even when the noisy spectrogram was added back in the second stage, the gains were still small.

In Table 4.1, "MM-SF" denotes the proposed spectrogram decomposition-based system. The number of parameters of "MM-SF" is almost the same as "CRN-stack". Compared to simple stacking, spectrogram decomposition can provide more than 0.1 PESQ improvement. Because the decomposed spectrogram is still noisy, PESQ had an 0.15 improvement from the baseline "CRN" that only takes the noisy spectrogram as input. This indicates that the speech component awareness is helpful for enhancement tasks. Moreover, "MM-SF" was more effective than other methods in maintaining the speech signal, e.g., it had about 0.42 SIG improvement from the "CRN-stack".

The Effect of Decomposed Spectrograms

Figure. 4.7 shows a sample of the decomposed spectrograms. We randomly selected some high value boundaries. It can be clearly seen that some speech components are highlighted. It shows that spectrogram decomposition can make the CNNs more sensitive.

Effect of Different Decomposition Numbers

Figure. 4.8 shows the evaluation metrics for different decomposition numbers. The decomposition number of 30 is shown in Table. 4.1. All decomposition methods had improved performance over baselines, especially when the decomposition numbers were 30 and 70. This trend was obvious for PESQ, SIG, and OVRL. It was not the case that finer spectrogram

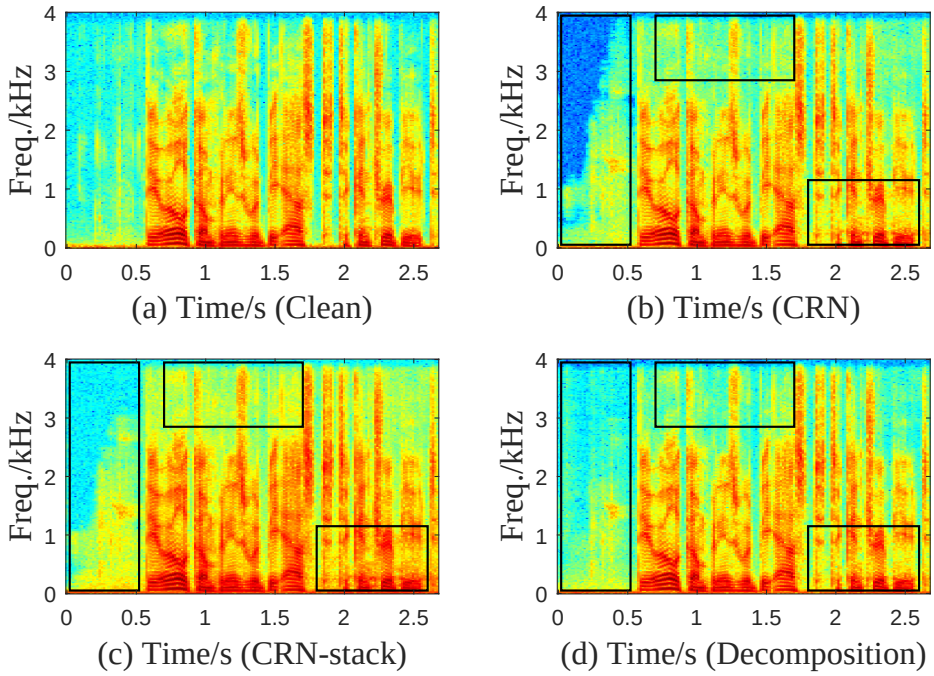


Fig. 4.9 Spectrograms of different enhancement systems: (a) clean speech; (b) CRN enhanced speech; (c) CRN-stack enhanced speech; (4) Decomposition enhanced speech.

decomposition could lead to better enhancement performance. This implies that the appropriate decomposition number needs to be found when decomposing the spectrogram. Moreover, the BAK was not much changed. The decompositions number was more likely to affect the recovery of speech components and the overall signal.

Effect of Spectrogram Decomposition on Spectrogram

Figure. 4.9 shows the spectrograms of different enhancement systems. "CRN" had serious information loss in the silent regions. Although "CRN-stack" had alleviated this problem, there was residual noise. Moreover, the energy of "CRN-stack" was greater in the speech regions, and some detailed information was lost. Compared with the other two methods, the proposed "MM-SF" had better spectrogram recovery.

Effect of Spectrogram Decomposition on Feature Maps

We selected some feature maps from output of conv2d_1, which are shown in Figure. 4.10. The noise in the feature maps extracted by "MM-SF" was greatly suppressed. In addition, the speech signal part of the feature map was also better preserved, especially the middle and

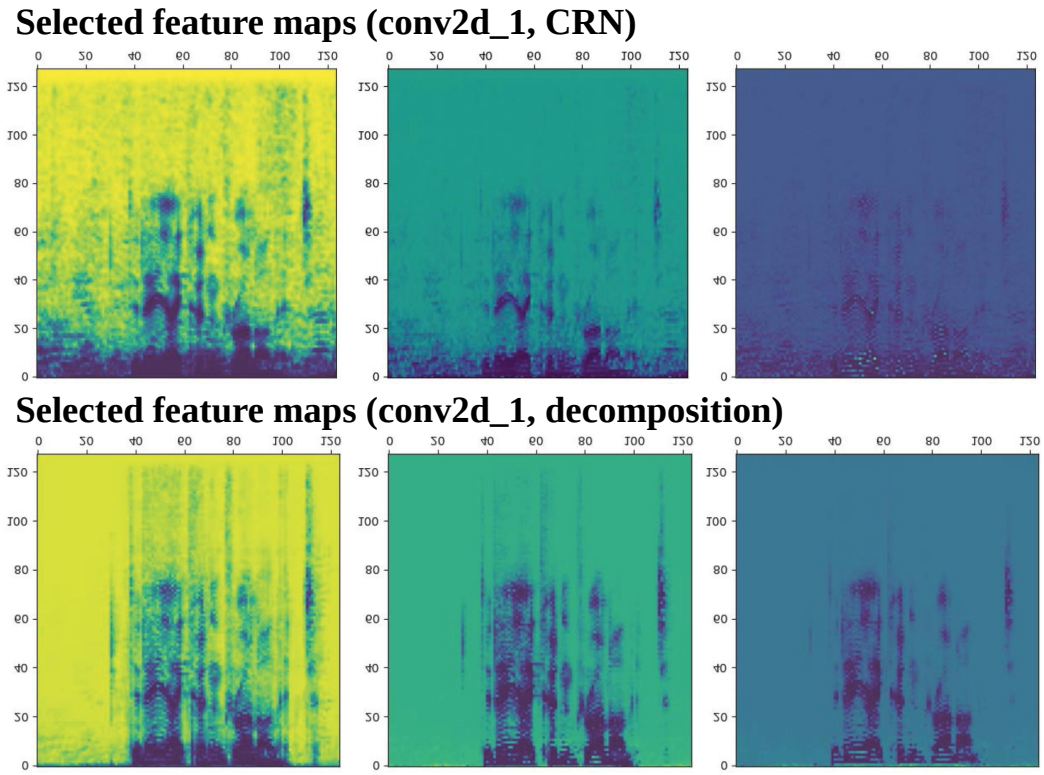


Fig. 4.10 Selected feature maps of baseline (CRN) and proposed decomposition method.

high frequency parts. This shows that the proposed spectrogram decomposition can help to alleviate the robustness problem.

4.3.5 Results of Multi-resolution Input System

Effect of Different Bandwidth

Table 4.2 shows the results of different SE systems. SE systems were greatly affected by the bandwidth of input and output features. Compared with the "16ms" and "8ms" systems, the "32ms" system obtains the best PESQ. With the increase of the bandwidth, the PESQ score tends to decrease. However, the wideband systems had the better speech signal recovery according to SIG, but the "8ms" system had the worst performance in suppressing intrusion noise (BAK) and overall signal recovery (OVRL). Due to the transient nature, the speech signal is periodic in the range of vowels. The "8ms" spectrogram is too short to cover transient stability, thus the "8ms" system had the worst performance.

Table 4.2 Results of different enhancement systems: 8ms (16ms, 32ms) feat. represents that 8ms (16ms, 32ms) feature as input and output feature; 8ms (16ms) aux. represents that the auxiliary feature is 8ms (16ms); 8, 16ms aux. represents that the auxiliary features are both 8ms and 16ms spectrograms.

SYSTEMS		**SIG**	**BAK**	**OVRL**	**PESQ**
noisy (original)		3.35	2.44	2.63	1.97
CRN	8ms feat.	3.61	2.92	2.92	2.26
	16ms feat.	3.62	3.07	3.02	2.48
	32ms feat.	3.51	2.98	3.02	2.56
	+ linear	3.56	3.14	3.01	2.50
MI-F	8ms aux.	3.69	**3.25**	3.16	2.66
	16ms aux.	3.61	3.14	3.07	2.57
	8, 16ms aux.	3.54	3.20	3.03	2.56
MI-L	8ms aux.	3.51	3.19	3.03	2.61
	16ms aux.	3.71	3.18	3.13	2.59
	8, 16ms aux.	**3.81**	3.22	**3.22**	**2.66**

Table 4.3 Input dimension (32ms, 16ms, 8ms) of Linear Block in different encoder layers: we use the output dimension of Conv Block (n) $\times$ the number of framing m.

Encoder Layers	**32ms**	**16ms**	**8ms**
1	128×1	64×3	32×7
2	63×1	31×3	15×7
3	31×1	15×3	7×7
4	15×1	7×3	3×7
5	7×1	3×3	1×7

Effect of Linear Block

We directly added a Linear Block to the 32ms-based system for fair comparisons. A Linear Block was added after each Conv Block in the encoder without introducing auxiliary information of other bandwidths. The results in Table 4.2 show that adding Linear Blocks slightly improved SIG and BAK scores. However, OVRL and PESQ of the enhanced speech signal are degraded.

Effect of MI-F

In the MI-F method, a Linear Block is added to the last layer of the encoder. The experimental results in Table 4.2 show that the best performance was obtained when using the "8ms aux.". With "16ms aux." and "8, 16ms aux.", SIG, BAK, and OVRL were improved but the improvement of PESQ was limited. The results show a trend that "8ms aux." was better

than “16ms aux.”, and “16ms aux.” was better than “8, 16ms aux.”. We reason that it is difficult for a single linear layer to incorporate a lot of information. Table 4.3 shows the input dimension of the Linear Block in different encoder layers. The 16ms spectrogram contains 9 dimensions (3×3) in the fifth encoder layer, while there are only 7 dimensions (1×7) for the 8ms spectrogram. High-dimensional (9-dimensional embedding for 16ms; 16-dimensional embedding for 8, 16ms aux.) features are not well fused by the single linear layer, resulting in a limited performance improvement.

Effect of MI-L

In the MI-L method, a Linear Block is added after each Conv Block in the encoder for information fusion. The experimental results in Table 4.2 show that the best performance was obtained when using the “8, 16ms aux.”, while “8ms aux.” and “16ms aux.” had limited improvement for PESQ. When 8ms and 16ms spectrograms were used into the network as auxiliary information simultaneously, all evaluation measures were greatly improved. This shows that with layer-by-layer fusion the different spectral information was fused well.

Difference Between MI-F and MI-L

In both MI-F and MI-L, “8ms aux.” achieved better performance than “16ms aux.”. Compared with the 16ms spectrogram, the 8ms spectrogram has a larger difference from the 32ms spectrogram. Therefore, spectral information with larger differences is more effective. In addition, with sufficient fusion capability, more information can lead to better performance. MI-L outperforms MI-F on all evaluation measures. Besides, MI-F needs the fusion layer to have a smaller dimension, while MI-L needs the fusion layer to have a larger dimension. Furthermore, the auxiliary spectrogram of MI-F needs to be much different from the main enhanced spectrogram, while auxiliary spectrograms of MI-L are required to have more complete information.

Effect of Proposed Methods on Spectrogram

Fig. 4.11 shows the spectrograms of different SE systems. The main difference between these SE methods is the restoration of high frequencies and the processing of silent segments. Part A is a silent segment, “CRN” lost a lot of energy, while “MI-L” has better signal recovery. Furthermore, both “MI-F” and “MI-L” achieved recovery of sharper high-frequency detail. For Part B, “MI-F” and “MI-L” had better high-frequency recoveries than “CRN”. For Part C, some noise was not removed in all enhanced spectrograms, but “MI-L” contains less noise. We reason that the time-varying information provided by the wideband spectrogram helps

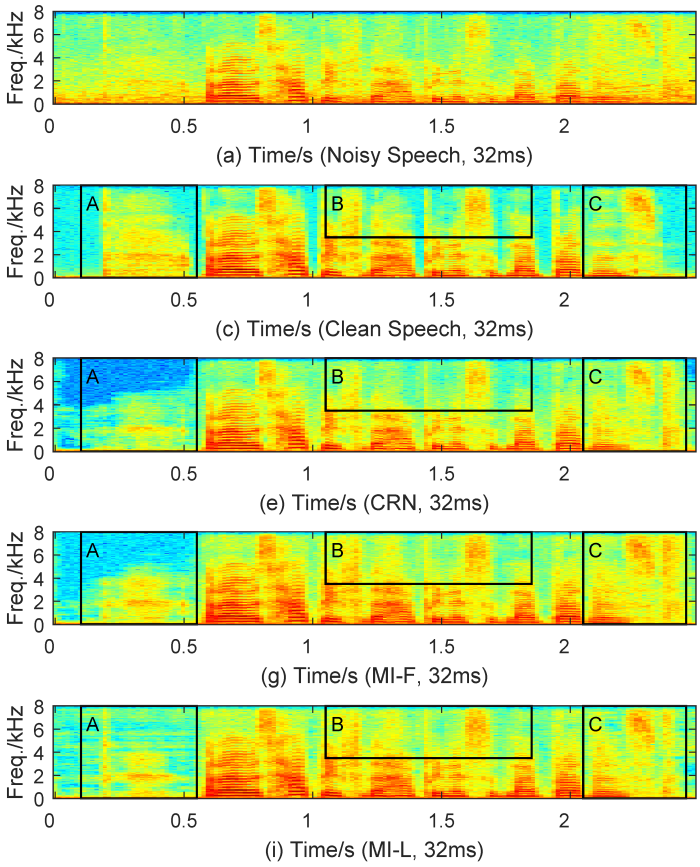


Fig. 4.11 Spectrograms of different SE systems.

narrowband spectrogram restoration. Furthermore, although the PESQ of "MI-F" was the same as that of "MI-L", there is still some information loss in "MI-F". Spectrograms with more bandwidth as input features help preserve spectral information.

4.4 Conclusion

In this chapter, we improved the SE with the input features. We first decomposed the spectrogram with a masking-based SE system. The feature maps are extracted with different threshold values in the mask. In this way, the components with high speech parts are obtained. The MM-SF system has better speech components and overall signal recovery. Besides, a proper decomposition number can bring better enhancement performance. Moreover, the proposed spectrogram decomposition helps the neural network by extracting feature maps with less noise and prominent speech components. The spectrogram can be divided into wideband and narrowband according to the resolution of the spectrogram. We improved the narrowband-based SE system with the wider bandwidth spectrograms as auxiliary infor-

mation. We propose multi-input final fusion (MI-F) and multi-input layer-by-layer fusion (MI-L) to incorporate information from different bandwidth spectrograms. MI-F adds a Linear Block only to the last layer of the encoder, while MI-L adds Linear Block after each Conv Block in the encoder for information fusion. With better fusion ability, MI-L achieves a better performance. Moreover, systems with larger differences in bandwidth achieve better performance. The proposed methods achieved better spectral recovery on silent segments and high-frequency spectrograms.

Chapter 5

Fusion of Spectrogram Features into Waveform-domain SE

5.1 Introduction

Waveform-domain SE [5, 6, 10] adopts speech waveform as input and output features. The magnitude and phase information is included in the waveform. With intensive studies, waveform-domain SE systems achieve state-of-the-art performance in many datasets [5, 6, 11]. However, it is often pointed out that the frequency-domain SE systems have more stable enhancement performance than waveform-domain SE systems [12] because the instability of the phase information [13] makes the waveform less stable than the frequency-domain magnitude of the spectrogram.

In order to improve the robustness of the waveform-domain SE method, we have proposed a waveform-spectrogram hybrid system (WaveSpecEnc) [94]. The proposed method complements waveform-domain DEMUCS [5] with the magnitude of spectrogram information. The waveform-spectrogram information fusion is done in the encoder. In each encoder layer, temporal and spectral information is first extracted by convolution processing at the utterance level. Then, the temporal feature maps are segmented and aligned with the spectral feature maps. The aligned spectral information is used to refine the segmental temporal information. The Hybrid DEMUCS [95] also integrates information on the temporal and spectrogram domains. The significant difference between the Hybrid DEMUCS [95] and our proposed WaveSpecEnc is that the Hybrid DEMUCS [95] employs shared encoder and decoder layers to process the information from different domains, while the WaveSpecEnc integrates the spectrogram information into the waveform encoding layer by layer.

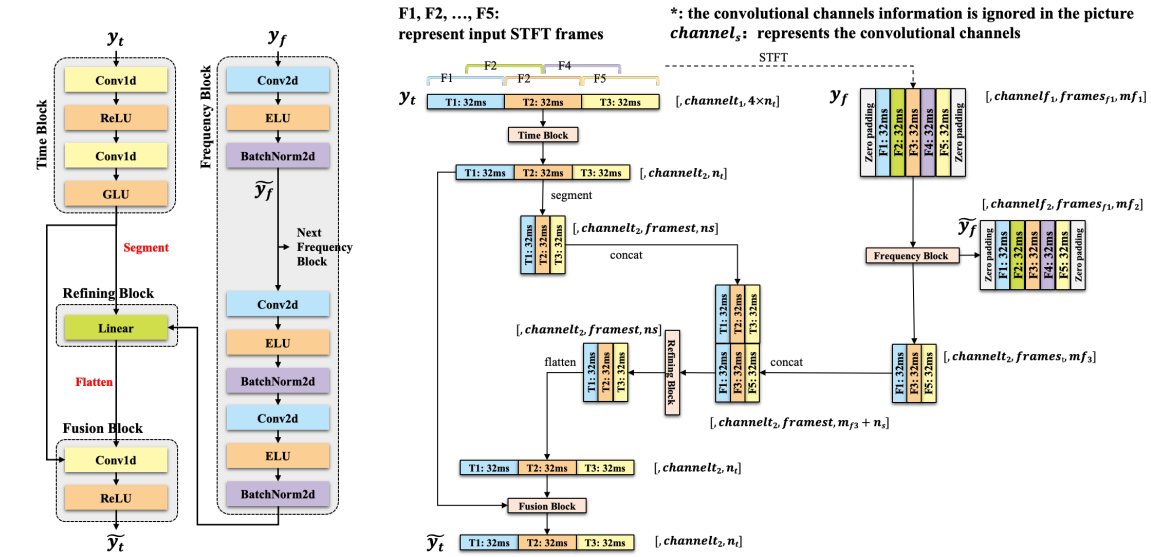
In addition to human hearing experience, improving automatic speech recognition (ASR) [96] in noisy conditions is crucial for the SE front-end. Previous works [97, 98] have found that information loss caused by the SE front-end affects the performance of ASR. To alleviate the problem, in this study, we improve the WaveSpecEnc by augmenting the encoding information of the ASR back-end with spectral information extracted in the SE module (WaveSpecEnc+). The enhanced spectral feature maps in the last layer in the WaveSpecEnc encoder are used to supplement the filter-bank (FBank) encoding in the ASR back-end. Different from previous works [97, 98], the enhanced waveform-domain and spectrogram-domain encodings are fused in this work instead of fusing the original noisy and enhanced spectrograms. Furthermore, previous work [99] has found that some speech information is highlighted after joint training, which means that spectral information useful for ASR is emphasized. In this manner, we aim to extract discriminative information from the enhanced spectral feature maps, which helps improve filterbank encoding performance in the ASR encoder. Compared to WaveSpecEnc [94], which only integrates spectrogram encoding information into the front-end's temporal information, WaveSpecEnc+ integrates spectrogram encoding information into both the front-end and the ASR back-end to enhance the performance of ASR.

5.2 Waveform-spectrogram Hybrid System

5.2.1 WaveSpecEnc SE Front-end

Although the performance of waveform-domain SE models has been improved, the instability of the phase information makes the waveform-domain representation less stable than the frequency representations. We propose a waveform-spectrogram Hybrid system (WaveSpecEnc) to address this problem. Specifically, we incorporate auxiliary frequency-domain information into waveform-domain features to improve the robustness. The magnitude of the spectrogram is adopted as frequency information.

The waveform-spectrogram information fusion is done in the encoder. Fig. 5.1–(a) shows the encoder layer of the proposed WaveSpecEnc. The waveform-domain feature maps are extracted by the same structure of **"Time Block"** in Eq. (2.13). The waveform-domain input from the previous WaveSpecEnc encoder layer or original waveform is denoted as $\boldsymbol{y_t}$. To get stable waveform-domain representations, the spectral features are used to refine the extracted feature maps by the "Time Block". $\boldsymbol{y_f}$ represents the spectral information from the previous **"Frequency Block"** or the magnitude of the spectrogram. Each "Frequency Block" stacks



(a) Encoder layer structure of the proposed method (b) Feature dimensions processed by different blocks in the proposed encoder layer

Fig. 5.1 (a) Encoder layer structure of the proposed waveform-spectrogram Hybrid system (WaveSpecEnc); (b) feature dimensions processed by different blocks in the proposed encoder layer: $\boldsymbol{y_t}$ represents the waveform-domain input from the previous WaveSpecEnc encoder layer or original waveform; $\boldsymbol{y_f}$ represents the spectral inputs from the previous frequency block or the magnitude of the spectrogram; $\widetilde{\boldsymbol{y}}_{\boldsymbol{f}}$ represent the spectral output (to the frequency block in the next encoder layer); $\widetilde{\boldsymbol{y}}_{\boldsymbol{t}}$ represents the final output (to the next encoder or LSTM layer).

"Conv_2d" layers:

$$BatchNorm2d(ELU(Conv2d(\cdot))). \tag{5.1}$$

With different kernel sizes, strides, and convolutional channels, the three "Conv_2d" layers have different purposes. The first "Conv_2d" layer enhances the spectral feature maps and keeps the feature frame the same as the original spectrogram. We denote the output of the first "Conv_2d" layer as $\widetilde{\boldsymbol{y}}_{\boldsymbol{f}}$, which serves as the input of the next "Frequency Block" in the next encoder layer and the second "Conv_2d" layer simultaneously.

However, it has different convolutional channels and frames from those of the temporal feature maps. Therefore, another two "Conv_2d" layers are introduced to extract deep encoded features with the same convolutional channels and frames as those of the temporal feature maps:

$$BatchNorm2d(ELU(Conv2d(BatchNorm2d(ELU(Conv2d(\cdot)))))). \tag{5.2}$$

For the number of layers to extract deep encoded features, we have tried to use one and three "Conv_2d" layers: the performance of one "Conv_2d" layer drops. The three "Conv_2d"

layers have the same performance as the two "Conv_2d" layers. Thus, we finally choose to use two "Conv_2d" layers.

After temporal and spectral information extraction, **"Refining Block"** is adopted to refine the temporal feature. The temporal feature is first segmented into 32-ms frames. The spectral feature is extracted with the same frames as the 32-ms segmented temporal feature. The "Refining Block" consists of one fully connected layer:

$$ReLU(linear(\cdot)). \tag{5.3}$$

Its input feature is the concatenation of 32-ms segmental waveform-domain and frequency-domain features. It maps the waveform-spectrogram hybrid feature maps into the refined temporal feature maps with the same dimensions as the waveform-domain features.

Finally, **"Fusion Block"** adopts one "Conv_1d" layer to fuse the original and refined feature maps. The output of each proposed encoder layer is represented as $\widetilde{\boldsymbol{y}}_{\boldsymbol{t}}$. Fig. 5.1–(b) shows more detailed illustration of these processes with the feature dimensions.

Other parts of the WaveSpecEnc are the same with DEMUCS [5]: two LSTM layers and a decoder with Eq. (2.14). Both the encoder and decoder contain five layers. The upsampling and downsampling processing are also included. The loss function is the same as Eq. (2.15).

It is also important to improve the performance of ASR with the SE front-end. We first try directly using WaveSpecEnc as the SE front-end for robust ASR, shown in Fig. 5.2–(a). The output of WaveSpecEnc is the waveform-domain speech waveform. The log Mel-filterbank (LMFB) is extracted from the enhanced waveform to input into the ASR back-end.

As the number of parameters in the end-to-end ASR back-end [100, 101] is usually much larger than that of the SE front-end, it is difficult to train or finetune it with a small amount of unseen noisy data. Therefore, we first pretrain the ASR back-end with a large amount of data. During joint training, the back-end parameters are frozen and only the front-end parameters are finetuned [14]. In this way, the plug-in front-end trained by a small amount of data can meet the needs of different robust noise conditions. The loss function for ASR ($\mathscr{L}_{Hybrid}$ with Eq. (2.24)) is adopted. This training scheme is same as the DEMUCS-based system.

5.2.2 WaveSpecEnc+ for Robust ASR

SE front-end often suppresses not only noise but also speech [97, 98]. This is good for human hearing but not for speech recognition. Some previous works [97, 98] fuse the original noisy spectral feature with the enhanced spectral feature to alleviate this drawback. However, using unprocessed noisy features will make it difficult for network learning. In this study, we can

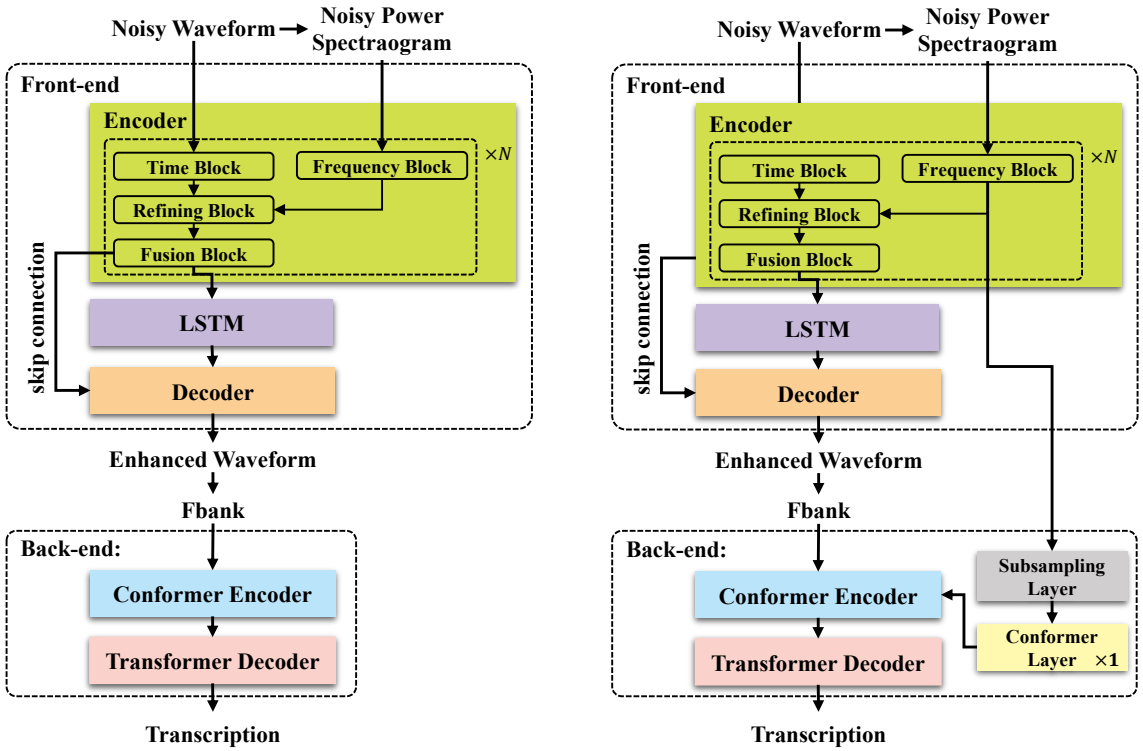


Fig. 5.2 Flowchart of different robust ASR systems: (a) ASR system with WaveSpecEnc front-end; (b) WaveSpecEnc+-based ASR system.

exploit or re-use the "Frequency Block" of WaveSpecEnc to augment the features for ASR. Fig. 5.2–(b) shows the flowchart of the WaveSpecEnc+-based robust ASR system.

The subsampling layer is adopted to subsample the output of the final "Frequency Block" in the front-end encoder, which ensures that the spectral information has the same frames as the feature in the ASR back-end. The Subsampling layer has the same neural network structure as the Subsampling layer in the ASR back-end: two Conv2d layers use a four-time subsampling rate. An additional Conformer layer encodes the spectral information with the attention mechanism. Finally, the fusion probabilities of two encoding information for each frame are obtained by a fully connected layer:

$$\mathbb{W} = ReLU(linear(\cdot)). \tag{5.4}$$

The LMFB encoding and spectral encoding are fused after the first Conformer layer in the encoder of the ASR back-end:

$$e_1^f = \mathbb{W}(e_1, s). \tag{5.5}$$

Table 5.1 (Unprocessed noisy data) Evalution metrics on real development and evaluation sets.

Condition	Real Development Set				Real Evaluation Set			
	SIG	OVRL	BAK	dMOS	SIG	OVRL	BAK	dMOS
BUS	1.4	1.2	1.3	2.7	1.6	1.3	1.3	2.4
STR	2.4	1.7	1.6	2.7	2.4	1.7	1.7	2.6
PED	3.0	2.1	2.1	2.9	2.2	1.5	1.5	2.4
CAF	2.5	1.6	1.6	2.7	2.1	1.4	1.4	2.5

s is the extracted spectral encoding information, and e_1 is the output of the first encoder layer of the ASR back-end. e_1^f is the fused feature, which is input to the second encoder layer of the ASR back-end. The ASR back-end is frozen during joint training using the loss function of ASR ($\mathscr{L}_{Hybrid}$ with Eq. (2.24)).

5.3 Experiments

5.3.1 Evaluations of Pre-trained Speech Enhancement Front-end

Experimental Settings

The experiments were conducted using the CHiME–4 dataset[1], which includes four noise conditions: bus (BUS), cafe (CAF), pedestrian area (PED), and street junction (STR). All data were digitized with 16K Hz sample rate. For SE front-end pre-training, the Channel 1 – Channel 6 simulated data from the training set were used; no development set was used during training. We have adopted the single-channel setting: the channel 5 data in the development and evaluation sets were used as the test sets. We tested in the following two scenarios.

Seen model: All noise conditions (BUS, CAF, PED, STR) were used in training as seen model.

Unseen model: We held out one noise condition to simulate the case of unseen scenario, that is we trained the model using three different noise scenarios and evaluated it in the remaining unseen noise scenario.

For baseline "Bi-LSTM" SE model, the input and output features were the magnitude of the spectrogram. The "Bi-LSTM" contained two Bi-directional LSTM (Bi-LSTM) layers and a fully connected layer. Each Bi-LSTM layer had 896 hidden nodes. For baseline

[1] https://spandh.dcs.shef.ac.uk/chime_challenge/CHiME4/index.html

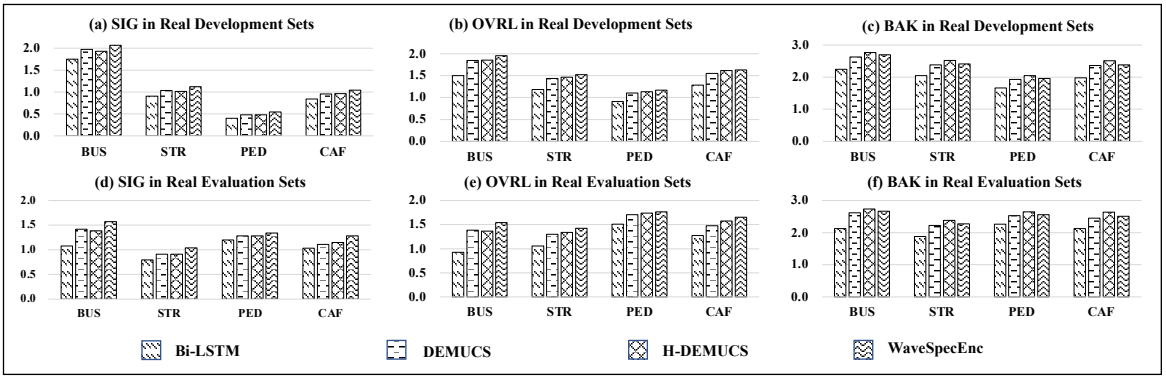


Fig. 5.3 (Seen) Relatively improvement of SIG / OVRL / BAK values (↑) compared with non-enhanced signals (Table 5.1) in real development and evaluation sets. All noise conditions are SEEN to the model.

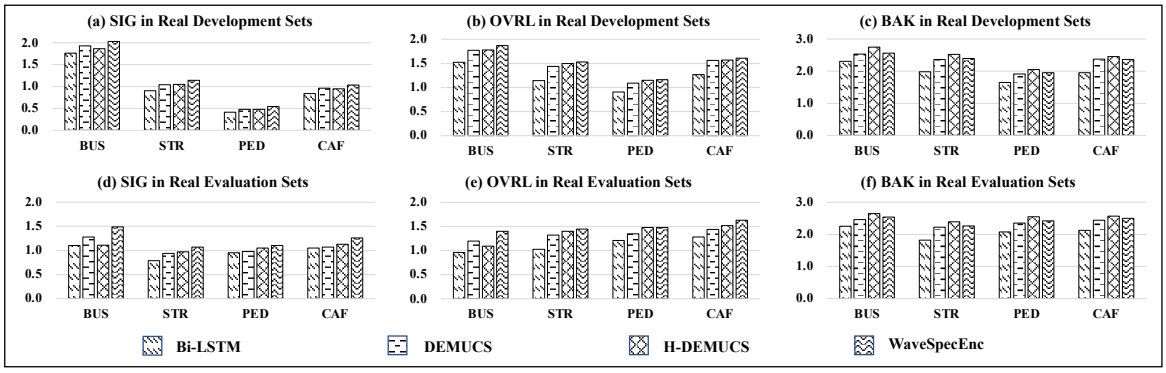


Fig. 5.4 (Unseen) Relatively improvement of SIG / OVRL / BAK values (↑) compared with non-enhanced signals (Table 5.1) in real development and evaluation sets. The test noise conditions are UNSEEN to the model.

"DEMUCS" and the proposed "WaveSpecEnc", the channels in different depths were {1, 48, 96, 192, 384, 768}. Each LSTM layer contained 768 nodes. In each "Time Block", the kernel size and stride for the two "Conv_1d" layers were {8, 1}, and {4, 1}, respectively. Each decoder layer's kernel size and stride of the "Conv_1d" and "DeConv_1d" layers were {1, 8}, and {1, 4}, respectively. For the first "Conv_2d" layer in each "Alignment Block", the kernel size and stride were 3 and 2, respectively. For the second "Conv_2d" layer in each "Alignment Block", the kernel size and stride were 3 and 1, respectively. The input and output dimensions of "Refining Block" were {640, 191, 63, 23, 9} and {512, 128, 32, 8, 2}. The input and output channels of "Fusion Block" were {96, 192, 384, 768, 1536} and {48, 96, 192, 384, 768}. The kernel size of Conv1d in "Fusion Block" is 1. For extracting the spectrogram, the STFT points were 32-ms; the Hanning window was used; the STFT hop

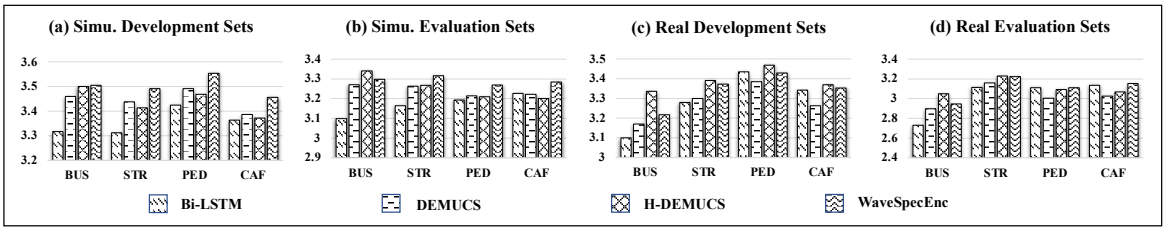


Fig. 5.5 (Seen) dMOS values (↑) in simulated and real sets. All noise conditions are SEEN to the model.

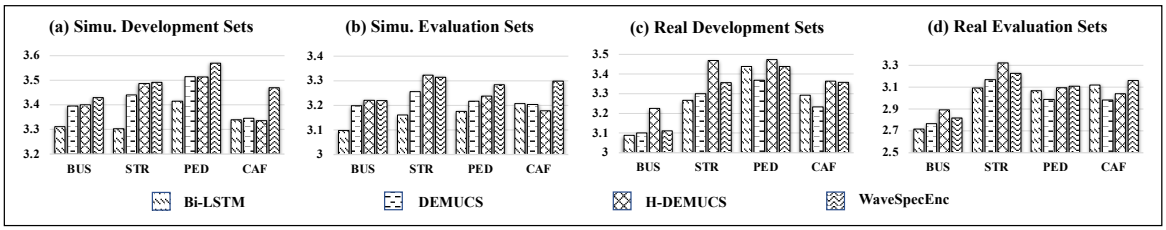


Fig. 5.6 (Unseen) dMOS values (↑) in simulated and real sets. The test noise conditions are UNSEEN to the model.

length was 16-ms. The hyperparameter α in Eq. (2.15) was set to 0.5. We also compared with "Hybrid DEMUCS (H-DEMUCS)" [95] by following the official source code[2].

All neural networks were implemented with PyTorch. All SE front-ends were trained with 200 epochs.

We used multiple linear regression analysis to form the following composite measures: signal distortion (SIG) [77], background intrusiveness (BAK) [77], overall quality (OVL) [77], the subjective Mean Opinion Score (dMOS). All of them are evaluated by open-source toolkit DNSMOS [102, 103], which is widely used in Deep Noise Suppression (DNS) challenge[3]. Table 5.1 shows values of these metrics of the unprocessed noisy development and evaluation sets.

Comparison of SE Systems in Different Domains

Fig. 5.3 and Fig. 5.4 show the SIG, OVRL, and BAK values of different SE systems in real development and evaluation sets in seen scenario and unseen scenario, respectively. "DEMUCS" outperforms "Bi-LSTM" on almost all noise conditions. It achieved better speech signal recovery, overall quality recovery, and noise suppression. SIG and OVRL were affected by noise conditions, especially on the evaluation set. The performance of most

[2]https://github.com/facebookresearch/demucs/blob/main/demucs/hdemucs.py

[3]https://www.microsoft.com/en-us/research/academic-program/deep-noise-suppression-challenge-icassp-2023/

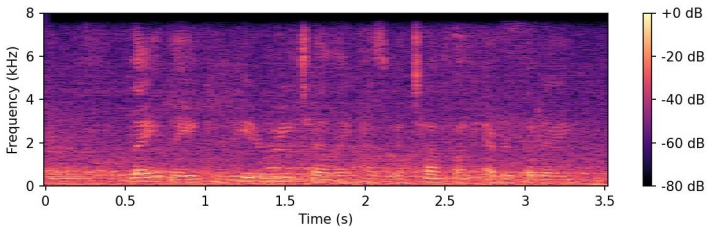
evaluation metrics is degraded under unseen conditions: the BUS noise condition was the most challenging.

Although "DEMUCS" achieves better speech signal recovery (SIG), overall quality recovery (OVRL), and noise suppression (BAK) than "Bi-LSTM", it is not as good at the dMOS values. Fig. 5.5 and Fig. 5.6 show the dMOS values in development and evaluation sets in seen scenario and unseen scenario, respectively. The waveform-domain SE system is sensitive to the noise conditions. This trend is not obvious in the simulated data sets but is evident in the real data sets. For the simulated noisy sets, "DEMUCS" outperforms "Bi-LSTM" in almost all noise conditions. Its superiority is almost diminished for the real noisy sets. "DEMUCS" shows a significant degradation for all PED and CAF noise conditions. This may be due to the large difference in data distribution between the training set and the evaluation sets in these two noise conditions. The frequency-domain model showed robustness against unseen noise conditions.

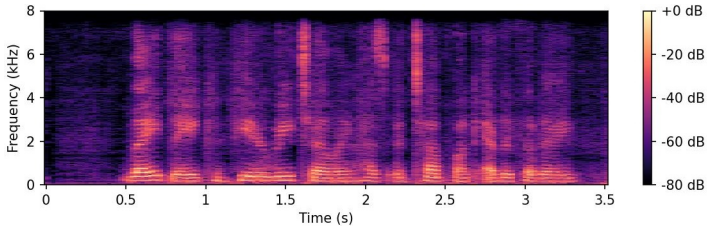
The reason for the dMOS degradation of "DEMUCS" may be because of the artificial noise. Fig. 5.7 shows the magnitude of the spectrogram enhanced by different SE systems. The spectrogram enhanced by "Bi-LSTM" still contains much noise. Although the low-frequency speech signal recovery quality of the "DEMUCS" is higher than the "Bi-LSTM", the high-frequency part introduces noticeable artificial noise.

Effect of Spectrogram Encoding

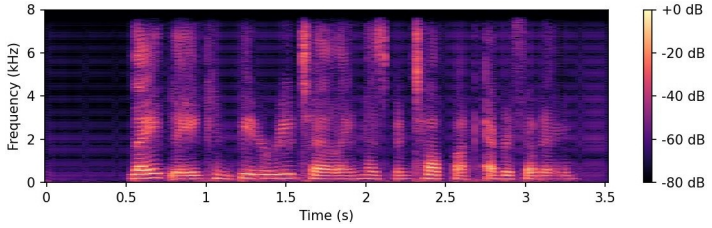
The proposed WaveSpecEnc system combines the advantages of waveform-domain and frequency-domain SE systems. In Fig. 5.3 and Fig. 5.4, "WaveSpecEnc" outperforms "Bi-LSTM" on all SIG, OVRL, and BAK evaluation metrics. Compared to "DEMUCS", the proposed system further improves SIG and OVRL by introducing spectral information. The proposed system had a slight improvement on BAK compared to "DEMUCS". For the dMOS value in Fig. 5.5 and Fig. 5.6, the proposed "WaveSpecEnc" performed best in all simulated noise conditions compared to "Bi-LSTM" and "DEMUCS". For real noisy conditions in development and evaluation sets, although the method proposed had slightly worse than "Bi-LSTM" in the PED noise condition, there were still large improvements from the waveform-domain "DEMUCS". This shows that incorporating spectral information into the waveform-domain SE system can improve the stability of the waveform-domain SE system. Furthermore, the proposed method had a similar low-frequency restoration ability with "DEMUCS", which is shown in Fig. 5.7. The difference between the middle and high frequencies was more noticeable, especially the high-frequency artificial noise introduced by "DEMUCS" was significantly suppressed. This indicates that spectrogram encoding helps to reduce the introduction of artificial noise.



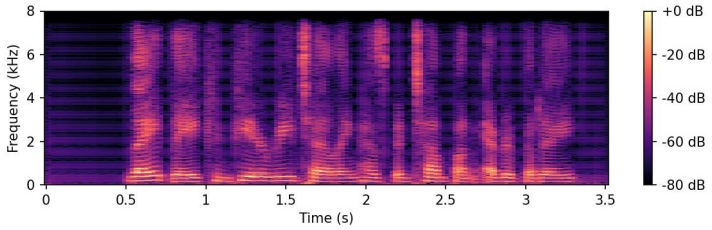
(a) Noisy



(b) Bi-LSTM



(c) DEMUCS



(d) WaveSpecEnc

Fig. 5.7 Enhanced magnitude spectrograms of the pre-trained SE front-end. The clip is a real noisy speech under PED noise condition: (a) Noisy, (b) Bi-LSTM enhanced, (c) DEMUCS enhanced, and (d) WaveSpecEnc enhanced.

Different ways of combining spectral and temporal information show varying performances on different evaluation metrics. Compared to "H-DEMUCS", "WaveSpecEnc" exhibited better speech signal restoration and overall quality restoration (SIG, OVRL), while "H-DEMUCS" showed better noise reduction ability (BAK). These two methods have their respective strengths and weaknesses in terms of dMOS improvement. It shows some complementarity between different fusion methods.

5.3.2 Evaluations of SE-based Adaptation for Noise-mismatched ASR Back-end

Compared with the SE front-end, the ASR back-end has much more parameters, and thus the necessary amount of the training data for the ASR back-end is usually far more than that of the SE front-end. Moreover, in many practical applications, it is often not allowed to finetune the ASR back-end but only possible to tune the SE front-end. In this Section, we first investigate an effective adaptation way to finetune the SE front-end only by freezing the ASR back-end when encountering a new noise scene. ASR performance can be improved by finetuning the SE front-end by propagating the ASR loss [14].

Experimental Settings

For the Conformer-based ASR back-end, the number of encoder layers was 6. In each encoder layer, the positional encoding layer type was the relative positional encoding module; the subsampling rate was 4 with 2 Conv2d layers; the dimension of multi-head attention was 512; the number of attention heads was 4; the number of units of position-wise feedforward was 2048; the activation function was swish; the dimension of the input LMFB was 80. The decoder was based on Transformer [100, 104]. The number of decoder layers was 6. In each decoder layer, the dimension of multi-head attention was 512; the number of attention heads was 4; the hidden units number of position-wise feedforward was 2048. We used all transcripts of CHiME-4, WSJ0 [4] and WSJ1 [5] to define a dictionary. The size of BPE vocabulary was 1014 including the $< blank >$, $< unk >$, and $< sos/eos >$.

Pretrained ASR back-end: WSJ0, WSJ1, and Librispeech (960 hours) [105] were used for ASR back-end pre-training. When pre-training, noises from the MUSAN dataset [106] were synthesized with clean speech. The signal-to-noise ratio (SNR) was randomly selected between 0 and 20 dB. The pre-trained ASR back-end was trained with 100 epochs. The SpecAug [107] was applied when training. Ten checkpoints that performed well on the development set were averaged as the final pre-trained model. The hyperparameter β in Eq. (2.24) was set to 0.7.

Freezing ASR back-end and finetuning SE front-end: During joint training, the pre-trained ASR back-end was frozen. This joint training was performed under the same training condition with CHiME–4 training data of front-end training. The development set of the corresponding noise conditions of CHiME–4 were used to select the model according to the minimum loss. All ASR systems were trained with 70 epochs.

[4] https://catalog.ldc.upenn.edu/LDC93s6a
[5] https://catalog.ldc.upenn.edu/LDC94S13A

Table 5.2 (Seen) Word Error Rate (%, ↓) in real development and evaluation sets. All noise conditions are seen to the model. FT represents whether the front-end has been finetuned. The back-end is not finetuned in this experiment.

Systems	Seen	FT	BUS	STR	PED	CAF	AVG
Real Development							
Conformer (pretrained)	-	✗	18.3	10.8	9.1	12.7	12.7
Bi-LSTM	✓	✗	27.9	19.6	15.5	22.1	21.3
DEMUCS	✓	✗	24.4	18.4	14.0	20.1	19.2
H-DEMUCS	✓	✗	28.8	20.7	16.6	19.8	21.5
WaveSpecEnc	✓	✗	21.9	14.0	12.4	17.1	16.3
Bi-LSTM	✓	✓	13.3	8.1	6.8	8.4	9.2
DEMUCS	✓	✓	11.7	7.7	6.3	6.9	8.2
H-DEMUCS	✓	✓	12.2	7.7	6.7	7.3	8.5
WaveSpecEnc	✓	✓	10.8	7.3	6.3	6.7	7.8
WaveSpecEnc+	✓	✓	**10.4**	**7.1**	**6.1**	**6.3**	**7.5**
Real Evaluation							
Conformer (pretrained)	-	✗	27.0	14.1	19.6	22.4	20.8
Bi-LSTM	✓	✗	61.9	26.3	34.9	40.6	40.9
DEMUCS	✓	✗	44.8	24.1	36.1	41.8	36.7
H-DEMUCS	✓	✗	51.9	26.7	35.0	38.2	37.9
WaveSpecEnc	✓	✗	45.6	23.7	37.4	37.7	36.1
Bi-LSTM	✓	✓	21.4	10.1	13.5	16.1	15.3
DEMUCS	✓	✓	19.2	9.2	12.8	14.9	14.0
H-DEMUCS	✓	✓	20.9	10.7	13.9	15.2	15.2
WaveSpecEnc	✓	✓	18.1	9.4	12.6	13.9	13.5
WaveSpecEnc+	✓	✓	**17.0**	**8.6**	**11.8**	**13.3**	**12.7**

Word error rate (WER) was used to evaluate the ASR performance.

Evaluation in ASR

Table 5.2 and Table 5.3 show the WER in real development and evaluation sets. Directly using a cascade system (the upper half of the real development and real evaluation sets in Table 5.2) built with the pre-trained SE front-end and ASR back-end significantly degraded the recognition performance, because the test noise in CHiME–4 dataset significantly differs from the noise used for pre-training SE front-end and the ASR back-end. The joint training (the lower half of the real development and real evaluation sets in Table 5.2) significantly

Table 5.3 (Unseen) Word Error Rate (%, ↓) in real development and evaluation sets. The test noise conditions are unseen to the model. Compared with the seen results in Table 5.2, the relative decrease percentage of WER under the unseen testing (Decrease). FT represents whether the front-end has been finetuned. The back-end is not finetuned in this experiment.

Systems	**Seen**	**FT**	**BUS**	**STR**	**PED**	**CAF**	**AVG**	**Decrease**
Real Development								
Conformer	-	✗	18.3	10.8	9.1	12.7	12.7	-
Bi-LSTM	✗	✓	14.0	8.5	7.3	8.8	9.7	5.4%
DEMUCS	✗	✓	14.4	7.9	6.6	7.1	9.0	9.8%
H-DEMUCS	✗	✓	15.0	7.7	6.8	7.4	9.2	8.2%
WaveSpecEnc	✗	✓	14.0	7.6	6.2	7.4	8.8	12.8%
WaveSpecEnc+	✗	✓	**13.3**	**7.0**	**6.0**	**6.6**	**8.2**	9.3%
Real Evaluation								
Conformer	-	✗	27.0	14.1	19.6	22.4	20.8	-
Bi-LSTM	✗	✓	24.7	10.7	13.7	15.9	16.2	5.9%
DEMUCS	✗	✓	26.0	10.2	13.3	15.4	16.2	15.7%
H-DEMUCS	✗	✓	30.3	10.1	13.8	14.6	17.2	13.2%
WaveSpecEnc	✗	✓	29.7	9.9	13.0	14.2	16.7	23.7%
WaveSpecEnc+	✗	✓	**23.0**	**9.0**	**11.8**	**13.7**	**14.4**	13.4%

improved recognition performance. "DEMUCS"-based front-end performs better than the "Bi-LSTM"-based front-end in almost all noise conditions in ASR, although it was not good at human hearing experiences under the PED and CAF noise conditions, which is evident in real evaluation sets. The performance of the ASR system is degraded largely when tested under unseen conditions as shown in Table 5.3. It shows that the degradation of "DEMUCS" is much larger than "Bi-LSTM". In particular, when the BUS noise data is not involved in the training, the "DEMUCS" had a significant performance degradation. This is the case with the proposed "WaveSpecEnc", although spectrogram encoding gives a considerable performance improvement in other noise conditions. This may be because the BUS noise condition was the most adversary as shown in Table 5.1. Unexpectedly, while "H-DEMUCS" greatly enhanced the dMOS, it did not bring strong performance for ASR.

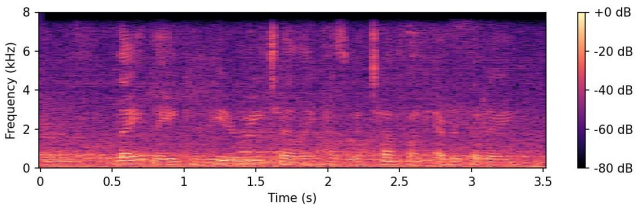
Incorporating spectrogram encoding information into the ASR back-end, "WaveSpecEnc+", significantly and consistently improved ASR performance under all noise conditions. It is also effective in the most challenging BUS condition. This result confirms that incorporating spectrogram encoding not only in the SE front-end but also ASR back-end is crucial. This proposed method significantly outperformed all other methods (p-value < 0.01),

Table 5.4 Number of Conformer Layers with Spectrogram Encoding (Num.). All noise conditions are seen to the model.

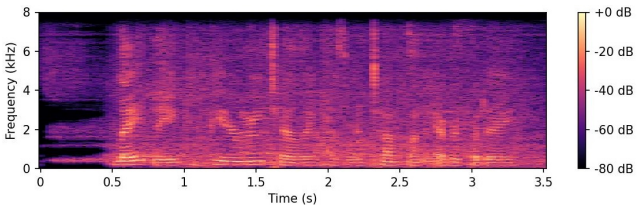
Num.	Fusion Layers 1	2	3	4	5	6	BUS	STR	PED	CAF	AVG
Development											
1	✓						10.4	7.1	6.1	6.3	7.5
2	✓	✓					10.6	6.8	6.0	6.2	7.4
3	✓	✓	✓				10.8	6.8	6.1	6.5	7.5
4	✓	✓	✓	✓			10.3	6.9	6.2	6.1	7.4
5	✓	✓	✓	✓	✓		10.3	6.8	6.1	6.1	7.3
6	✓	✓	✓	✓	✓	✓	10.8	7.0	6.4	6.7	7.7
Evaluation											
1	✓						17.0	8.6	11.8	13.3	12.7
2	✓	✓					17.7	9.0	12.0	13.3	13.0
3	✓	✓	✓				18.2	8.9	11.9	13.5	13.1
4	✓	✓	✓	✓			17.8	8.7	11.8	13.4	12.9
5	✓	✓	✓	✓	✓		17.7	8.8	11.8	13.2	12.9
6	✓	✓	✓	✓	✓	✓	18.0	9.0	12.2	13.7	13.2

although the baseline "Bi-LSTM" shows better robustness (least decrease from the seen condition).

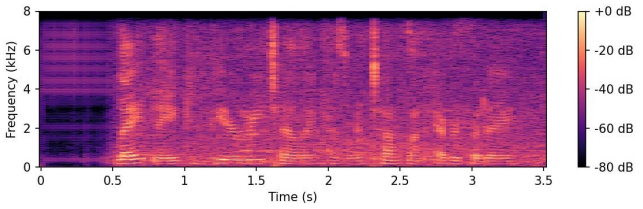
Fig. 5.8 shows the enhanced magnitude spectrograms of different SE front-ends after joint training. The front-end output after joint training is similar to the noisy spectrogram in the speech parts, but the energy of some speech information is more prominent between the noisy and enhanced features. This shows that the front-end SE with joint training preserves the speech signal as much as possible while highlighting the effective ASR-related speech components. Compared with the other enhancement front-ends, the speech components are not highlighted in the "Bi-LSTM" spectrogram. Moreover, some high-frequency information is blurred. The spectrogram of "DEMUCS" introduces artificial noise in the high-frequency parts. "WaveSpecEnc" has some noise reduction effect but "WaveSpecEnc+" has better noise reduction. This is because the joint training keeps the information of the spectrogram encoding intact, and removes adversary artificial noise. We also conducted the dMOS evaluation for the finetuned SE front-end. The evaluation showed that the finetuned SE front-end considerably decreased performance compared with the pre-trained front-end model.



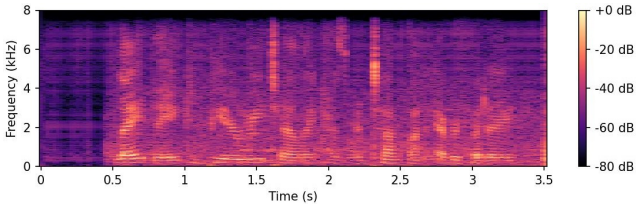
(a) Noisy



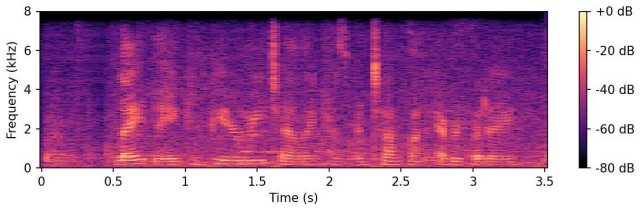
(b) Bi-LSTM



(c) DEMUCS



(d) WaveSpecEnc



(e) WaveSpecEnc+

Fig. 5.8 Enhanced magnitude spectrograms of SE front-end after joint training. The clip is a real noisy speech under PED noise condition: (a) Noisy, (b) Bi-LSTM enhanced, (c) DEMUCS enhanced, (d) WaveSpecEnc enhanced, (e) WaveSpecEnc+ enhanced.

Effect of Fusion Layers in ASR Back-end

"WaveSpecEnc+" incorporates the spectrogram encoding in the first encoder layers of the ASR back-end. The layer-by-layer fusion was compared in Table 5.4. Fusion in many layers

is not so effective for improving ASR performance. Despite the recognition performance improvements observed in all models, deep-level incorporation of the spectrogram encoding did not yield more noticeable performance gains. Instead, the most significant performance improvement was obtained when incorporating the spectrogram encoding at the shallow layer. As the encoder layers in the end-to-end ASR model become deeper, the content within these layers tends to extract semantic information. In contrast, the shallower layers contain mostly environmental and noise-related details. Fusing in the shallower layers shows more effective.

5.3.3 Evaluations of Finetuning Both SE Front-end and ASR Back-end

Finetuning the ASR back-end using data from a new noise environment is a direct and effective adaptation method. In this Section, we simultaneously finetune both the SE front-end and ASR back-end using the CHiME-4 dataset. It should be noted that this is possible when a large dataset is available. "Conv-Tasnet" [108] is also compared as the SE front-end, which was pretrained with the simulated CHiME–4 data with 100 epochs. The hyperparameter settings were the same as those of ESPnet [6].

Experimental Settings

We tried two types of acoustic models. The first one was the same as the pre-trained acoustic model in Chapter 5.3.2: the Conformer pre-trained with Librispeech-960 and MUSAN noise. The second one adopted WavLM as the acoustic model.

Finetuning ASR and SE same as in Chapter 5.3.2: We conducted joint training, in which the SE front-end and the ASR back-end parameters were finetuned using the CHiME–4 dataset. All simulated and real data from the training set were used.

Moreover, we also incorporated the language model (LM) to further improve the ASR performance. We utilized the transformer-based LM. It contained 16 encoder layers. In each encoder layer, there was no positional encoding layer; the dimension of multi-head attention was 512; the number of attention heads was 8; the number of units of position-wise feedforward was 2048. The training text consisted of two parts: the first part included text data extracted from the CHiME-4 training set; the second part was obtained from the "wsj1_lng", totaling approximately 1.7 million text samples. Shallow fusion was adopted to integrate the LM and acoustic model with a fusion weight of 0.6 and 0.4.

[6] https://github.com/espnet/espnet/blob/master/egs2/chime4/enh1/conf/tuning/train_enh_conv_tasnet.yaml

Table 5.5 Word Error Rate (%, ↓) in real development and evaluation sets. All noise conditions are seen to the model. LM denotes whether to use the external language model.

Systems	LM	BUS	STR	PED	CAF	AVG
Real Development						
Conformer (pretrained)	-	18.3	10.8	9.1	12.7	12.7
Finetuned	-	12.0	10.0	9.0	10.8	10.5
Finetuned	✓	9.3	7.8	7.4	8.7	8.3
Bi-LSTM	✓	8.6	8.2	7.8	9.3	8.5
Conv-Tasnet	✓	7.4	6.9	6.7	7.0	7.0
DEMUCS	✓	7.8	7.4	6.7	7.3	7.3
H-DEMUCS	✓	7.9	7.2	6.7	7.4	7.3
WaveSpecEnc	✓	7.6	7.3	6.7	7.4	7.3
WaveSpecEnc+	✓	7.6	7.4	7.1	7.5	7.4
Real Evaluation						
Conformer (pretrained)	-	27.0	14.1	19.6	22.4	20.8
Finetuned	-	18.5	10.9	14.4	15.8	14.9
Finetuned	✓	13.9	9.2	11.2	12.6	11.7
Bi-LSTM	✓	12.6	8.8	9.9	10.7	10.5
Conv-Tasnet	✓	11.0	8.3	8.5	10.1	9.5
DEMUCS	✓	11.3	8.0	8.6	10.4	9.6
H-DEMUCS	✓	11.6	8.0	8.6	9.7	9.5
WaveSpecEnc	✓	10.5	8.2	8.6	9.6	9.2
WaveSpecEnc+	✓	**10.3**	**7.5**	**7.9**	**9.2**	**8.7**

Finetuning WavLM and SE: We used the pretrained model checkpoint available on HuggingFace [7], which consists of 24 layers of Transformer architecture. We used the character-based dictionary. All parameters were finetuned with the CTC loss. The WaveSpecEnc+ spectrogram encoding is approximately 1.25 times the number of feature frames extracted by the WavLM feature extraction module. Thus, the features need to be time-aligned. We simply drop the sixth frame after every five consecutive frames of the spectrogram encoding.

Evaluation in ASR

Table 5.5 shows the ASR performance for real sets. Finetuning the ASR back-end using the CHiME–4 dataset significantly enhanced ASR performance. Additionally, incorporating an

[7] https://huggingface.co/microsoft/wavlm-large

Table 5.6 Word Error Rate (%, ↓) in real development and evaluation sets. All noise conditions are seen to the model. WavLM is adopted as the acoustic model.

Systems	BUS	STR	PED	CAF	AVG
Real Development					
WavLM	6.3	5.4	4.6	5.2	5.4
Bi-LSTM	5.8	4.3	3.6	4.5	4.5
Conv-Tasnet	6.1	4.3	3.9	4.2	4.6
DEMUCS	5.6	4.8	3.6	4.2	4.5
H-DEMUCS	5.8	4.2	3.6	4.0	4.4
WaveSpecEnc	5.5	4.8	3.5	4.1	4.5
WaveSpecEnc+	5.6	4.5	3.5	4.0	4.4
Real Evaluation					
WavLM	8.2	5.8	6.7	6.6	6.8
Bi-LSTM	7.7	4.5	5.9	6.1	6.0
Conv-Tasnet	8.0	5.0	6.0	6.0	6.2
DEMUCS	7.8	4.6	5.7	5.8	6.0
H-DEMUCS	7.7	4.8	5.9	5.9	6.1
WaveSpecEnc	7.4	4.8	6.2	5.6	6.0
WaveSpecEnc+	7.1	4.5	5.8	5.8	5.8

additional LM further improved the performance. Therefore, in subsequent experiments, we employed LM during decoding.

From this new baseline, the "Bi-LSTM"-based system did not show significant improvement for the evaluation set. "DEMUCS" showed a notable improvement. Compared to finetuning the SE front-end only, jointly optimizing the ASR back-end and the SE front-end led to large performance improvements for "H-DEMUCS". While "WaveSpecEnc" slightly performs better than these, "WaveSpecEnc+" significantly outperformed "DEMUCS" and "H-DEMUCS" in the real evaluation set (p-value < 0.01) since it integrates effective information into the ASR back-end. In addition, we also compared "Conv-Tasnet"-based ASR system. The "Conv-Tasnet" system showed slightly better performance in real development sets (p-value > 0.05), but note that the ASR model was selected according to the checkpoints in the development set. On the other hand, the proposed "WaveSpecEnc+" significantly outperformed "Conv-Tasnet" in real evaluation sets (p-value < 0.01).

Table 5.6 shows the ASR performance with the WavLM-based acoustic model. Pretraining based on self-supervised learning significantly improved the ASR performance. The SE front-ends still significantly improved the performance of ASR. Although the performance

Table 5.7 Comparison between different single channel automatic speech recognition systems (Word Error Rate, %, ↓).

Systems	SSL	Dev. Set		Eval. Set	
		Simu.	Real	Simu.	Real
DNN-HMM Hybrid ASR					
Kaldi [109]	✗	6.8	5.6	12.2	11.4
Yang *et al.* [110]	✗	5.0	3.4	8.6	6.3
Wang *et al.* [111]	✗	5.0	3.5	9.4	6.8
End-to-End ASR					
ESPnet (Conformer)	✗	11.3	9.2	16.8	15.9
IFF-Net [98]	✗	7.9	6.4	13.4	12.4
DPSL-ASR [112]	✗	7.2	5.9	12.2	11.3
WaveSpecEnc+ (this study)	✗	7.3	7.4	12.0	8.7
Transformer - HuBERT [113]	✓	11.6	9.1	18.0	20.4
Transformer - WavLM [113]	✓	5.9	4.0	8.3	4.5
IRIS [113]	✓	3.2	2.0	6.1	3.9
WaveSpecEnc+ - WavLM (this study)	✓	3.0	4.4	5.7	5.8

difference among the SE methods is small, the proposed "WaveSpecEnc+" resulted in the best performance in the real evaluation set.

Comparison Between Different ASR Systems

Table 5.7 lists the performance of different ASR systems under single-channel conditions for the CHiME–4 evaluations. DNN-HMM Hybrid ASR systems perform better with a small amount of data than end-to-end ASR systems. We expect the end-to-end model to perform better with a large amount of training data. Pretraining based on self-supervised learning solves this problem. Particularly, noise-aware pretrained model, such as WavLM, is effective. We have demonstrated the effectiveness of the proposed method (WaveSpecEnc+) in this setting as well. While our ASR back-end is based on simple CTC in Table 5.6, "IRIS" and "Transformer - WavLM" used WavLM as a feature extractor and incorporated additional Transformer layers for ASR. They also adopted an external LM. For fair comparison, we also conducted an experiment based on the IRIS pipeline and replaced IRIS's ConvTasnet with the proposed front-end. Experimental results confirm that the system based on the proposed WaveSpecEnc+ performs better for the real evaluation set, though there is no significant

difference for all test sets. Although the ASR performance is almost saturated with the strong ASR back-end, the effect of the proposed front-end was more clearly observed with the lightweight ASR back-end in Table 5.5 and 5.6.

5.4 Conclusion

In this chapter, we improve the robustness of waveform-domain speech enhancement SE with spectrogram encoding ("WaveSpecEnc"). The temporal feature maps at each encoder layer in the SE front-end are refined by spectral information. The proposed time-spectrogram hybrid system improved the dMOS score. Artificial noise introduced by the waveform-domain SE front-end can be reduced by the using of spectrogram-domain information. However, "WaveSpecEnc"-based ASR system had minor improvement over the "DEMUCS"-based ASR system. Thus, we incorporate the spectral information of the encoder layer into the ASR back-end ("WaveSpecEnc+"). Compared with "DEMUCS", "WaveSpecEnc+" significantly improved ASR performance in all noise conditions on CHiME–4 evaluation sets. Several acoustic models were used to evaluate the effectiveness of "WaveSpecEnc+"-based ASR systems. Firstly, the experimental results with a frozen pre-trained acoustic model showed that incorporating spectrogram encoding in the ASR back-end is crucial. It is effective to fuse features in only shallow encoder layers of the Conformer-based ASR system. Secondly, the SE front-end and the pre-trained acoustic model were jointly finetuned with the CHiME–4 training set. The experimental results showed that integrating the spectral encoding into the ASR back-end is still effective. Thirdly, we also tried WavLM as the acoustic model. The experimental results showed that the SE front-ends still improved the ASR performance, although the performance differences among the SE front-ends were small. Finally, we replaced the "Conv-Tasnet" in the IRIS system with our proposed "WaveSpecEnc+". Experimental results confirm that the system based on the "WaveSpecEnc+" performs better for the real evaluation set.

Chapter 6

Fusion of Different Feature Extraction Modules for Adapter-based ASR

6.1 Introduction

Deep learning-based ASR models [100, 114], however proficient, exhibit susceptibility to performance deterioration when exposed to unseen noise sources. In today's large-scale models [45, 47], the imperative to adapt models to new noisy conditions with minimal data becomes crucial. Domain adaptation [115] in ASR involves finetuning a model from one (source) domain to work effectively in another (target) domain, managing challenges arising from distribution disparities. Data augmentation [116, 99, 117], transfer learning [118, 119], domain adversarial training [120, 121], and multi-task learning [122] have shown beneficial for adapting the model to specific tasks.

Compared with the SE front-end, the ASR back-end has much more parameters. That is the reason we tried to only finetune the SE front-end in Chapter 5.3.2 [123].

Adapter, which belongs to transfer learning [124], has been very popular recently. It has been applied to various ASR tasks, empowering models to adapt efficiently to specific challenges. Adapters serve as a crucial bridge in addressing various low-resource ASR tasks [115, 52, 125–129]. A primary application domain of adapters lies in accent and dialect tasks [52, 125, 126]. Moreover, they have shown effectiveness in the disorder and children ASR [115, 127, 128]. Furthermore, when applying the pretrained model [45, 47], how to better transfer the knowledge in the large model to new scenarios with limited data is particularly important. Because the adapter meets such requirements, inserting the adapter into the pretrained large model has also been widely studied [115, 124, 126, 130]. Despite the widespread application of adapters across various ASR tasks, limited investigation has been

conducted towards noise-robust ASR. In this work, we explore adapter-based noise-robust ASR, considering a range of viewpoints. Our primary focus encompasses investigating the adapter insertion points, the data employed for adapter training, and the synergy with speech enhancement (SE) front-end models.

Self-supervised learning (SSL) [131–133, 45] is a machine learning approach in which a model discerns patterns from unlabeled data by autonomously creating supervisory signals or labels. Compared to the conventional supervised learning that relies on human-annotated data for training [100], SSL extracts features and encoded representations with massive unlabeled data for subsequent tasks [134, 135]. SSL greatly improves the performance of automatic speech recognition (ASR) systems [45, 47, 136, 137]. The common SSL model for ASR [45, 47, 48] contains a feature extraction (FE) module and a Transformer encoder [100]. The unsupervised FE module is trained well and universally with massive data training. Thus, it is common to freeze the parameters of the FE [124, 138] and finetune the following Transformer layers only.

Although this finetuing method benefits many speech-oriented tasks [134, 135], two mismatches emerge in the FE when targeting noise-robust ASR. The primary mismatch arises because the pretraining data of the FE is largely based on clean and simulated noisy speech. In contrast, the main application needs to tackle with real noisy speech [45, 47]. The second mismatch is caused by the divergent data distribution between clean and noisy speech [47]. While some pretraining datasets incorporate noisy speech, noise in unsupervised training may lead to erroneous cluster assignments during the quantization process [47, 48]. Although finetuning the FE parameters can mitigate the mismatch for noise-robust ASR, the pretrained information of the FE will be diminished. Furthermore, SSL-based models typically consists of a huge number of parameters [45, 47, 136, 137]. Consequently, efficient methods to adapt the model to new scenarios is critical. Inserting adapters within the model presents a simple but effective method. Inserting an adapter in the encoder layer or after the layer can achieve good effects at the encoding level [53]. In this process, the parameters of the Transformer encoder are frozen, and only the adapters are finetuned [53]. Addressing the mismatch mentioned above at the feature level is also necessary for noise-robust ASR, in addition to encoding-level adaptation. However, there are limited studies on FE adaptation for SSL-based pretrained models.

In this work, we investigate adaptation of the FE module based on finetuning and the adapters. Based on the observation, we propose a dual-path adaptation of FE for improving the noise-robust ASR. It consiss of a frozen-pretrained FE path and an adapted-finetuned FE path. The frozen-pretrained FE path keeps the information learned from massive pretrained data, while the adapted-finetuned FE path deal with real noise. These two paths are fused

with convolutional layers in a masking way similar to speech enhancement [139, 140]. The fusion of the two paths aims to utilize the complementarity between them.

6.2 Exploration of Adapter for Noise Robust ASR

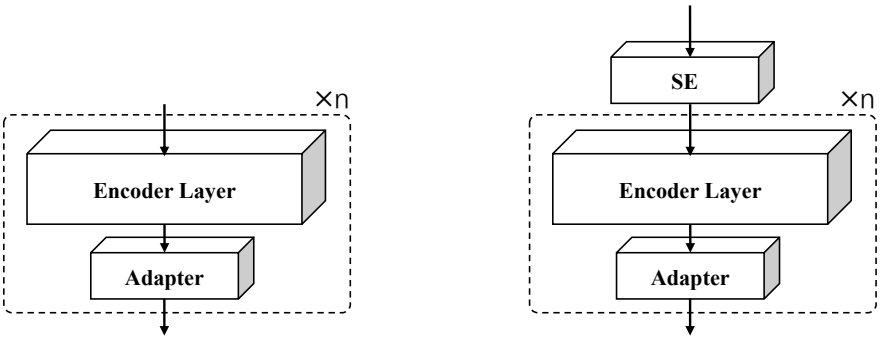
Incorporating adapters into ASR models is a prevalent practice for tasks like accent ASR, children ASR, and multi-lingual ASR. However, noise-robust ASR, another application that necessitates adaptation, has received limited attention in adapter studies. Consequently, this paper delves into the application of adapters for enhancing noise-robust speech recognition. In order to gain a deeper understanding of the adapter's role within this context, our study takes a comprehensive approach, thoroughly investigating its effect from multiple perspectives.

(1) Where should the adapter be inserted?
The position where the adapter is inserted notably impacts the model's performance. To address this, some methods employ machine learning techniques to automatically identify the optimal layers for adapter integration [130]. However, it is worth noting that the best place to insert the adapter can differ based on the specific task. To gain a clear and intuitive understanding of the adapter's influence on noise-robust ASR, we investigate the most effective layer for adapter insertion and assess whether stacking adapters yield improved effect.

(2) How to configure the embedding nodes in the adapter?
Modifications to the network structure can have a substantial impact on model performance. However, in existing adapter-based research, focus is often limited to a single chosen embedding dimension. Therefore, the impact of the embedding dimension in the adapter on noise-robust speech recognition is explored.

(3) How training data affects the adapter?
Training data is crucial for deep learning-based ASR. To explore the effects of different data quantities and types on adapter-based noise-robust ASR, we conducted experiments employing diverse training datasets. We evaluate how simulation and real training data with varying data quantities affect adaptation. Furthermore, we investigate whether incorporating simulated data could enhance the model's performance on real data. Moreover, we contrast the impact of models trained on a single noise scene with those trained under multi-noise conditions.

(4) Can the utilization of the adapter lead to further improvements for SE-based noise-robust ASR system?
Utilizing a SE front-end can significantly improve the performance of ASR systems. The SE front-end enhances features and adapts to the system at the feature level to a certain



(a) Insert adapter after encoder layer (b) Insert adapter within SE-based system

Fig. 6.1 Insert position of adapter for ASR back-end.

extent. Hence, while addressing noise-robust ASR challenges, the emphasis is frequently placed on enhancing the performance of the SE front-end. This could result in adapters being relatively uncommon in noise-robust ASR tasks. Hence, in this paper, we investigate whether adapters can further enhance the performance of models adapted at the feature level.

The adapter is positioned after the encoder layer, shown in Figure 6.1–(a). We perform pretraining on the ASR backend at the outset using a substantial dataset. Following this, we freeze all parameters of the pretrained ASR backend and insert the adapter after the encoder layer. To examine the adapter's impact in unseen noise scenarios, the training noise for the pretrained ASR back-end differs from the noise used for evaluation. The SE-based system is shown in Figure 6.1–(b).

6.3 Dual-path Adaptation for Feature Extraction Module

The pretrained feature extraction (FE) module has already learned excellent feature representations. Therefore, the FE module is often frozen during finetuning to maintain the information learned from a massive amount of data. However, there is a mismatch between the pretraining speech and the real noisy speech during evaluation.

In this paper, we propose a dual-path adaptation of the FE module for noise-robust ASR, which is depicted in Fig. 6.2–(b). The proposed FE module contains two paths: the pre-trained FE path aims to keep the information learned from the massive unlabeled data; the adapted FE path is finetuned with in-domain noisy data, which is more suitable for noise-robust ASR, but will lose the information learned in the pretraining. These two paths

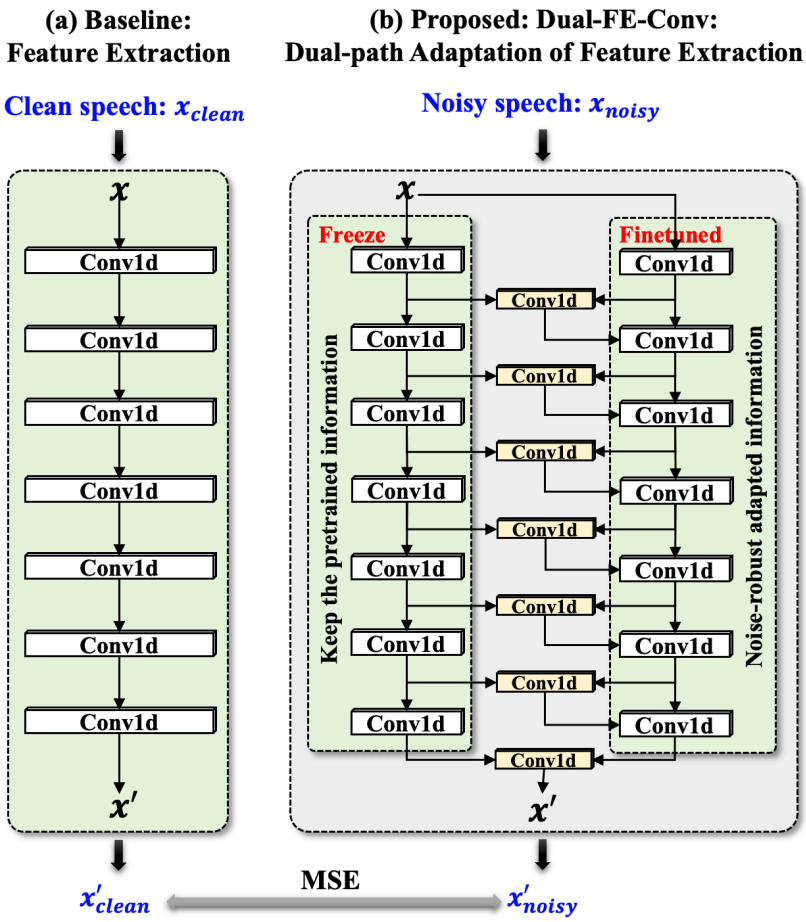


Fig. 6.2 Neural network structure of (a) baseline feature extraction module; (b) proposed dual-path adaptation of feature extraction module (Dual-FE-Conv).

can be combined by simply adding:

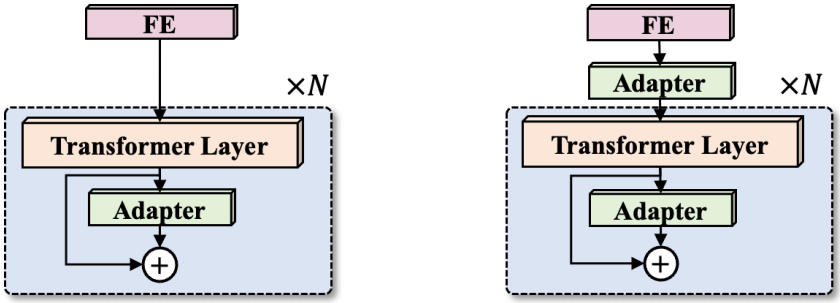
$$x_{fused} = x_{frozen} + x_{finetuned}, \tag{6.1}$$

where x_{frozen}, $x_{finetuned}$, and x_{fused} denote the features derived from the frozen FE module, the finetuned FE module, and the fused feature, respectively. The adding fusion method is denoted as **Dual-FE-Add**.

We also propose to use additional 1-D convolutional layers to fuse information from the two paths layer by layer:

$$x_{fused} = Conv_{1d}(\text{Concat}(x_{frozen}, x_{finetuned})). \tag{6.2}$$

$Conv_{1d}$ denotes the convolutional 1-D layer, and *Concat* denotes the concatenation. In this paper, the kernel size of the $Conv_{1d}$ layers are 1. $1 \times 1 - conv$ block is also known as



(a) Transformer Encoder Adaptation **(b) Feature Extraction Adaptation**

Fig. 6.3 Flowchart of (a) adapter-based adaptation for Transformer encoder; (b) adapter-based adaptation for both FE module and Transformer encoder.

pointwise convolution. Thus, the $Conv_{1d}$ layer, which is similar to masking way in the speech enhancement [108], fuses effective information from dual-path features.

We also introduce pretraining of the dual-path FE. Clean speech is input to the frozen FE module to obtain the target x'_{clean}. Then, it is compared against the adapted noisy feature x'_{noisy} derived from the proposed method to calculate the mean squared error loss:

$$\mathscr{L} = ||x'_{clean} - x'_{noisy}||^2. \tag{6.3}$$

The pretraining of the dual-path FE is shown in Fig. 6.2 in blue fonts.

Furthermore, another adapter is incorporated to the Transformer encoder. It is added after each Transformer encoder layer, shown in Fig. 6.3–(a). The number of the adapter parameters is much smaller than that of the encoder of the Transformer encoder. Only finetuning the adapter can efficiently adapt the model to different noise scenarios.

6.4 Experiments

6.4.1 Dataset

The experiments conducted utilized the CHiME-4 dataset[1]. It includes four different noise conditions: bus (bus), cafe (caf), pedestrian area (ped), and street junction (str). The audio in the dataset was digitized at a sampling rate of 16 kHz. Data from channels 1 to 6 were utilized during the model training phase. The Channel 5 real noisy data from development and evaluation sets were used for testing.

[1] https://spandh.dcs.shef.ac.uk/chime_challenge/CHiME4/index.html

6.4.2 Experimental Settings

Experimental Settings for Adapter Exploration

The Conformer-based ASR backend employed 6 encoder layers. Each encoder layer utilized the relative positional encoding module for positional encoding, encompassing a subsampling rate of 4 facilitated by 2 Conv2d layers. The multi-head attention dimension was set at 512 with 4 attention heads. Position-wise feedforward units numbered 2048, and the activation function used was swish. The input LMFB feature was 80-dimension. The decoder, based on the Transformer architecture [100], comprised 6 decoder layers, each incorporating 512 dimensions for multi-head attention with 4 attention heads. Hidden units for position-wise feedforward were specified as 2048. The dictionary was constructed utilizing transcripts from CHiME-4, WSJ0 [2], and WSJ1 [3]. The BPE vocabulary consisted of 1014 elements, which included $< blank >$, $< unk >$, and $< sos/eos >$. The ASR backend was pretrained using WSJ0, WSJ1, and Librispeech (960 hours) [105]. In the pretraining phase, we synthesized noisy speech by combining the MUSAN dataset [106] with clean speech, randomly selecting signal-to-noise ratios (SNRs) within the range of 0 to 20 dB. Pretraining was conducted 100 epochs with the SpecAug [107]. The final pretrained model was an average of ten well-performing checkpoints on the development set.

"Bi-LSTM" and "DEMUCS" [5] were chosen as the SE front-end. For "Bi-LSTM", the feature used was the magnitude of the spectrogram. The architecture included two Bi-directional LSTM (Bi-LSTM) layers and a fully connected layer. Each Bi-LSTM layer was equipped with 896 hidden nodes. There were 896 hidden nodes in each Bi-LSTM layer. For "DEMUCS", we followed the original neural network architecture[4]. The SE frontend was pretrained with the CHiME–4 dataset with 200 epochs.

For the adapter, the input and output dimensions were all 512. During adapter training, all parameters within the pretrained ASR backend remained frozen. We positioned the adapters following the encoder layers. Moreover, both the SE frontend and the adapters were updated simultaneously when utilizing the SE frontend.

Experimental Settings for Feature Extraction Adaptation

HuBERT model was employed following the same configuration as fairseq toolkit[5]. The FE module contained 7 Conv_1d layers. Except for the input channel number in the first

[2]https://catalog.ldc.upenn.edu/LDC93s6a

[3]https://catalog.ldc.upenn.edu/LDC94S13A

[4]https://github.com/facebookresearch/denoiser

[5]https://github.com/facebookresearch/fairseq/blob/main/examples/hubert
/config/finetune/base_10h.yaml

Conv_1d layer, the other input and output channel numbers were all 512. Moreover, their kernel size and stride were (10, 5), (3, 2), (3, 2), (3, 2), (3, 2), (2, 2), (2, 2), respectively. The frozen and finetuned FE modules were following this setting. For the fusion Conv_1d layers, the input and output channels were 1024 and 512 respectively. Moreover, the kernel size and stride were all 1. For the Transformer encoder, we used HuBERT Extra Large. It contained 48 Transformer layers. In each Transformer layer, the embedding dimension was 1280, the inner FFN dimension was 5120, the number of the attention heads were 16, and the projection dimension was 1024. For the adapters, the input and output dimenssons were 1280, and the middle dimension was 16. The finetuning epoch was 40.

We used various ASR back-ends to evaluate the effectiveness of the proposed method. HuBERT models were employed by following the same configuration as fairseq toolkit[6].

- **HuBERT–extraLarge trained with clean speech**: We used the HuBERT model trained with Librispeech-960 as the baseline[7] in order to make a mismatched scenario between training and testing (Exp.–1). It contained 48 Transformer layers. In each Transformer layer, the embedding dimension was 1280, the inner FFN dimension was 5120, the number of attention heads was 16, and the projection dimension was 1024.

- **HuBERT–Large trained with noisy speech**: We evaluated with the ASR back-end finetuned with noisy data. We finetuned the pretrained checkpoint of HuBERT Large[8] with Librispeech (960 hours) [105] and the MUSAN noise dataset [106] (Exp.–10). The noisy speech was made with randomly selected signal-to-noise ratios (SNRs) within the range of 0 to 20 dB. The FE is based on HuBERT Large. It contained 24 Transformer layers with 1024 embedding dimensions and 4096 inner FFN dimensions, and the number of attention heads was 16.

- **WavLM–Large trained with noisy speech**: We also explore the performance of the proposed method with a noise-robust FE. We finetuned the pretrained checkpoint of WavLM Large[9] with Librispeech (960 hours) [105] and the MUSAN noise dataset [106] (Exp.–17). It contained 24 Transformer encoder layers, 1024-dimensional hidden states, and 12 attention heads. Furthermore, it adopted the gated relative position bias in the self-attention.

[6]https://github.com/facebookresearch/fairseq/blob/main/examples/hubert/config/finetune/base_10h.yaml
[7]https://dl.fbaipublicfiles.com/hubert/hubert_xtralarge_ll60k_finetune_ls960.pt
[8]https://huggingface.co/facebook/hubert-large-ll60k
[9]https://huggingface.co/microsoft/wavlm-large/blob/main/pytorch_model.bin

We used the HuBERT model trained with Librispeech-960 as the baseline[10]. For the baseline, we directly used these parameters to decoding the CHiME–4 development and evaluation sets, as shown in Exp.–1. Then, based on this model, we tried several experiments: adapter-based adaptation in the Transformer encoder (Exp.–2); adapter-based adaptation after FE module (Exp.–3); adapter-based adaptation both in the Transformer encoder and FE module (Exp.–4); only finetuning the FE module (Exp.–5); finetuning the FE module and adapters in Transformer encoder (Exp.–6); finetuning the FE module and adapters in both Transformer encoder and FE module (Exp.–7). The proposed method was also conducted based on these parameters.

It should be emphasized that inserting adapters in the Transformer encoder layer does not require parameter pretraining. However, when adapters are inserted into or after the FE module, performance is much degraded without pretraining. Therefore, we introduced pretraining based on the loss in Eq. (6.3) for the adapters in the FE module. The training data was taken from the CHiME–4 dataset. The training epoch was 2.

Table 6.1 Performance of baseline pretrained ASR model.

System	Development Sets					Evaluation Sets				
	bus	**str**	**ped**	**caf**	**avg.**	**bus**	**str**	**ped**	**caf**	**avg.**
Pretrained	22.0	15.1	11.2	16.6	16.3	30.3	16.1	23.4	26.5	24.1

6.4.3 Results and Analysis of Adapter Exploration

Effect of the Position of Adapter

Table 6.2 and Table 6.3 shows the comparison of placing the adapters into different encoder layers. Compared with the pretrained models in Table 6.1, inserting the adapter into any encoder layer can bring considerable performance improvement to the system. Through experiments 1–6, it is more effective to insert the adapter in the shallow layer: the performance of the adaptation gradually improves as the number of layers becomes shallower. Some studies [141] have revealed that the shallower layers within the ASR model encompass signal-level information, such as speech structure. In contrast, the deeper layers tend to hold abstract information. Therefore, the shallow layer may have more noise-related information, which is why the shallow layer is more effective for adaptation.

In addition, we also compared the effects of multi-layer adaptation through experiments 7–11. Compared to solely adapting the first encoder layer, incorporating further adaptations

[10]https://dl.fbaipublicfiles.com/hubert/hubert_xtralarge_ll60k_finetune_ls960.pt

Table 6.2 Effect of placing the adapter into different encoder layers on development sets (**trained using the entire CHiME–4 training dataset**).

Exp.	Position of Adapters						Development Sets				
	E1	E2	E3	E4	E5	E6	bus	str	ped	caf	avg.
1						✓	17.1	10.3	8.5	11.1	11.8
2					✓		15.9	9.6	7.9	9.8	10.8
3				✓			14.3	8.8	7.2	9.2	9.9
4			✓				14.4	8.6	6.8	8.9	9.7
5		✓					13.5	8.0	**6.6**	8.2	9.1
6	✓						**13.4**	**7.8**	6.7	**8.0**	**9.0**
7	✓	✓					13.0	7.9	6.5	8.1	8.9
8	✓	✓	✓				13.1	7.8	6.5	8.2	8.9
9	✓	✓	✓	✓			12.9	7.9	6.8	7.9	8.9
10	✓	✓	✓	✓	✓		12.9	7.9	6.7	8.0	8.9
11	✓	✓	✓	✓	✓	✓	13.1	7.9	6.9	8.1	9.0

Table 6.3 Effect of placing the adapter into different encoder layers on evaluation sets (**trained using the entire CHiME–4 training dataset**).

Exp.	Position of Adapter						Evaluation Sets				
	E1	E2	E3	E4	E5	E6	bus	str	ped	caf	avg.
1						✓	25.9	12.1	17.8	20.4	19.1
2					✓		24.4	11.5	16.3	19.1	17.8
3				✓			23.1	10.6	15.0	17.1	16.5
4			✓				23.1	10.8	15.2	17.5	16.7
5		✓					21.6	**9.8**	13.9	16.2	15.4
6	✓						**20.8**	10.2	**13.7**	**15.9**	**15.1**
7	✓	✓					20.3	9.9	13.7	16.1	15.0
8	✓	✓	✓				20.4	10.1	13.8	16.1	15.1
9	✓	✓	✓	✓			20.5	9.6	13.3	15.6	14.7
10	✓	✓	✓	✓	✓		20.4	9.5	13.7	15.6	14.8
11	✓	✓	✓	✓	✓	✓	20.3	9.6	13.3	15.6	14.7

in deeper layers did not result in substantial performance enhancements. Considering the abovementioned analysis, leveraging more noise-related information, the self-adaptation at the shallow layer has already achieved satisfactory effect. Attempting to enhance information encoding in the deep layer by reducing noise-related information is challenging, resulting in minimal performance improvements. However, the best performance was achieved by

Table 6.4 Effect of different embedding dimensions in adapter (**trained using the entire CHiME–4 training dataset**).

Exp.	Emb Dim	Development Sets					Evaluation Sets				
		bus	str	ped	caf	avg.	bus	str	ped	caf	avg.
12	16	13.1	7.9	6.8	8.1	9.0	20.6	9.8	13.4	15.6	14.9
13	32	13.2	7.8	6.8	8.1	9.0	20.8	9.9	13.4	15.9	15.0
11	64	13.1	7.9	6.9	8.1	9.0	20.3	9.6	13.3	15.6	14.7
14	96	13.3	8.1	6.9	7.7	9.0	20.6	9.6	13.5	15.8	14.9
15	128	12.9	7.9	6.9	8.0	8.9	20.6	9.8	13.6	15.6	14.9

inserting the adapter after all encoder layers on the evaluation sets (experiment 11). Thus, in subsequent experiments, we placed adapters after all encoder layers.

Effect of the Adapter Embedding Dimension

Table 6.4 shows the comparison of different embedding dimensions in the adapter. We tried adapters with embedding dimensions of 16, 32, 64, 96, and 128 (experiments 11–15). Despite the wide range of embedding dimensions, the results demonstrate consistent performance. This indicates a degree of robustness in the adapter, as it appears unaffected by the embedding dimension. Based on the results of these models on the evaluation sets, we ultimately chose to utilize the 64-dimensional embedding adapter for the subsequent experiments.

Effect of the Training Data

Table 6.5 and Table 6.6 show the comparison of different training sets for adapter training. When comparing experiments 11 and 16, it becomes evident that incorporating simulated data during training leads to better performance for real noisy sets: the relative improvement was 4% and 7% for development and evaluation sets, respectively. This trend becomes more pronounced when the quantity of real data diminishes: the relative improvements for experiments 17–18, 20–21, 23–24 were 6%, 16%, and 25% in development sets; and 8%, 22%, and 27% in evaluation sets, respectively. Nevertheless, using simulated data might constrain the model from achieving further improvements in performance with real data. When comparing experiments 18–21 (21–24), it becomes evident that the addition of more real data can lead to substantial performance enhancements, resulting in relative improvements of 11% (11%) and 15% (9%) for development and evaluation sets, respectively. On the other hand, incorporating the same simulated data did not have any effect in the comparative experiments 17, 20, and 23.

Table 6.5 Effect of different training sets for adapter-based adaptation on development sets. "**Held**" represents whether the specific noise conditions were excluded during model training; when utilizing the held-out approach, both the training and testing sets utilize a single noise type condition. "**Real**" represents whether the real noisy data is used during the training process. "**Simu.**" represents whether the simulated noisy data is used during training. "**Utt.**" represents how many utterances (channels 1 to 6 of the same utterance are considered single utterances) are used during the training process. ♣ represents the number of all utterances in the corresponding noisy condition (this is due to the slightly different amounts of simulated data for the four noise conditions). ★ represents that 100 sentences are selected from four noise conditions to constitute a training set.

Exp.	Training Data				Development Sets					
	Held	**Real**	**Utt.**	**Simu.**	**Utt.**	**bus**	**str**	**ped**	**caf**	**avg.**
11	✗	✓	1,600	✓	7,138	13.1	7.9	6.9	8.1	9.0
16	✗	✓	1,600	✗	-	13.7	8.2	7.1	8.6	9.4
17	✓	✓	400	✓	♣	13.6	8.2	7.0	8.4	9.3
18	✓	✓	400	✗	-	14.5	8.8	7.2	9.1	9.9
19	✓	✗	-	✓	400	15.1	8.9	7.5	9.1	10.1
20	✓	✓	200	✓	♣	13.7	8.1	7.0	8.4	9.3
21	✓	✓	200	✗	-	16.6	9.6	7.9	10.4	11.1
22	✓	✗	-	✓	200	18.1	10.0	8.9	11.7	12.2
23	✓	✓	100	✓	♣	13.9	8.2	7.0	8.5	9.4
24	✓	✓	100	✗	-	18.4	11.2	8.8	11.7	12.5
25	✓	✗	-	✓	100	20.6	11.9	10.6	13.8	14.2
26	✗	✓	★400	✗	-	14.8	8.7	7.2	8.9	9.9

Real data yields better adaptation performance when using the same amount of data than simulated data. This becomes particularly noticeable, especially when dealing with smaller amounts of data: the relative improvements for experiments 18–19, 21–22, 24–25 were 2% (development sets), 9% (development sets), and 12% (development sets); and 2% (evaluation sets), 5% (evaluation sets), and 7% (evaluation sets), respectively. This could be attributed to the distinct distribution of simulated data compared to real data. Nevertheless, with a substantial volume of simulated data, certain instances might exhibit a distribution comparable to real data, consequently improving the model's performance on the real test sets.

Furthermore, we investigated how multi-condition training influences the adapter's effectiveness. The performance of experiments 18 and 26 were the same. This could be due to shared noises (because each noise condition is composed of multiple noises) among the

Table 6.6 Effect of different training sets for adapter-based adaptation on evaluation sets. "**Held**" represents whether the specific noise conditions were excluded during model training; when utilizing the held-out approach, both the training and testing sets utilize a single noise type condition. "**Real**" represents whether the real noisy data is used during the training process. "**Simu.**" represents whether the simulated noisy data is used during training. "**Utt.**" represents how many utterances (channels 1 to 6 of the same utterance are considered single utterances) are used during the training process. ♣ represents the number of all utterances in the corresponding noisy condition (this is due to the slightly different amounts of simulated data for the four noise conditions). ★ represents that 100 sentences are selected from four noise conditions to constitute a training set.

Exp.	Training Data				Evaluation Sets					
	Held	**Real**	**Utt.**	**Simu.**	**Utt.**	**bus**	**str**	**ped**	**caf**	**avg.**
11	✗	✓	1,600	✓	7,138	20.3	9.6	13.3	15.6	14.7
16	✗	✓	1,600	✗	-	22.6	9.8	14.1	16.7	15.8
17	✓	✓	400	✓	♣	21.3	10.2	13.6	15.6	15.2
18	✓	✓	400	✗	-	23.2	10.3	14.8	17.8	16.5
19	✓	✗	-	✓	400	23.6	10.5	15.9	17.4	16.9
20	✓	✓	200	✓	♣	21.4	10.1	13.9	15.9	15.3
21	✓	✓	200	✗	-	25.4	13.4	18.1	21.1	19.5
22	✓	✗	-	✓	200	26.4	13.6	20.2	22.0	20.5
23	✓	✓	100	✓	♣	22.0	10.2	13.8	16.1	15.5
24	✓	✓	100	✗	-	27.2	14.7	20.3	23.1	21.3
25	✓	✗	-	✓	100	29.4	15.2	22.2	24.5	22.8
26	✗	✓	★400	✗	-	23.2	10.5	14.7	17.3	16.4

four noise scenes in the CHiME–4 dataset. As a result, the noise is only partially unseen, thus partially limiting the potential effects of multi-condition training. It also serves as an inspiration that incorporating similar noisy real data as augmented data can result in substantial performance improvements (compare experiments 24 and 26).

Effect of the Adapter for SE-based robust ASR

Table 6.7 shows the adapter's performance for different SE-based robust ASR systems. Experiments 27 and 29 significantly improved the performance from the pretrained model when the SE front-end was used. Incorporating adapters within the SE-based robust ASR system further improved recognition performance. While feature enhancement of the SE front-end has been achieved, significant benefits still arise from adaptation at the backend.

Table 6.7 Effect of adapter for different SE-based robust ASR systems (**trained using the entire CHiME–4 training dataset**).

Exp.	System	Development Sets					Evaluation Sets				
		bus	str	ped	caf	avg.	bus	str	ped	caf	avg.
27	Bi-LSTM	13.3	8.1	6.8	8.4	9.2	21.4	10.1	13.5	16.1	15.3
28	+ adapter	12.4	7.5	6.7	7.9	8.6	19.7	9.8	12.4	14.1	14.0
29	DEMUCS	11.7	7.7	6.3	6.9	8.2	19.2	9.2	12.8	14.9	14.0
30	+ adapter	10.7	6.7	6.4	6.5	7.6	16.8	8.3	11.4	13.2	12.4

This is because the SE frontend might introduce information loss or distortion and the adapter plays a role in mitigating these issues.

6.4.4 Results and Analysis of Feature Extraction Adaptation

Comparison of adapter-based adaptation

Table 6.8 and Table 6.9 (upper rows) shows the performance of different ASR systems for the development and evaluation sets, respectively. By comparing Exp.–1 and Exp.–2, adding an adapter into the Transformer layer significantly improved the performance of ASR. This shows that adapter-based encoder-level adaptation is very effective. By comparing Exp.–1 and Exp.–3, adding an adapter in the FE module also significantly improved the performance of ASR. However, Exp.–4 shows that inserting an adapter into both the FE module and Transformer encoder does not improve ASR performance from Exp.–2. The result suggests that combining these two-module adaptations with the adapters presents a challenge. The Transformer encoder adaptation more readily influences the overall performance of the model.

Then, we tried to finetune only the FE module instead of using adapters. According to the results of Exp.–5, the performance is not improved in the development sets and degraded in the evaluation sets. This result shows that finetuning the FE module does not achieve effective noise reduction or adaptation. On the other hand, as shown in Exp.–6, combining the Transformer adapter with FE finetuning significantly improved the performance. Compared with Exp.–2, which only inserts an adapter to the encoder, it showed 29% and 31% relative improvements in the development and evaluation sets, respectively. It also significantly outperforms Exp.–5. The result shows that FE finetuning is effective only when combined with the encoder adaptation, which addresses the mismatch.

The similar trend is observed with the noise speech-trained ASR systems, which were shown in Table 6.10 to 6.13.

Table 6.8 Evaluation with HuBERT finetuned with LibriSpeech–960 on development sets: "FE" represents the feature extraction module; "Enc" representes the Transformer encoder; "FT" means finetuning all parameters; "Ada" means the use of adapters.

	Exp.	FE		Enc	BUS	STR	PED	CAF	AVE.
		FT	Ada.	Ada.					
1	Clean Trained HuBERT				25.1	23.9	15.5	21.9	21.6
2				✓	18.4	17.6	10.6	15.4	15.5
3			✓		21.1	18.5	11.8	16.8	17.1
4			✓	✓	19.2	18.0	10.2	15.1	15.6
5		✓			24.4	23.5	14.7	21.3	21.0
6		✓		✓	14.7	11.5	8.5	10.8	11.4
7	Dual-FE-Add	✓		✓	12.3	9.1	6.8	8.5	9.2
8	Dual-FE-Conv	✓			14.9	11.8	8.7	11.0	11.6
9	Dual-FE-Conv	✓		✓	11.9	8.1	6.6	8.3	8.7

Table 6.9 Evaluation with HuBERT finetuned with LibriSpeech–960 on evaluation sets: "FE" represents the feature extraction module; "Enc" representes the Transformer encoder; "FT" means finetuning all parameters; "Ada" means the use of adapters.

	Exp.	FE		Enc	BUS	STR	PED	CAF	AVE.
		FT	Ada.	Ada.					
1	Clean Trained HuBERT				42.5	25.3	28.7	33.2	32.4
2				✓	30.4	17.7	19.9	23.6	22.9
3			✓		35.3	19.9	21.5	25.5	25.5
4			✓	✓	31.4	18.5	19.4	23.3	23.2
5		✓			42.2	24.9	29.1	33.9	32.5
6		✓		✓	22.5	11.6	13.8	15.8	15.9
7	Dual-FE-Add	✓		✓	19.4	9.3	11.1	13.0	13.2
8	Dual-FE-Conv	✓			23.1	11.0	13.7	17.0	16.2
9	Dual-FE-Conv	✓		✓	17.7	8.6	11.0	12.4	12.5

Effect of dual-path FE for clean speech-trained HuBERT

Table 6.8 and Table 6.9 (lower rows) show results of the proposed dual-path adaptation of the FE for clean speech-trained HuBERT. Simple adding (Exp.–7) provides performance improvement compared with Exp.–6, but a much larger improvement is gained when using the convolutional layers to fuse the two features layer by layer. Compared with Exp.–6, Exp.–9 showed 24% and 21% relative improvements in real data of the development and evaluation sets, respectively. These results confirm information complementarity between the

Table 6.10 Evaluation with HuBERT finetuned with LibriSpeech–960 and MUSAN noises on development sets.

Exp.		FE FT	FE Ada.	Enc Ada.	BUS	STR	PED	CAF	AVE.
10	Noise Trained				19.8	16.5	13.8	14.9	16.3
11				✓	15.1	12.8	9.5	11.5	12.3
12	HuBERT	✓			16.9	13.8	11.6	13.3	13.9
13		✓		✓	14.0	11.8	9.4	11.2	11.6
14	Dual-FE-Add	✓		✓	12.4	9.6	8.1	9.4	9.9
15	Dual-FE-Conv	✓			15.9	13.6	11.2	12.8	13.4
16	Dual-FE-Conv	✓		✓	11.9	9.8	8.9	8.2	9.7

Table 6.11 Evaluation with HuBERT finetuned with LibriSpeech–960 and MUSAN noises on evaluation sets.

Exp.		FE FT	FE Ada.	Enc Ada.	BUS	STR	PED	CAF	AVE.
10	Noise Trained				26.3	17.2	17.8	20.1	20.3
11				✓	21.8	13.4	15.4	16.0	16.6
12	HuBERT	✓			22.8	13.8	16.1	16.9	17.4
13		✓		✓	20.0	11.8	14.3	14.9	15.3
14	Dual-FE-Add	✓		✓	18.4	10.2	13.6	14.3	14.1
15	Dual-FE-Conv	✓			21.5	13.5	15.7	16.9	16.9
16	Dual-FE-Conv	✓		✓	17.8	10.1	13.7	13.0	13.7

two features. This complementarity can be effectively utilized with more complex networks like $Conv_{1d}$ layers for the clean model. The performance difference between Exp.–9 and Exp.–7 is statistically significant (p-value < 0.05).

The result without adapters in the encoder (Exp.–8) shows an improvement from the same setting (Exp.–5), but it is much degraded from Exp.–9, showing the importance of the adapter.

We also compared the proposed system with directly finetuning the HuBERT–extraLarge with CHiME–4 dataset. The average WER of the real evaluation sets was 13.5, which is worse than the proposed methods (Exp.–9).

Table 6.12 Evaluation with WavLM finetuned with LibriSpeech–960 and MUSAN noises on development sets.

Exp.		FE FT	FE Ada.	Enc Ada.	BUS	STR	PED	CAF	AVE.
17	Noise Trained WavLM				11.0	9.8	7.9	9.1	9.5
18				✓	9.5	7.8	6.9	7.9	8.1
19		✓			10.9	9.0	7.7	9.0	9.1
20		✓		✓	9.1	7.9	6.9	8.0	8.0
21	Dual-FE-Add	✓		✓	8.2	7.0	5.9	6.2	6.8
22	Dual-FE-Conv	✓			11.2	9.9	8.1	9.2	9.6
23	Dual-FE-Conv	✓		✓	8.7	7.3	6.5	7.2	7.4

Table 6.13 Evaluation with WavLM finetuned with LibriSpeech–960 and MUSAN noises on evaluation sets.

Exp.		FE FT	FE Ada.	Enc Ada.	BUS	STR	PED	CAF	AVE.
17	Noise Trained WavLM				14.1	9.0	10.1	10.8	11.0
18				✓	12.2	8.1	9.5	9.4	9.8
19		✓			13.2	8.9	9.7	10.6	10.6
20		✓		✓	11.6	8.1	8.8	9.4	9.5
21	Dual-FE-Add	✓		✓	10.5	6.8	7.9	8.3	8.4
22	Dual-FE-Conv	✓			13.1	9.0	9.9	10.8	10.7
23	Dual-FE-Conv	✓		✓	11.7	7.2	8.6	8.6	9.0

6.4.5 Evaluations with noisy speech-trained HuBERT

Table 6.10 and Table 6.11 show the results with noisy speech-trained HuBERT. The FE module adaptation was also effective in this model. The improvements by Exp.–14 and Exp.–16 from Exp.–13 are significant (p-value < 0.01) for development and evaluation sets. However, the improvement without adapters (from Exp.–12 and Exp.–15) is not so large. The Dual-FE-Conv (Exp.–16) was better than Dual-FE-Add (Exp.–14), but the difference between them is not significant.

6.4.6 Evaluations with noisy speech-trained WavLM

Table 6.12 and Table 6.13 show the results with WavLM. The proposed method was also effective for this model. The improvement from Exp.–20 to Exp.–23 is statistically significant

(p-value < 0.05) for the development and evaluation sets. In this model, however, the performance of Dual-FE-Add (Exp.–21) was better than Dual-FE-Conv (Exp.–23) (p-value < 0.05). The synergy of the proposed dual-path FE adaptation with adapters within the encoder is confirmed, but the complex fusion mechanism is not needed in the noise-robust model.

6.5 Conclusion

In this chapter, we investigate how to adapt the ASR back-end with limited data. We first explored the effect of the adapter on noise-robust ASR. We conducted a comprehensive exploration from various perspectives, including the optimal insertion position for the adapter, the quantity and type of data used for adapter training, and the synergy of the adapter with the SE. The experimental results demonstrate that incorporating adapters in the shallow layer yields more effectiveness compared to the deep layer. Furthermore, the number of embedding nodes in the adapter does not significantly impact the adaptation process. Moreover, the training dataset plays a vital role in adapter training: When considering the same data amount, using real data is more effective than simulated data, but adding simulated data can enhance the performance of the real test sets. Combining adapters with the SE front-end leads to further performance improvement. With SSL, ASR performance has been significantly improved. We have proposed a dual-path adaptation of the feature extraction (FE) module to address the data mismatch between pretraining and evaluation. The proposed FE module combines a frozen-pretrained and adaptive-finetuned FE path. The features extracted by these two paths contain information complementarity. Furthermore, 1-D convolutional layers are adopted to fuse the information between these two paths layer by layer. Moreover, we used adapters to adapt the Transformer encoder. The experimental results using the CHiME–4 dataset show that the combination of finetuning FE with adapters in the encoder provides synergy, and the proposed method significantly improved the ASR performance.

Chapter 7

Comparision of Different SE Methods

All the proposed methods within Chapter 3 to 5 belong to deterministic methods. Deterministic SE systems learn optimal deterministic mapping from noisy speech to clean speech [4, 108, 142–147]. The proposed methods in Chapter 3 and 4 are all frequency-domain methods. The magnitude of the spectrogram is enhanced. In Chapter 3, we first proposed subband-based spectrogram fusion (SBSF) to enhance the poor-performance sub-band information within the full-band enhanced spectrogram. Then, we proposed minimum difference masks-based spectrogram fusion (MDMs-SF) to fuse the better recovery parts of the mapping and masking enhanced spectrograms. In Chapter 4, we first proposed the mask-masked spectrograms fusion (MM-SF) to highlight the speech component within the noisy spectrogram. Then, we proposed the multi-resolution spectrograms fusion (MR-SF) to utilize the complementarity between multi-resolution spectrograms. In Chapter 5, we proposed incorporating frequency information into the waveform-domain SE methods (WaveSpecEnc).

On the other hand, probabilistic SE systems capture the target distribution, either implicitly or explicitly [148, 149, 142, 150]. Among probabilistic systems, diffusion models have been investigated across various tasks [151, 152]. Diffusion models are inspired by non-equilibrium thermodynamics. The data are gradually transformed into noise, during which a neural network learns to reverse the incremental process of noise addition. The score-based diffusion model shows excellent performance for various SE tasks such as speech denoising, dereverberation, blind source separation, and target speech extraction (TSE) [149, 142, 153–155]. This model is based on a stochastic differential equation (SDE), which makes the training fully probabilistic without any prior noise distribution assumptions [142, 156]. Its reverse diffusion process is also based on SDE [156].

The diffusion model is hard to use directly for ASR [157, 158] because it is time-consuming. Thus, we proposed a unified system (GP-Unified) that uses jointly deterministic and probabilistic decoders to speed up the diffusion process. Nevertheless, the diffusion

Table 7.1 Comparison of different proposed systems. “Dete.” represents the deterministic methods; “Prob.” represents the probabilistic methods.

System				**SIG**	**BAK**	**OVRL**	**PESQ**
Input Feature		Noisy		3.35	2.44	2.63	1.97
Dete.	Frequency	Bi-LSTM	mapping (Table 3.1)	3.85	2.55	3.23	2.60
			masking (Table 3.1)	3.65	2.49	3.07	2.51
			SBSF (Table 3.1)	4.09	3.12	3.42	2.74
			MDMs-SF	4.01	2.65	3.35	2.67
		CRN	masking (Table 4.1)	3.51	2.98	3.02	2.56
			two-stage (Table 4.1)	3.60	3.04	3.10	2.62
			MM-SF (Table 4.1)	4.02	3.10	3.37	2.72
			MR-SF (Table 4.2)	3.81	3.22	3.22	2.66
	Waveform	DEMUCS	mapping	4.22	3.25	3.52	2.93
			WaveSpecEnc	4.40	3.52	3.77	3.07
Prob.	Complex	NCSN++	sgmse+	-	-	-	2.93
			GP-Unified	-	-	-	2.97

model is still time-consuming. In this Chapter, we also compare all the deterministic and probabilistic methods of human hearing experiences.

We compared all proposed methods in Table 7.1. “SBSF” is a mapping-based system. Compared to the “Bi-LSTM (mapping)”, it shows a stronger noise suppression effect, as the improvement in noise suppression (BAK) was more significant than the improvements in signal recovery (SIG) and overall quality (OVRL). “MDMs-SF” combines both spectrograms of mapping and masking systems. Compared to using mapping (“Bi-LSTM (mapping)”) or masking (“Bi-LSTM (masking)”) only, it shows a more consistent improvement in signal recovery (SIG), noise suppression (BAK), and overall signal quality (OVRL). However, compared to “SBSF”, “MDMs-SF” performed significantly worse in noise suppression (BAK).

“CRN (masking)” has comparable performance to “Bi-LSTM (masking)”. Nevertheless, “CRN (masking)” had better noise suppression (BAK), while “Bi-LSTM (masking)” had greater advantages in signal recovery. “CRN (two-stage)” further improved the performance of “CRN (masking)” in all aspects (signal recovery (SIG), noise suppression (BAK), and overall signal quality (OVRL)). “MM-SF” aims to emphasize the speech component as the input feature, and the experimental results demonstrate its strong speech restoration ability (SIG). “MR-SF” uses the multi-resolution spectrogram as the input feature. Compared to “CRN (masking)”, it shows improvements across all aspects (signal recovery, noise suppression, and overall signal quality). Compared to “MM-SF”, “MR-SF” achieved better noise suppression (BAK) but has poorer signal recovery (SIG). “MM-SF” had comparable

performance to “SBSF”, especially for signal recovery (SIG). This suggests that re-enhancing poor performance subbands is beneficial for restoring speech signals. However, designing input features to emphasize speech components was more effective, as “Bi-LSTM (masking)” had better signal restoration ability compared to “CRN (masking)”. “MR-SF” had best noise suppression (BAK) among all proposed frequency methods (compared to “SBSF”, “MDMs-SF”, and “MM-SF”). This suggests that the model more effectively extracts noise information from multi-resolution spectral features, which helps to improve enhancement performance.

“DEMUCS (mapping)” directly processes the waveform-domain feature. With the phase information, “DEMUCS (mapping)” had a better performance compared to the frequency-domain methods. All the speech recovery (SIG) and noise suppression (BAK) were improved. However, compared to “MR-SF”, “DEMUCS (mapping)” improved performance by better restoring both the speech signal (SIG) and the overall signal (OVRL), as their noise suppression (BAK) capabilities was comparable. With spectral information, “WaveSpecEnc” improved performance in all aspects (signal recovery (SIG), noise suppression (BAK), and overall signal quality (OVRL)). Additionally, compared to “DEMUCS (mapping)”, “WaveSpecEnc” had superior noise suppression performance (BAK). This also indicates that incorporating spectral information into the model improves noise information capture, resulting in enhanced performance. We used a score-based diffusion model as the probabilistic model. The model used the complex spectrogram as the input and output feature. “sgmse+” had a comparable performance to “DEMUCS (mapping)”.

Chapter 8

Conclusions

8.1 Contributions

This thesis focuses on effective extracting complementary representations from single speech audio and incorporating them into neural networks to improve SE and ASR performance. We fuse the complementary representations from the enhanced features and input features.

In Chapter 3, we proposed the spectrogram fusion methods. Although the mapping-based and masking-based enhanced spectrograms show some complementarities, few works have analyzed the reason. Thus, we first analyzed the complementary between these two learning targets. Then, we proposed the subband-based spectrogram fusion (SBSF) to refine the poor performance sub-bands based on the conclusions. Furthermore, we proposed the minimum difference masks-based spectrogram fusion (MDMs-based SF) to improve the ASR performance by fusing the mapping-based and masking-based spectrograms. The experimental results show that the complementary features help to improve the SE and ASR.

In Chapter 4, we investigated how to use spectrograms more effectively. We first proposed the multi-masked spectrogram fusion (MM-SF) to highlight the speech component in the spectrogram. The spectrogram was extracted according to the mask of a pretrained masking-based SE. The multi-masked spectrogram was inputted to the neural network. The experimental results show that the MM-SF helps the neural network to extract better-hidden representations. Furthermore, the spectrogram can be divided into wideband and narrowband according to the resolution. We proposed the multiple-resolution spectrograms system. The proposed methods achieved better spectral recovery on silent segments and high-frequency spectrograms.

In Chapter 5, we further investigated how to use the spectrogram information. We improve the robustness of waveform-domain SE with spectrogram encoding. The temporal feature maps at each encoder layer in the SE front-end are refined by spectral information.

Furthermore, we incorporate the spectral information of the encoder layer into the ASR back-end ("WaveSpecEnc+") to conduct effective joint training. The proposed system shows robustness in unseen conditions, and also effective with the finetuned ASR system. Besides, we tried to use the SE front-end to adapt the ASR back-end in the unseen noise conditions. We found that only finetuning the SE front-end helps the ASR system under unseen noise conditions.

In Chapter 6, we investigated how to adapt the ASR back-end under unseen noise conditions more effectively. We first explored the noise-robust ASR with the adapter. We conducted a comprehensive exploration from various perspectives, including the optimal insertion position for the adapter, the quantity and type of data used for adapter training, and the synergy of the adapter with the SE. The ASR performance was significantly improved with self-supervised learning (SSL). We also proposed a dual-path adaptation of the feature extraction (FE) module to address the data mismatch between pretraining and evaluation. The proposed FE module combines a frozen-pretrained and adaptive-finetuned FE path. The features extracted by these two paths contain information complementarity. The experimental results show that the proposed method utilized the complementarity between the two paths and improved the ASR performance significantly.

In Chapter 7, we made a comparision among all proposed methods. For frequency-domain methods, the "Bi-LSTM" methods had better speech signal recovery (according to SIG), while the "CRN" methods had better noise suppression (according to BAK). Significant improvements in frequency-domain speech enhancement were confirmed with "SBSF", "MDMs-SF", "MM-SF", and "MR-SF". The waveform-domain method, "DEMUCS", showed significantly performance better than frequency-domain methods. "WaveSpecEnc" demonstrated significant improvement from "DEMUCS (mapping)", with improvements in speech recovery and noise suppression.

8.2 Future Work

We have found that 1) complementarity between the different learning targets; 2) complementarity between multiple resolution magnitude spectrograms; 3) complementarity between waveform and the magnitude spectrogram; 4) complementarity between the finetuned and pre-trained FE module of SSL model. There are still many feature representations need to be investigated, e.g., complex spectrogram. Besides, this thesis only focuses on additive noise. In actual application environments, reverberation and inference from other speakers also affect speech quality.

Thus, in the future, we will try more feature representation combinations. Complex-domain features show potential, because it contains both the magnitude and phase information. We will use the complex-domain features to improve the robustness of waveform-domain models. Besides, our proposed model only fuses the multiple-resolution spectrograms or waveform-spectrogram in the encoder part. Improving the decoder and hidden embedding layers has another possibility. Furthermore, the single-channel SE has limitation. The multi-channel SE front-end has more potential. Thus, we will improve the current model from the sing-channel to the multi-channel system.

We only tried the adapter-based method for unseen noise adaptation of the ASR back-end. Some other adaptation methods still exist. We will improve the adapter structure and try to combine the adapter with other adaptation methods. Besides, we only try to use the adapter to adopt the "M"-level amount of model parameters. The "B"-level amount of the model parameter also needs to be verified.

Finally, multi-speaker inferences will be considered in our model. Currently, the multi-speaker front-end brings more speech distortion and loss compared with additive noise. As a result, the front-end is rarely considered for multi-speaker ASR systems. We will also design the hybrid front-end for the multi-speaker front-end for ASR.

References

[1] D. Yu and L. Deng. *Automatic speech recognition*, volume 1. Springer, 2016.

[2] M. Benzeghiba, R. De Mori, O. Deroo, S. Dupont, T. Erbes, D. Jouvet, L. Fissore, P. Laface, A. Mertins, C. Ris, R. Rose, V. Tyagi, and C. Wellekens. Automatic speech recognition and speech variability: A review. *Speech Communication*, 49(10):763–786, 2007.

[3] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee. A regression approach to speech enhancement based on deep neural networks. *IEEE/ACM TASLP*, 23(1):7–19, 2015.

[4] H. Shi, L. Wang, M. Ge, S. Li, and J. Dang. Spectrograms fusion with minimum difference masks estimation for monaural speech dereverberation. In *Proc. ICASSP*, pages 7544–7548, 2020.

[5] A. Défossez, G. Synnaeve, and Y. Adi. Real Time Speech Enhancement in the Waveform Domain. In *Proc. Interspeech*, pages 3291–3295, 2020.

[6] A. Pandey and D. Wang. A new framework for CNN-based speech enhancement in the time domain. *IEEE/ACM TASLP*, 27(7):1179–1188, 2019.

[7] Y. Hu, Y. Liu, S. Lv, M. Xing, S. Zhang, Y. Fu, J. Wu, B. Zhang, and L. Xie. DCCRN: Deep Complex Convolution Recurrent Network for Phase-Aware Speech Enhancement. In *Proc. Interspeech*, pages 2472–2476, 2020.

[8] K. Tan and D. Wang. Complex spectral mapping with a convolutional recurrent network for monaural speech enhancement. In *Proc. ICASSP*, pages 6865–6869, 2019.

[9] T. Gerkmann, M. Krawczyk-Becker, and J. Le Roux. Phase processing for single-channel speech enhancement: History and recent advances. *IEEE Signal Processing Magazine*, 32(2):55–66, 2015.

[10] K. Tan and D. Wang. A convolutional recurrent neural network for real-time speech enhancement. In *Proc. Interspeech*, pages 3229–3233, 2018.

[11] M. Kolbæk, Z.-H. Tan, S. H. Jensen, and J. Jensen. On loss functions for supervised monaural time-domain speech enhancement. *IEEE/ACM TASLP*, 28:825–838, 2020.

[12] Y. Zhao and D. Wang. Noisy-Reverberant Speech Enhancement Using DenseUNet with Time-Frequency Attention. In *Proc. Interspeech*, pages 3261–3265, 2020.

[13] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee. An experimental study on speech enhancement based on deep neural networks. *IEEE Signal Processing Letters*, 21(1):65–68, 2014.

[14] J. Woo, M. Mimura, K. Yoshii, and T. Kawahara. End-to-end music-mixed speech recognition. In *Proc. APSIPA ASC*, pages 800–804, 2020.

[15] S. Dang, T. Matsumoto, Y. Takeuchi, and H. Kudo. Using Semi-supervised Learning for Monaural Time-domain Speech Separation with a Self-supervised Learning-based SI-SNR Estimator. In *Proc. INTERSPEECH*, pages 3759–3763, 2023.

[16] S. Dang, T. Matsumoto, Y. Takeuchi, and H. Kudo. A separation priority pipeline for single-channel speech separation in noisy environments. In *Proc. ICASSP*, pages 12511–12515, 2024.

[17] S. Dang, T. Matsumoto, H. Kudo, and Y. Takeuchi. A restriction training recipe for speech separation on sparsely mixed speech. In *Proc. ICONIP*, volume 1517 of *Communications in Computer and Information Science*, pages 736–743. Springer, 2021.

[18] K. Han, Y. Wang, and D. Wang. Learning spectral mapping for speech dereverberation. In *Proc. ICASSP*, pages 4628–4632, 2014.

[19] X. Lu, Y. Tsao, S. Matsuda, and C. Hori. Speech enhancement based on deep denoising autoencoder. In *Proc. Interspeech*, volume 2013, pages 436–440, 2013.

[20] D. Wang and G. J. Brown. *Contributors*, pages xix–xx. 2006.

[21] D. Wang. On ideal binary mask as the computational goal of auditory scene analysis . In *Proc. Speech Separation by Humans and Machines*, pages 181–197. Springer, 2005.

[22] A. Narayanan and D. Wang. Ideal ratio mask estimation using deep neural networks for robust speech recognition. In *Proc. ICASSP*, pages 7092–7096, 2013.

[23] S. R. Park and J. Lee. A fully convolutional neural network for speech enhancement. *arXiv preprint arXiv:1609.07132*, 2016.

[24] L. Sun, J. Du, L.-R. Dai, and C.-H. Lee. Multiple-target deep learning for LSTM-RNN based speech enhancement. In *Proc. HSCMA*, pages 136–140, 2017.

[25] M. Ge, L. Wang, N. Li, H. Shi, J. Dang, and X. Li. Environment-dependent attention-driven recurrent convolutional neural network for robust speech enhancement. In *Proc. Interspeech*, pages 3153–3157, 2019.

[26] S. Fu, Y. Tsao, and X. Lu. SNR-aware convolutional neural network modeling for speech enhancement. In *Proc. Interspeech*, pages 3768–3772, 2016.

[27] T Gao, J Du, L Dai, and C. Lee. SNR-based progressive learning of deep neural network for speech enhancement. In *Proc. Interspeech*, pages 3713–3717, 2016.

[28] H. Shi, K. Shimada, M. Hirano, T. Shibuya, Y. Koyama, Z. Zhong, S. Takahashi, T. Kawahara, and Y. Mitsufuji. Diffusion-based speech enhancement with joint generative and predictive decoders. In *Proc. ICASSP*, pages 12951–12955, 2024.

[29] H. Choi, J. Kim, J. Huh, A. Kim, J. Ha, and K. Lee. Phase-aware speech enhancement with deep complex U-Net. In *Proc. ICLR*, 2018.

[30] S. Pascual, A. Bonafonte, and J. Serrà. SEGAN: Speech enhancement generative adversarial network. In *Proc. Interspeech*, pages 3642–3646, 2017.

[31] D. Wang and J. Chen. Supervised speech separation based on deep learning: an overview. *IEEE/ACM TASLP*, 26(10):1702–1726, 2018.

[32] F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. Le Roux, J. R Hershey, and B. Schuller. Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR. In *Proc. International conference on latent variable analysis and signal separation*, pages 91–99. Springer, 2015.

[33] A. Graves. *Long Short-Term Memory*, pages 37–45. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[34] J. Smith and P. Gossett. A flexible sampling-rate conversion method. In *Proc. ICASSP*, volume 9, pages 112–115, 1984.

[35] Y.-H. Tu, J. Du, and C.-H. Lee. Speech enhancement based on teacher–student deep learning using improved speech presence probability for noise-robust speech recognition. *IEEE/ACM TASLP*, 27(12):2080–2091, 2019.

[36] H. Huang, Y. Peng, J. Yang, W. Xia, and G. Gui. Fast beamforming design via deep learning. *IEEE Transactions on Vehicular Technology*, 69(1):1065–1069, 2020.

[37] H.-Y. Lee, J.-W. Cho, M. Kim, and H.-M. Park. Dnn-based feature enhancement using doa-constrained ica for robust speech recognition. *IEEE Signal Processing Letters*, 23(8):1091–1095, 2016.

[38] Y. Fu, K. Inoue, D. Lala, K. Yamamoto, C. Chu, and T. Kawahara. Dual variational generative model and auxiliary retrieval for empathetic response generation by conversational robot. *Advanced Robotics*, 37(21):1406–1418, 2023.

[39] Y. Fu, K. Inoue, D. Lala, K. Yamamoto, C. Chu, and T. Kawahara. Improving empathetic response generation with retrieval based on emotion recognition. In *Proc. IWSDS*, 2023.

[40] Y. Fu, K. Inoue, C. Chu, and T. Kawahara. Reasoning before responding: Integrating commonsense-based causality explanation for empathetic response generation. In *Proc. SIGDIAL*, pages 645–656, 2023.

[41] Y. Fu, S. Okada, L. Wang, L. Guo, Y. Song, J. Liu, and J. Dang. Consk-gcn: conversational semantic-and knowledge-oriented graph convolutional network for multimodal emotion recognition. In *Proc. ICME*, pages 1–6. IEEE, 2021.

[42] Y. Fu, S. Okada, L. Wang, L. Guo, Y. Song, J. Liu, and J. Dang. Context-and knowledge-aware graph convolutional network for multimodal emotion recognition. *IEEE multimedia*, 29(3):91–100, 2022.

[43] S. Dang, T. Matsumoto, Y. Takeuchi, H. Kudo, T. Tsuboi, Y. Tanaka, and M. Katsuno. Using self-learning representations for objective assessment of patient voice in dysphonia. In *Proc. APSIPA ASC*, pages 359–363, 2022.

[44] S. Dang, T. Matsumoto, Y. Takeuchi, T. Tsuboi, Y. Tanaka, D. Nakatsubo, S. Maesawa, R. Saito, M. Katsuno, and H. Kudo. Developing vocal system impaired patient-aimed voice quality assessment approach using asr representation-included multiple features, 2024.

[45] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In *Proc. NIPS*, volume 33, pages 12449–12460, 2020.

[46] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *Proc. ICML*, page 369–376, 2006.

[47] W. Hsu, B. Bolte, Y. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed. HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. *IEEE/ACM TASLP*, 29:3451–3460, 2021.

[48] S. Chen, Z. Wang, C.and Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei. WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing. *IEEE JSTSP*, 16(6):1505–1518, 2022.

[49] S.-A. Rebuffi, H. Bilen, and A. Vedaldi. Learning multiple visual domains with residual adapters. In *Proc. NIPS*, volume 30, 2017.

[50] Y. Gao, H. Shi, C. Chu, and T. Kawahara. Speech emotion recognition with multi-level acoustic and semantic information extraction and interaction.

[51] Y. Gao, H. Shi, C. Chu, and T. Kawahara. Enhancing two-stage finetuning for speech emotion recognition using adapters. In *Proc. ICASSP*, pages 11316–11320. IEEE, 2024.

[52] Y. Qian, X. Gong, and H. Huang. Layer-Wise Fast Adaptation for End-to-End Multi-Accent Speech Recognition. *IEEE/ACM TASLP*, 30:2842–2853, 2022.

[53] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv*, 2021.

[54] D. S. Williamson, Y. Wang, and D. Wang. Complex ratio masking for joint enhancement of magnitude and phase. In *Proc. ICASSP*, pages 5220–5224, 2016.

[55] H. Yin, H. Shi, L. Wang, L. Qiang, S. Li, M. Ge, G. Zhang, and J. Dang. Simultaneous progressive filtering-based monaural speech enhancement. In *Neural Information Processing*, pages 213–221, Cham, 2021. Springer International Publishing.

[56] Y. Xu, J. Du, Z. Huang, L.-R. Dai, and C.-H. Lee. Multi-objective learning and mask-based post-processing for deep neural network based speech enhancement. In *Proc. Interspeech*, pages 1508–1512, 2015.

[57] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux. Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. In *Proc. ICASSP*, pages 708–712, 2015.

[58] F. Weninger, J. R. Hershey, J. Le Roux, and B. Schuller. Discriminatively trained recurrent neural networks for single-channel speech separation. In *Proc. GlobalSIP*, pages 577–581, 2014.

[59] X. Li and R. Horaud. Multichannel speech enhancement based on time-frequency masking using subband long short-term memory. In *Proc. WASPAA*, pages 298–302, 2019.

[60] Q. Wang, J. Du, L.-R. Dai, and C.-H. Lee. Joint noise and mask aware training for DNN-based speech enhancement with sub-band features. In *Proc. HSCMA*, pages 101–105, 2017.

[61] X. Hao, X. Su, R. Horaud, and X. Li. Fullsubnet: A full-band and sub-band fusion model for real-time single-channel speech enhancement. In *Proc. ICASSP*, pages 6633–6637, 2021.

[62] Y. Xu, J. Du, Z. Huang, L. Dai, and C. Lee. Multi-objective Learning and Mask-based Post-processing for Deep Neural Network based Speech Enhancement. In *Proc. Interspeech*, pages 1508–1512, 2015.

[63] T. Menne, R. Schlüter, and H. Ney. Investigation into Joint Optimization of Single Channel Speech Enhancement and Acoustic Modeling for Robust ASR. In *Proc. ICASSP*, pages 6660–6664, 2019.

[64] D. Bagchi, P. Plantinga, A. Stiff, and E. Fosler-Lussier. Spectral Feature Mapping with MIMIC Loss for Robust Speech Recognition. In *Proc. ICASSP*, pages 5609–5613, 2018.

[65] H. Shi, L. Wang, S. Li, C. Ding, M. Ge, N. Li, J. Dang, and H. Seki. Singing voice extraction with attention-based spectrograms fusion. In *Proc. Interspeech*, pages 2412–2416, 2020.

[66] Y. Xu, C. Weng, L. Hui, J. Liu, M. Yu, D. Su, and D. Yu. Joint training of complex ratio mask based beamformer and acoustic model for noise robust asr. In *Proc. ICASSP*, pages 6745–6749, 2019.

[67] T. Gao, J. Du, L. Dai, and C. Lee. Joint training of front-end and back-end deep neural networks for robust speech recognition. In *Proc. ICASSP*, pages 4375–4379, 2015.

[68] Z. Wang and D. Wang. A joint training framework for robust automatic speech recognition. *IEEE/ACM TASLP*, 24(4):796–806, 2016.

[69] L. Dong, S. Xu, and B. Xu. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *Proc. ICASSP*, pages 5884–5888. IEEE, 2018.

[70] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Proc. NIPS*, pages 5998–6008, 2017.

[71] C. Veaux, J. Yamagishi, and S. King. The voice bank corpus: Design, collection and data analysis of a large regional accent speech database. In *Proc. O-COCOSDA/CASLRE*, pages 1–4, 2013.

[72] J. Thiemann, N. Ito, and E. Vincent. The diverse environments multi-channel acoustic noise database: A database of multichannel environmental noise recordings. *J. Acoust. Soc. Am*, 133(5):3591–3591, 2013.

[73] Y. Wang, J. Zhang, S. Chen, W. Zhang, Z. Ye, X. Zhou, and L. Dai. A study of multichannel spatiotemporal features and knowledge distillation on robust target speaker extraction. In *Proc. ICASSP*, pages 431–435, 2024.

[74] K. Kinoshita, M. Delcroix, S. Gannot, E. A. P. Habets, R. Haeb-Umbach, W. Kellermann, V. Leutnant, R. Maas, T. Nakatani, B. Raj, A. Sehr, and T. Yoshioka. A summary of the REVERB challenge: State-of-the-art and remaining challenges in reverberant speech processing research. *EURASIP Journal on Advances in Signal Processing*, 2016(1):7, 2016.

[75] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. In *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*, pages 1–5, 2017.

[76] Z. Tian, J. Yi, J. Tao, Y. Bai, and Z. Wen. Self-Attention Transducers for End-to-End Speech Recognition. In *Proc. Interspeech*, pages 4395–4399, 2019.

[77] Y. Hu and P. C. Loizou. Evaluation of objective quality measures for speech enhancement. *IEEE TASLP*, 16(1):229–238, 2008.

[78] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra. Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In *Proc. ICASSP*, volume 2, pages 749–752, 2001.

[79] D. Klatt. Prediction of perceived phonetic distance from critical-band spectra: A first step. In *Proc. ICASSP*, volume 7, pages 1278–1281, 1982.

[80] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen. An algorithm for intelligibility prediction of time–frequency weighted noisy speech. *IEEE TASLP*, 19(7):2125–2136, 2011.

[81] N. Zheng and X.-L. Zhang. Phase-aware speech enhancement based on deep neural networks. *IEEE/ACM TASLP*, 27(1):63–76, 2019.

[82] D. Takeuchi, K. Yatabe, Y. Koizumi, Y. Oikawa, and N. Harada. Effect of spectrogram resolution on deep-neural-network-based speech enhancement. *Acoustical Science and Technology*, 41(5):769–775, 2020.

[83] S. Cheung and J.S. Lim. Combined multi-resolution (wideband/narrowband) spectrogram. In *Proc. ICASSP*, pages 457–460 vol.1, 1991.

[84] A. V. Oppenheim. Speech spectrograms using the fast fourier transform. *IEEE Spectrum*, 7(8):57–62, 1970.

[85] Y. Fu, H. Song, T. Zhao, and T. Kawahara. Enhancing personality recognition in dialogue by data augmentation and heterogeneous conversational graph networks. In *Proc. IWSDS*, 2024.

[86] L. Qiang, H. Shi, M. Ge, H. Yin, N. Li, L. Wang, S. Li, and J. Dang. Speech dereverberation based on scale-aware mean square error loss. In *Neural Information Processing*, pages 55–63, Cham, 2021. Springer International Publishing.

[87] Y. Koizumi, N. Harada, and Y. Haneda. Trainable adaptive window switching for speech enhancement. In *Proc. ICASSP*, pages 616–620, 2019.

[88] C. Xu, W. Rao, E. S. Chng, and H. Li. Spex: Multi-scale time domain speaker extraction network. *IEEE/ACM TASLP*, 28:1370–1384, 2020.

[89] X. Xiang, X. Zhang, and H. Chen. A convolutional network with multi-scale and attention mechanisms for end-to-end single-channel speech enhancement. *IEEE Signal Processing Letters*, 28:1455–1459, 2021.

[90] M. Ge, L. Wang, N. Li, H. Shi, J. Dang, and X. Li. Environment-Dependent Attention-Driven Recurrent Convolutional Neural Network for Robust Speech Enhancement. In *Proc. Interspeech*, pages 3153–3157, 2019.

[91] Y. Fu, L. Guo, L. Wang, Z. Liu, J. Liu, and J. Dang. A sentiment similarity-oriented attention model with multi-task learning for text-based emotion recognition. In *MMM*, pages 278–289. Springer, 2021.

[92] J. Liu, S. Chen, L. Wang, Z. Liu, Y. Fu, L. Guo, and J. Dang. Multimodal emotion recognition with capsule graph convolutional based representation fusion. In *Proc. ICASSP*, pages 6339–6343. IEEE, 2021.

[93] K. Tan and D. Wang. A Convolutional Recurrent Neural Network for Real-Time Speech Enhancement. In *Proc. Interspeech*, pages 3229–3233, 2018.

[94] H. Shi, M. Mimura, L. Wang, J. Dang, and T. Kawahara. Time-domain speech enhancement assisted by multi-resolution frequency encoder and decoder. In *Proc. ICASSP*, pages 1–5, 2023.

[95] A. Défossez. Hybrid spectrogram and waveform source separation. In *Proc. ISMIR*, 2021.

[96] C. Chen, N. Hou, Y. Hu, S. Shirol, and E. S. Chng. Noise-robust speech recognition with 10 minutes unparalleled in-domain data. In *Proc. ICASSP*, pages 4298–4302, 2022.

[97] C. Fan, J. Yi, J. Tao, Z. Tian, B. Liu, and Z. Wen. Gated recurrent fusion with joint training framework for robust end-to-end speech recognition. *IEEE/ACM TASLP*, 29:198–209, 2021.

[98] Y. Hu, N. Hou, C. Chen, and E. S. Chng. Interactive feature fusion for end-to-end noise-robust speech recognition. In *Proc. ICASSP*, pages 6292–6296, 2022.

[99] H. Shi, L. Wang, S. Li, C. Fan, J. Dang, and T. Kawahara. Spectrograms fusion-based end-to-end robust automatic speech recognition. In *Proc. APSIPA ASC*, pages 438–442, 2021.

[100] L. Dong, S. Xu, and B. Xu. Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition. In *Proc. ICASSP*, pages 5884–5888, 2018.

[101] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang. Conformer: Convolution-augmented Transformer for Speech Recognition. In *Proc. Interspeech*, pages 5036–5040, 2020.

[102] C. KA Reddy, V. Gopal, and R. Cutler. Dnsmos: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors. In *Proc. ICASSP*, pages 6493–6497. IEEE, 2021.

[103] C. KA Reddy, V. Gopal, and R. Cutler. Dnsmos p.835: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors. In *Proc. ICASSP*. IEEE, 2022.

[104] L. Guo, L. Wang, J. Dang, Y. Fu, J. Liu, and S. Ding. Emotion recognition with multimodal transformer fusion framework based on acoustic and lexical information. *IEEE MultiMedia*, 29(2):94–103, 2022.

[105] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: An asr corpus based on public domain audio books. In *Proc. ICASSP*, pages 5206–5210, 2015.

[106] D. Snyder, G. Chen, and D. Povey. MUSAN: A Music, Speech, and Noise Corpus, 2015.

[107] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *Proc. Interspeech*, pages 2613–2617, 2019.

[108] Y. Luo and N. Mesgarani. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM TASLP*, 27(8):1256–1266, 2019.

[109] S.-J. Chen, A. S. Subramanian, H. Xu, and S. Watanabe. Building state-of-the-art distant speech recognition using the CHiME-4 challenge with a setup of speech enhancement baseline, 2018.

[110] Y. Yang, P. Wang, and D. Wang. A Conformer Based Acoustic Model for Robust Automatic Speech Recognition, 2022.

[111] Z.-Q. Wang, P. Wang, and D. Wang. Complex spectral mapping for single- and multi-channel speech enhancement and robust asr. *IEEE/ACM TASLP*, 28:1778–1787, 2020.

[112] Y. Hu, N. Hou, C. Chen, and E. S. Chng. Dual-Path Style Learning for End-to-End Noise-Robust Speech Recognition, 2023.

[113] X. Chang, T. Maekaku, Y. Fujita, and S. Watanabe. End-to-End Integration of Speech Recognition, Speech Enhancement, and Self-Supervised Learning Representation, 2022.

[114] F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. Le Roux, J. R. Hershey, and B. Schuller. Speech Enhancement with LSTM Recurrent Neural Networks and its Application to Noise-Robust ASR. In *Proc. Latent Variable Analysis and Signal Separation*, pages 91–99, 2015.

[115] R. Fan, Y. Zhu, J. Wang, and A. Alwan. Towards Better Domain Adaptation for Self-Supervised Models: A Case Study of Child ASR. *IEEE JSTSP*, 16(6):1242–1252, 2022.

[116] X. Cui, V. Goel, and B. Kingsbury. Data Augmentation for Deep Neural Network Acoustic Modeling. *IEEE/ACM TASLP*, 23(9):1469–1477, 2015.

[117] T. Song, Q. Xu, M. Ge, L. Wang, H. Shi, Y. Lv, Y. Lin, and J. Dang. Language-specific Characteristic Assistance for Code-switching Speech Recognition. In *Proc. Interspeech*, pages 3924–3928, 2022.

[118] S. Wang, W. Li, S. Siniscalchi, and C. Lee. A Cross-Task Transfer Learning Approach to Adapting Deep Speech Enhancement Models to Unseen Background Noise Using Paired Senone Classifiers. In *Proc. ICASSP*, pages 6219–6223, 2020.

[119] Q. Xu, T. Song, L. Wang, H. Shi, Y. Lin, Y. Lv, M. Ge, Q. Yu, and J. Dang. Self-Distillation Based on High-level Information Supervision for Compressing End-to-End ASR Model. In *Proc. Interspeech*, pages 1716–1720, 2022.

[120] S. Sun, C. Yeh, M. Hwang, M. Ostendorf, and L. Xie. Domain Adversarial Training for Accented Speech Recognition. In *Proc. ICASSP*, pages 4854–4858, 2018.

[121] Y. Gao, S. Okada, L. Wang, J. Liu, and J. Dang. Domain-invariant feature learning for cross corpus speech emotion recognition. In *Proc. ICASSP*, pages 6427–6431. IEEE, 2022.

[122] Y. Hu, C. Chen, R. Li, Q. Zhu, and E. Chng. Gradient Remedy for Multi-Task Learning in End-to-End Noise-Robust Speech Recognition. In *Proc. ICASSP*, pages 1–5, 2023.

[123] H. Shi, M. Mimura, and T. Kawahara. Waveform-domain speech enhancement using spectrogram encoding for robust speech recognition. *IEEE/ACM TASLP*, 32:3049–3060, 2024.

[124] B. Thomas, S. Kessler, and S. Karout. Efficient Adapter Transfer of Self-Supervised Speech Models for Automatic Speech Recognition. In *Proc. ICASSP*, pages 7102–7106, 2022.

[125] Z. Yu, Y. Zhang, K. Qian, C. Wan, Y. Fu, Y. Zhang, and Y. C. Lin. Master-ASR: Achieving Multilingual Scalability and Low-Resource Adaptation in ASR with Modular Learning. In *Proc. Machine Learning Research*, volume 202, pages 40475–40487, 2023.

[126] C. H. H. Yang, B. Li, Y. Zhang, N. Chen, R. Prabhavalkar, T. N. Sainath, and T. Strohman. From English to More Languages: Parameter-Efficient Model Reprogramming for Cross-Lingual Speech Recognition. In *Proc. ICASSP*, pages 1–5, 2023.

[127] K. M. Sathyendra, T. Muniyappa, F. J. Chang, J. Liu, J. Su, G. P. Strimel, A. Mouchtaris, and S. Kunzmann. Contextual Adapters for Personalized Speech Recognition in Neural Transducers. In *Proc. ICASSP*, pages 8537–8541, 2022.

[128] T. Munkhdalai, Z. Wu, G. Pundak, K. C. Sim, J. Li, P. Rondon, and T. N. Sainath. NAM+: Towards Scalable End-to-End Contextual Biasing for Adaptive ASR. In *Proc. SLT*, pages 190–196, 2023.

[129] Y. Yang, H. Shi, Y. Lin, M. Ge, L. Wang, Q. Hou, and J. Dang. Adaptive attention network with domain adversarial training for multi-accent speech recognition. In *Proc. ISCSLP*, pages 6–10, 2022.

[130] J. Huang, K. Ganesan, S. Maiti, Y. Min Kim, X. Chang, P. Liang, and S. Watanabe. FindAdaptNet: Find and Insert Adapters by Learned Layer Importance. In *Proc. ICASSP*, pages 1–5, 2023.

[131] A. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. In *arXiv*, 2019.

[132] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proc. CVPR*, June 2020.

[133] J. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, B. Piot, k. kavukcuoglu, R. Munos, and M. Valko. Bootstrap your own latent - a new approach to self-supervised learning. In *Proc. NIPS*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020.

[134] S. w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. Jeff Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K. t. Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H. y. Lee. Superb: Speech processing universal performance benchmark. In *arXiv*, 2021.

[135] A. Mohamed, H.-y. Lee, L. Borgholt, J. D. Havtorn, J. Edin, C. Igel, K. Kirchhoff, S.-W. Li, K. Livescu, L. Maaløe, T. N. Sainath, and S. Watanabe. Self-supervised speech representation learning: A review. *IEEE JSTSP*, 16(6):1179–1210, 2022.

[136] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli. data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language. In *Proc. ICML*, volume 162, pages 1298–1312, 17–23 Jul 2022.

[137] A. T. Liu, S.-w. Yang, P.-H. Chi, P.-c. Hsu, and H.-y. Lee. Mockingjay: Unsupervised Speech Representation Learning with Deep Bidirectional Transformer Encoders. In *Proc. ICASSP*, pages 6419–6423, 2020.

[138] S. Kessler, B. Thomas, and S. Karout. An Adapter Based Pre-Training for Efficient and Scalable Self-Supervised Speech Representation Learning. In *Proc. ICASSP*, pages 3179–3183, 2022.

[139] Y. Wang and D. Wang. A structure-preserving training target for supervised speech separation. In *Proc. ICASSP*, pages 6107–6111, 2014.

[140] Y. Wang, A. Narayanan, and D. Wang. On training targets for supervised speech separation. *IEEE/ACM TASLP*, 22(12):1849–1858, 2014.

[141] C.-Y. Li, P.-C. Yuan, and H.-Y. Lee. What does a network layer hear? analyzing hidden representations of end-to-end asr through speech synthesis. In *Proc. ICASSP*, pages 6434–6438, 2020.

[142] J. Richter, S. Welker, J.-M. Lemercier, B. Lay, and T. Gerkmann. Speech Enhancement and Dereverberation With Diffusion-Based Generative Models. *IEEE/ACM TASLP*, 31:2351–2364, 2023.

[143] H. Shi, L. Wang, S. Li, J. Dang, and T. Kawahara. Monaural Speech Enhancement Based on Spectrogram Decomposition for Convolutional Neural Network-sensitive Feature Extraction. In *Proc. Interspeech*, pages 221–225, 2022.

[144] S.-W. Fu, C. Yu, T.-A. Hsieh, P. Plantinga, M. Ravanelli, X. Lu, and Y. Tsao. Metricgan+: An improved version of metricgan for speech enhancement, 2021.

[145] A. Li, C. Zheng, L. Zhang, and X. Li. Glance and gaze: A collaborative learning framework for single-channel speech enhancement. *Applied Acoustics*, 187:108499, 2022.

[146] H. Shi, L. Wang, S. Li, J. Dang, and T. Kawahara. Subband-based spectrogram fusion for speech enhancement by combining mapping and masking approaches. In *Proc. APSIPA ASC*, pages 286–292, 2022.

[147] H. Shi, Y. Shu, L. Wang, J. Dang, and T. Kawahara. Fusing multiple bandwidth spectrograms for improving speech enhancement. In *Proc. APSIPA ASC*, pages 1938–1943, 2022.

[148] S. Pascual, A. Bonafonte, and J. Serrà. SEGAN: Speech Enhancement Generative Adversarial Network. In *Proc. Interspeech*, pages 3642–3646, 2017.

[149] Y.-J. Lu, Z.-Q. Wang, S. Watanabe, A. Richard, C. Yu, and Y. Tsao. Conditional Diffusion Probabilistic Model for Speech Enhancement. In *Proc. ICASSP*, pages 7402–7406, 2022.

[150] S. Leglaive, L. Girin, and R. Horaud. A variance modeling framework based on variational autoencoders for speech enhancement. In *Proc. MLSP*, pages 1–6, 2018.

[151] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Proc. NIPS*, volume 32. Curran Associates, Inc., 2019.

[152] D. Kingma, T. Salimans, B. Poole, and J. Ho. Variational diffusion models. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Proc. NIPS*, volume 34, pages 21696–21707. Curran Associates, Inc., 2021.

[153] N. Kamo, M. Delcroix, and T. Nakatani. Target Speech Extraction with Conditional Diffusion Model. In *Proc. Interspeech*, pages 176–180, 2023.

[154] H. Shi, N. Kamo, M. Delcroix, T. Nakatani, and S. Araki. Ensemble inference for diffusion model-based speech enhancement. In *Proc. ICASSPW*, pages 735–739, 2024.

[155] H. Yen, F. G. Germain, G. Wichern, and J. L. Roux. Cold diffusion for speech enhancement. In *Proc. ICASSP*, pages 1–5, 2023.

[156] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *Proc. ICLR*, 2021.

[157] H. Shi and T. Kawahara. Dual-path Adaptation of Pretrained Feature Extraction Module for Robust Automatic Speech Recognition. In *Proc. Interspeech*, 2022.

[158] H. Shi and T. Kawahara. Exploration of adapter for noise robust automatic speech recognition. *arXiv preprint arXiv:2402.18275*, 2024.

List of Publications

Refereed International Journal Papers

1) Hao Shi, Masato Mimura, Tatsuya Kawahara: Waveform-domain Speech Enhancement Using Spectrogram Encoding for Robust Speech Recognition, IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol.32, pp.3049–3060, 2024. (**Chapter 5**)

Refereed International Conference Papers

2) Hao Shi, and Tatsuya Kawahara: Dual-path Adaptation of Pretrained Feature Extraction Module for Robust Automatic Speech Recognition, in Proc. INTERSPEECH, 2024. (**Chapter 6**,)

3) Hao Shi, Naoyuki Kamo, Marc Delcroix, Tomohiro Nakatani, and Shoko Araki: Ensemble Inference for Diffusion Model-based Speech Enhancement, in Proc. ICASSPW, pp.735-739, 2024.

4) Hao Shi, Kazuki Shimada, Masato Hirano, Takashi Shibuya, Yuichiro Koyama, Zhi Zhong, Shusuke Takahashi, Tatsuya Kawahara, and Yuki Mitsufuji: Diffusion-Based Speech Enhancement with Joint Generative and Predictive Decoders, in Proc. ICASSP, pp.12951–12955, 2024.

5) Hao Shi, Masato Mimura, Longbiao Wang, Jianwu Dang, and Tatsuya Kawahara: Time-Domain Speech Enhancement Assisted by Multi-Resolution Frequency Encoder and Decoder, in Proc. ICASSP, pp.1–5, 2023. (**Chapter 5**)

6) Hao Shi, Yuchun Shu, Longbiao Wang, Jianwu Dang, and Tatsuya Kawahara: Fusing Multiple Bandwidth Spectrograms for Improving Speech Enhancement, in Proc. APSIPA ASC, pp.1935–1940, 2022. (**Chapter 4**)

7) Hao Shi, Longbiao Wang, Sheng Li, Jianwu Dang, and Tatsuya Kawahara: Subband-Based Spectrogram Fusion for Speech Enhancement by Combining Mapping and Masking Approaches, in Proc. APSIPA ASC, pp.1935–1940, 2022. (**Chapter 3**)

8) Hao Shi, Longbiao Wang, Sheng Li, Jianwu Dang, and Tatsuya Kawahara: Monaural speech enhancement based on spectrogram decomposition for convolutional neural network-sensitive feature extraction, in Proc. INTERSPEECH, pp.221–225, 2022. (**Chapter 4**)

9) Hao Shi, Longbiao Wang, Sheng Li, Cunhang Fan, Jianwu Dang, and Tatsuya Kawahara: Spectrograms Fusion-based End-to-end Robust Automatic Speech Recognition, in Proc. APSIPA ASC, pp.438–442, 2021. (**Chapter 3**)

Refereed Technical Reports

10) Hao Shi, and Tatsuya Kawahara: Investigation of Adapter for Automatic Speech Recognition in Noisy Environment, in SIG Technical Reports, pp.1–6, 2023. (**Chapter 6**)