12-2021

Deep Learning Based Speech Enhancement and Its Application to Speech Recognition

Ju Lin
jul@clemson.edu

Deep Learning based Speech Enhancement and Its Application to Speech Recognition

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Computer Engineering

by
Ju Lin
December 2021

Accepted by:
Dr. Melissa C. Smith, Committee Chair
Dr. Kuang-Ching Wang
Dr. Yingjie Lao
Dr. Jerome McClendon

Abstract

Speech enhancement is the task that aims to improve the quality and the intelligibility of a speech signal that is degraded by ambient noise and room reverberation. Speech enhancement algorithms are used extensively in many audio- and communication systems, including mobile handsets, speech recognition, speaker verification systems and hearing aids. Recently, deep learning has achieved great success in many applications, such as computer vision, nature language processing and speech recognition. Speech enhancement methods have been introduced that use deep-learning techniques, as these techniques are capable of learning complex hierarchical functions using large-scale training data. This dissertation investigates the deep learning based speech enhancement and its application to robust Automatic Speech Recognition (ASR).

We start our work by exploring generative adversarial network (GAN) based speech enhancement. We explore the techniques to extract information about the noise to aid in the reconstruction of the speech signals. The proposed framework, referred to as ForkGAN, is a novel general adversarial learning-based framework that combines deep-learning with conventional noise reduction techniques. We further extend ForkGAN to M-ForkGAN, which integrates feature mapping and mask learning into a unified framework using ForkGAN. Another variant of ForkGAN, named S-ForkGAN, operates on spectral-domain features, which could directly apply to ASR. Systematic evaluations demonstrate the effectiveness of the proposed approaches.

Then, we propose a novel multi-stage learning speech enhancement system. Each stage comprises a self-attention (SA) block followed by stacks of temporal convolutional network (TCN) blocks with doubling dilation factors. Each stage generates a prediction that is refined in a subsequent stage. A fusion block is inserted at the input of later stages to re-inject original information. Moreover, we design several multi-scale architectures with perceptual loss. Experiments show that our proposed architectures can achieve the state of the art performance on several public datasets.

Recently, modeling to learn the acoustic noisy-clean speech mapping has been enhanced by including auxiliary information such as visual cues, phonetic and linguistic information, and speaker information. We propose a novel speaker-aware speech enhancement (SASE) method that extracts speaker information from a clean reference using long short-term memory (LSTM) layers, and then uses a convolutional recurrent neural network (CRN) to embed the extracted speaker information. The SASE framework is extended with a self-attention mechanism. It is shown that a few seconds of clean reference speech is sufficient, and that the proposed SASE method performs well for a wide range of scenarios.

Even though speech enhancement methods that are based on deep learning have demonstrated state-of-the-art performance when compared with conventional methodologies, current deep learning approaches heavily rely on supervised learning, which requires a large number of noisy- and clean-speech sample pairs for training. This is generally not practical in a realistic environment. One cannot simultaneously obtain both noisy and clean speech samples. Thus, most speech enhancement approaches are trained with simulated speech and clean targets. In addition, it would be hard to collect large-scale dataset for the low-resource languages. We propose a novel noise-to-noise speech enhancement (N2N-SE) method that addresses the parallel noisy-clean training data issue, we leverage signal reconstruction techniques by only using corrupted speech. The proposed N2N-SE framework includes a noise conversion module that is an auto-encoder that learns to mix noise with speech, and a speech enhancement module, that learns to reconstruct corrupted speech signals.

In addition to additive noise, speech is also affected by reverberation, which is caused by the attenuated and delayed reflections of sound waves. These distortions, particularly when combined, can severely degrade speech intelligibility for human listeners and impact applications, e.g., automatic speech recognition (ASR) and speaker recognition. Thus, effective speech denoising and dereverberation will benefit both speech processing applications and human listeners. We investigate the deep-learning based approaches for both speech dereverberation and speech denoising using the cascade Conformer architecture. The experimental results show that the proposed cascade Conformer can be effective to suppress the noise and reverberation.

Dedication

This dissertation is dedicated to my family.

Acknowledgments

I would like to give my sincerest gratitude to many people who helped me along my path to completing this dissertation.

First, I would like to express my deepest gratitude to my advisors Prof. Melissa C. Smith and Prof. Kuang-Ching Wang for their continuous support during my Ph.D. study. Dr. Smith provided me with an opportunity to study at Clemson University and always gave me continuous guidance to my research papers and presentations. Moreover, Dr. Smith always gave me great freedom to explore the research topic that I am most interested in. Many thanks to Prof. Wang funded me throughout my Ph.D. study. Prof. Wang always gave me high-level research insights, and continuous guidance on my weekly reports, especially for the CARD project.

Second, I would like to show my great gratitude to Dr. Adriaan J van Wijngaarden, who gave me invaluable feedback on my dissertation and research papers. Although I haven't had a chance to meet Adriaan in-person, his insights on research problems, passionate attitude and patient guidance help me a lot to make progress on my research. I really appreciate Adriaan's effort and time on my research papers.

Besides my advisors, I would like to thank my committee members for reviewing my dissertation and giving me many suggestions: Prof. Yingjie Lao and Prof. Jerome McClendon.

In addition, I also would like to thank my advisor Prof. Jinsong Zhang during my master study. Thanks Prof. Zhang provided me with an opportunity to study speech processing and computer-aid language learning at SAIT Lab. I have been learning from him to be kind and patient when communicating with other people. He also gave me sincere encouragement and confidence during my Ph.D. application.

During my Ph.D. study, I am fortunate to have two research internships at Facebook. I would like to thank my mentors Dr. Yun Wang and Dr. Kaustubh Kalgaonkar for giving me the opportunity to have two wonderful internships at Facebook AI speech team. Moreover, I am so excited that I have earned an opportunity to return to the speech team as a research scientist after completing my Ph.D. study.

I also would like to thank my fellow lab mates and friends at Clemson for the discussions on research and all the fun we have had in the past years at Clemson. They are Sufeng, Xiang, Jianxin, Ben, Colin, Brad and other people in my lab. I appreciate this journey with them.

Finally, I owe my greatest gratitude to my parents for always standing by my side. I would not have been possible to achieve success in my graduate study without their unconditional love and persistent support. I also want to express my deepest gratitude to Shuju, who always encourages me. I constantly learn about how to be a better person from her. I want to thank God for bringing her into my life.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In modern day life, our interactions with voice-based devices and services continue to increase, such as personal assistant devices. However, in real-world environments, speech is inevitably degraded by various noises like environmental noises such as traffic noise, alarms, other speaker's speech, and electrical noise from devices such as refrigerators, air conditioning and so forth. Besides additive background noise, reverberation is another major distortion that we encounter in daily life. Reverberation is typically caused by time-delayed reflections of sound waves in an enclosed space (e.g., a conferencing room). Monaural (single-channel) speech enhancement is the task that is used to improve the quality and the intelligibility of a speech signal that is degraded by ambient noise and reverberation. Speech enhancement is essential both for interhuman communications and for human-to-computer communication, where spoken language is converted into text by computers, a process referred to as automatic speech recognition. Monaural speech enhancement is extremely challenging [115] due to lacking of spatial information provided by a microphone array. It is used extensively in many audio- and communication systems, including mobile handsets, speaker verification systems and hearing aids. Popular classic techniques include spectral-subtraction algorithms, statistical model-based methods that use maximum-likelihood (ML) estimators, Bayesian estimators, minimum mean squared error (MMSE) methods, subspace algorithms based on single value decomposition and noise-estimation algorithms (see [65] and references therein).

Modern techniques often use deep learning. Early examples include a recurrent neural network (RNN) to model long-term acoustic characteristics [70], and a deep auto-encoder that denoises speech signals with greedy layer-oriented pre-training [66]. In [124], a deep neural network

(DNN) was used as a non-linear regression function. In [100], a convolutional recurrent neural network (CRN) was used, consisting of a convolutional encoder-decoder architecture and multiple long short-term memory (LSTM) layers that aim to capture long-context information. Other speech enhancement systems use a generative adversarial network (GAN), which is known for its ability to generate natural-looking signals in the time or frequency domain [83, 18, 3]. Recent studies consider the use of an attention mechanism [22, 44, 130, 51, 82, 129]. Self-attention [113] is an efficient context information aggregation mechanism that operates on the input sequence itself and that can be utilized for any task that has a sequential input and output. In [82], self-attention is combined with a dense convolutional neural network. A time-frequency (T-F) attention method, proposed in [129], combines time-domain and frequency-domain attention to perform denoising and dereverberation at the same time.

Recent research shows that models based on a temporal convolutional network (TCN) achieve excellent performance for text-to-speech [110], speech enhancement [81, 127, 52, 49, 64], and speech separation [69]. A TCN consists of dilated 1-D convolutions that create a large temporal receptive field with fewer parameters than other models. In [127], a speech enhancement system was proposed that uses a multi-branch TCN, in short MB-TCN, which effectively performs a split-transform-aggregate operation and enables the model to learn and determine an accurate representation by aggregating the information from each branch. In [49], the TCN used in [69] for speech separation was adapted for speech enhancement and integrated in a multi-layer encoder-decoder architecture. The use of a complex Short-Time Fourier transform (STFT) for TCN-based speech enhancement rather than magnitude or time-domain features was investigated in [52].

Recently, in addition to directly mapping the noisy signal to clean signal, modeling to learn the acoustic noisy-clean speech mapping has been enhanced by including auxiliary information such as visual cues [31], phonetic and linguistic information [59, 67], and speaker information [8]. In particular, the utilization of three kinds of broad phonetic class (BPC) information for speech enhancement can achieve notable improvements [67]. In [8], a speaker-aware deep denoising autoencoder (SaDAE) extracts speaker representation from the noisy input using a DNN model. Target speaker extraction was investigated in [119, 40, 86].

The above-mentioned methods can generally be classified as feature-mapping and mask-learning methods, which are two commonly used deep-learning approaches for single-channel speech enhancement methods for stereo data. Feature mapping approaches enhance the noisy features using a mapping network that minimizes the mean square error between the enhanced and clean features. Mask-learning approaches estimate the ideal ratio mask, the ideal binary mask or the complex ratio mask, and then use this mask to filter noisy speech signals and reconstruct the clean speech signals. Mask-learning methods usually perform better than feature mapping methods in terms of speech quality metrics [120, 6, 75].

Even though speech enhancement methods that are based on deep learning have demonstrated the state-of-the-art performance when compared with conventional methodologies, they have the following limitations: current deep learning approaches heavily rely on supervised learning, which requires a large number of noisy- and clean-speech sample pairs for training. This is generally not practical in a realistic environment. One cannot simultaneously obtain both noisy and clean speech samples. Thus, most speech enhancement approaches are trained with simulated speech and clean targets. Another case is that we cannot collect large-scale dataset for the low-resources languages. The challenge therefore is to effectively train a deep model for speech enhancement without using clean speech samples.

In addition, in daily listening environments, speech is inevitably corrupted by background noise. Besides additive noises, reverberation caused by the attenuated and delayed reflections of sound waves in a room is another major distortion that we face everyday. Besides speech denoising only, deep learning based speech enhancement under noisy-reverberant condition is also worth exploring. In [27], a DNN based approach was proposed to learn a nonlinear mapping from the log magnitude spectrum of noisy-reverberant speech to that of clean-anechoic speech. Two-stage approaches are investigated in [131], where noise and reverberation were removed in two separate stages, respectively.

1.1 Problem Formulation

The objective of a monaural speech enhancement module is to filter a received *noisy speech signal* and to generate an *enhanced signal* that is as close as possible to the original speech signal. Let $\{x(t)|t \in \mathbb{Z}\}$ denote a deterministic discrete-time data sequence that is obtained by sampling a received continuous-time noisy speech signal. $v(t)$ and $s(t)$ denote noise and clean speech signal. As such,

$$x(t) = v(t) + s(t), \tag{1.1}$$

the quantitative objective of the speech enhancement module is to output a signal $\hat{\boldsymbol{s}}$ that is as close as possible to the original speech signal $\boldsymbol{s}$.

Time-domain based speech enhancement approaches directly operate on waveform signal. For frequency-domain based speech enhancement, a short-time Fourier transform (STFT) is applied on time-domain signals. The STFT of length N of $\{x(t)\}$ with window function $w(t)$ of length N and hop-length T_h is given by

$$X_{\tau,\omega} = \sum_{n=0}^{N-1} w(n)x(\tau T_h + n) \exp\left(-j\frac{2\pi\omega n}{N}\right), \tag{1.2}$$

where τ is the index of the sliding window and $0 \leq \omega < N$ is the frequency index. In this dissertation, a Hanning window is used, where

$$w(t) = \sin^2\left(\frac{\pi t}{N-1}\right). \tag{1.3}$$

Let $\mathbf{X}$ and $\mathbf{\Omega}$ denote the STFT magnitude and phase, i.e., $\mathbf{X} = \{|X_{\tau,\omega}|\} \in \mathbb{R}^{F\times T}$ and $\mathbf{\Omega} \in \mathbb{R}^{F\times T}$, where $F = N/2 + 1$ denotes the number of frequency bins and $T = T_\ell/T_h + 1$. Similarly, the frequency-domain representation of noisy signal can be written as:

$$\mathbf{X} = \mathbf{V} + \mathbf{S}. \tag{1.4}$$

In addition to directly spectral-mapping, previous studies have shown that estimating mask usually achieve better performance in terms of speech quality metrics. Two commonly used masks are ideal ratio mask (IRM) and ideal binary mask (IBM). The IBM is a time-frequency (T-F) mask constructed from premixed signals. For each T-F unit, the corresponding mask value is set to 1 if the local SNR is greater than a local criterion (denoted as LC), otherwise it is set to 0. IBM is defined as:

$$\mathrm{IBM}(t,f) = \begin{cases} 1, & \mathrm{SNR}(t,f) > \mathrm{LC} \\ 0, & \text{otherwise}, \end{cases} \tag{1.5}$$

where $\mathrm{SNR}(t,f)$ denotes the local SNR within the T-F unit at time t and frequency f. The IRM is defined as follows:

$$\mathrm{IRM}(t,f) = \sqrt{\frac{|\boldsymbol{S}(t,f)|^2}{|\boldsymbol{S}(t,f)|^2 + |\boldsymbol{V}(t,f)|^2}}, \tag{1.6}$$

where $\boldsymbol{S}(t,f)^2$ and $\boldsymbol{V}(t,f)^2$ represent the clean speech energy and noise energy with a T-F unit, respectively. The enhanced speech can be obtained by the following equation:

$$\hat{\boldsymbol{S}} = \text{MASK}\odot\boldsymbol{X} \tag{1.7}$$

where the operator $\odot$ denotes the Hadamard product and **MASK** denotes IBM or IRM.

For the noisy-reverberant condition, let $h(t)$ denotes room impulse response (RIR) function, the noisy-reverberant speech $y(t)$ can be expressed as:

$$y(t) = v(t) + s(t) * h(t) = v(t) + z(t) \tag{1.8}$$

where $*$ stands for the convolution operator; $s(t)$ and $z(t)$ denote anechoic speech and its reverberant speech, respectively. The objective of noisy-reverberant speech enhancement is to recover $s(t)$ from the observed $y(t)$.

1.2 Dissertation Research

In this dissertation, we aim to develop monaural speech enhancement systems to improve speech intelligibility and quality of a speech signal that is degraded by ambient noise and reverberation. We have proposed several novel architectures for speech enhancement and the contributions can be summarized as follows:

- Generative adversarial network (GAN) have shown great success in many applications, e.g., speech synthesis, image processing. We first explore GAN based speech enhancement approaches. We proposed a novel GAN based speech enhancement framework, named ForkGAN, which decouples the noisy signal into speech and noise simultaneously in the time domain, and then uses the estimated noise signal to aid in the reconstruction of the speech signals. We further extend ForkGAN to M-ForkGAN, which takes the advantages of mask-learning based speech enhancement. Finally, we investigate the frequency-domain based ForkGAN in which the enhanced feature can be directly fed into ASR module.

- Recent studies show that self-attention [113] is an efficient context information aggregation mechanism. TCN is comprised of dilated 1-D convolutions that have a large temporal receptive field with fewer parameters than other models. We proposed a self-attentive temporal convolutional network (SA-TCN) for speech enhancement, in which the self-attention can be used for modeling the cross channel information and TCN is used to learn the longer context dependency. Furthermore, we also investigate a multi-stage SA-TCN, in which each stage generates a prediction that is refined in a subsequent stage.

- TCN are typically stacked multiple layers to model a longer temporal contextual field, in which the dilation rate in each block is exponentially increased. As the number of layers increases, the resulting large dilation rate makes the model pay more attention to long term dependency. However, the corresponding local information may be neglected in the higher layers. To mitigate these limitations, we propose three multi-scale TCN architectures for speech denoising. The first architecture refers to TCN-dual, which has intra-parallel dilation rates in the TCN block. The second architecture refers to TCN-flatten, in which we adopt a fixed number of dilation rates. It re-samples the feature maps to learn the representations from large effective receptive field. Instead of directly concatenating the parallel outputs that are fed into a convolutional based fusion layer, we adopt the pyramid dilated convolutions, which uses the hierarchical feature fusion (HFF) [72]. We named this architecture to TCN-pyramid. In addition, we also explored several loss functions and their combination for our proposed framework.

- Previous studies have shown that auxiliary information is also useful for improving the performance of speech enhancement. Given that it is generally possible to collect a few seconds of clean reference speech in applications, e.g., similar to a smart virtual assistant that needs a few-second clean speech record during its setup stage, or extracted from (prior) high-SNR recordings, it is worthwhile investigating how a few seconds of clean reference can be best used to improve speech enhancement performance. We propose a novel speaker-aware speech enhancement (SASE), which extracts speaker information from a clean reference using long short-term memory (LSTM) layers, and then uses a convolutional recurrent neural network (CRN) to embed the extracted speaker information. It will be shown that a few seconds of clean reference speech is sufficient, and that the proposed SASE method performs well for a wide range of scenarios.

- Most existing speech enhancement approaches rely on noisy-clean paired training data, which typically is the simulated data. This is generally not practical in a realistic environment, where only corrupted speech is available. This could have a mismatch between the training and testing. To resolve the parallel noisy-clean training data issue, we propose a novel noise-to-noise speech enhancement (N2N-SE) method that leverages signal reconstruction techniques [56, 53, 108] by only using corrupted speech.

- The combination of room reverberation and background noise is particularly disruptive for speech perception. For the noisy-reverberant condition, we compare the performance between one-stage and two-stage approaches using more advanced techniques, e.g., Conformer [26]. In addition, we also explore different training strategies for two-stage approaches.

1.3 Contributions and Outline

The reminder of this dissertation is organized as follows.

Chapter 2 explores the ways of Generative adversarial network (GAN) based speech enhancement. We propose a novel framework that decouples the noisy speech into speech and noise components and the estimated noise information is used to aid speech enhancement performance.

Chapter 3 presents the details and design architectures of the proposed self-attentive multi-stage temporal neural networks. The idea is to utilize multi-stage learning which is an effective technique to invoke multiple deep-learning modules sequentially.

Chapter 4 explores three multi-scale TCN architectures that learn the feature representations using multiple dilate rates in each TCN blocks. Several training targets are explored for the proposed architectures and we also compare the performance with top systems on the INTERSPEECH2020 DNS challenge.

Chapter 5 studies a novel speaker-aware speech enhancement (SASE) method that extracts speaker information using neural networks. We explore to use only a few seconds clean reference speech, which can be collected in the real applications and devices.

Chapter 6 investigates the speech enhancement approach using non-parallel noisy-clean paired data. Our proposed framework contains a noise conversion module and a speech enhancement module. The noise conversion module, which is an auto-encoder, generates diverse noise augmented speech data. The speech enhancement module uses the generated multiple noisy speech signal as targets to filter out original clean signals.

Chapter 7 presents a two-stage speech enhancement method where denoising and dereverberation are performed sequentially using a cascaded conformer. Multiple training strategies are investigated.

Chapter 8 concludes this dissertation and discusses future directions.

Chapter 2

Generative Adversarial Network-Based Speech Enhancement

This chapter presents the Generative adversarial network (GAN) based speech enhancement approaches. The work presented in this chapter has been published in INTERSPEECH2019 [61] and INTERSPEECH2020 [60].

2.1 Introduction

Several speech enhancement methods have been developed and refined during the last several decades, including spectral-subtraction algorithms, statistical model-based methods that use maximum-likelihood (ML) estimators, Bayesian estimators, minimum mean squared error (MMSE) methods, subspace algorithms based on single value decomposition and noise-estimation algorithms [65, 94, 4].

Recently, speech enhancement methods have been introduced that use deep-learning techniques, as these techniques are capable of learning complex hierarchical functions using training data, see [23] and references therein. In [70], a recurrent neural network (RNN) was used to model long-term acoustic characteristics, and in [66], a deep auto-encoder was used to denoise the signal by employing a greedy layer-wise pre-training-with-fine-tuning strategy. In [124], a deep neural network (DNN) was used as a non-linear regression function. In the training phase, a DNN-based regression model was trained using the log-power spectral features from pairs of noisy and clean speech data. Large training sets were used that encompassed many possible combinations of speech and noise types. Further enhancements were made to improve the DNN-based system, including global variance equalization and noise-aware training strategies. The resulting system outperforms MMSE-based techniques in terms of perceptive and objective measures. However, DNN methods require additional feature extraction steps. More recent contributions propose to use an auto-encoder that processes the received speech waveform, and a generative adversarial network (GAN) [24] that has been trained with samples from multiple speakers and a wide variety of noise conditions [83, 23]. In [79], the auto-encoder was replaced by a multi-layer feed-forward network, and the performance was determined when using an L_1- and an L_2-norm for training. In [13], it was proposed to operate a GAN on log-Mel filter bank spectra instead of waveforms, and in [98], source separation was achieved using a Wasserstein-GAN based method [2]. The latter was shown to outperform conventional generative source separation methods such as non-negative matrix factorization (NMF) algorithms [55].

2.2 GAN Concept

Let $\boldsymbol{x} \in \mathbb{R}^n$ denote a received n-sample noisy speech signal that is fed to the speech enhancement module, and let $\boldsymbol{s}$ denote the clean speech waveform. The quantitative objective of the speech enhancement module is to output a signal $\hat{\boldsymbol{s}}$ that is as close as possible to the original speech signal $\boldsymbol{s}$.

In the context of deep learning, speech enhancement can be viewed as a generative model that conditions on a noisy speech signal. Instead of learning an input-output mapping, one can also employ a generative adversarial network (GAN), which directly learns the data distribution without explicitly defining the objective function. As such, speech enhancement can be inherently embedded into a GAN-based framework to learn non-linear filtering functions. The speech enhancement generative adversarial network (SEGAN) model, which was introduced in [83], and its variants, e.g., in [73], achieve state-of-the-art performance that is comparable to DNN methods. The SEGAN model can be reformulated such that it is expressed as a combination of two networks: 1) a *generator* network $\mathbf{G}$ that extracts the latency representation $\boldsymbol{c} \in \mathbb{R}^d$ of the noisy speech signal $\boldsymbol{x} \in \mathbb{R}^n$ and 2) a *discriminator* network $\mathbf{D}$ that learns the implicit loss. Generator $\mathbf{G}$ takes a noisy speech signal $\boldsymbol{x}$ as input and uses the encoder operation $\boldsymbol{c} = \boldsymbol{\Phi}(\boldsymbol{x})$ to extract its latency representation $\boldsymbol{c} \in \mathbb{R}^d$. The generator $\mathbf{G}$ then performs the decoding operation $\boldsymbol{\Psi}(\cdot)$, given by

$$\hat{\boldsymbol{s}} = \boldsymbol{\Psi}(\boldsymbol{c}, \boldsymbol{z})\,, \tag{2.1}$$

where $\boldsymbol{z}$ is a random vector. Decoder $\boldsymbol{\Psi}(\cdot)$ concatenates vector $\boldsymbol{c}$ with a random vector $\boldsymbol{z}$ to output an enhanced signal $\hat{\boldsymbol{s}}$. Generator $\mathbf{G}$ is thus represented as $\mathbf{G}(\boldsymbol{z}, \boldsymbol{x}) = \Psi\left(\Phi\left(\boldsymbol{x}\right), \boldsymbol{z}\right)$. SEGAN employs adversarial learning to train the generator $\mathbf{G}$, and uses an L_1-norm to measure the distance between the generated and clean samples. The corresponding loss functions $\mathcal{L}_{\mathbf{D}}$ and $\mathcal{L}_{\mathbf{G}}$ are defined by

$$\begin{aligned}
\mathcal{L}_{\mathbf{D}} &= \frac{1}{2}\mathbb{E}_{\boldsymbol{x},\boldsymbol{s}\sim p_{\text{data}}(\boldsymbol{x},\boldsymbol{s})}[(\mathbf{D}(\boldsymbol{s},\boldsymbol{x})-1)^2] \\
&\quad + \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z}),\boldsymbol{s}\sim p_{\text{data}}(\boldsymbol{x})}[\mathbf{D}(\mathbf{G}(\boldsymbol{z},\boldsymbol{x}),\boldsymbol{x})^2] && (2.2) \\
\mathcal{L}_{\mathbf{G}} &= \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z}),\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[(\mathbf{D}(\mathbf{G}(\boldsymbol{z},\boldsymbol{x}),\boldsymbol{x})-1)^2] \\
&\quad + \lambda\,\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z}),\boldsymbol{s},\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x},\boldsymbol{s})}\|\mathbf{G}(\boldsymbol{z},\boldsymbol{x})-\boldsymbol{s}\|_1, && (2.3)
\end{aligned}$$

where $\mathbf{D}(\boldsymbol{s}, \boldsymbol{x})$ denotes the discriminator network and λ denotes a hyperparameter.

Several studies have shown that noise information is beneficial when using a deep neural network for speech recognition, e.g., noise-aware training [96] and speech enhancement [45]. The non-linear relationship between noisy-speech, clean-speech and the noise signal can be modeled by the non-linear layers of a DNN by directly providing the noise log-spectra as an input to the network. Note that the analysis in [96] is based on the assumption that the noise is stationary, i.e., the noise signal is fixed and is obtained using the first few frames for each sentence. However, in reality, noise could be both stationary and non-stationary. Hence, it might be useful to estimate the noise patterns and to use this information to decouple the noise and speech signals. The proposed ForkGAN framework adds a dedicated noise decoder in parallel with the speech decoder. Instead of enhancing pure speech signals, it simultaneously enhances the speech and the noise signals.

In this chapter, we explore techniques to extract the noise information to aid in the reconstruction of the speech signals. The proposed framework, referred to as ForkGAN, is a novel general adversarial learning-based framework that combines deep-learning with conventional noise reduction techniques. ForkGAN forks the received waveform after initial processing to a speech extraction module and a noise pattern identification module, respectively. As such, the speech and noise signals are effectively decoupled. The noise pattern identification module learns to generate noise signals conditioned on the input latent variables. An end-to-end speech enhancement model for input speech waveforms is achieved by introducing two auxiliary loss functions to detach the noise from the source signal: 1) a margin-based loss that pushes the speech and noise signals apart, and 2) a time-domain noise reduction loss component that combines conventional signal processing noise subtraction with the neural network predictions. As such, ForkGAN learns to generate additive noise and clean speech within the adversarial learning framework. We further extend ForkGAN to M-ForkGAN, which integrates the mask-learning and time-domain feature-mapping methods into one unified framework to take advantage of both approaches. The generated speech and noise signals are fed into two separate short-time Fourier transform (STFT) convolution 1-D layers to generate the speech and noise spectrograms, which are used to calculate the speech mask. The time-domain feature mapping component can preserve the phase information, which is useful to improve the speech quality [77]. Both ForkGAN and M-ForkGAN operate on time-domain signals, we also investigate the ForkGAN architecture on the frequency-domain signals, we refer this method to S-ForkGAN. We will detail each proposed architectures in the following sections.

2.3 ForkGAN Architecture

The ForkGAN architecture comprises two parallel GAN-based decoders for speech and noise. This is illustrated in Fig. 2.1. The generator $\mathbf{G}$ performs the following encoder and decoder operations:

- Step 1. Use the encoder operation $\boldsymbol{c} = \boldsymbol{\Phi}(\boldsymbol{x})$ to extract latent vector $\boldsymbol{c}$ from the received noisy speech signal $\boldsymbol{x}$.

- Step 2. Decouple the latent vector $\boldsymbol{c}$ and extract the two latent features $\boldsymbol{c}_s$ and $\boldsymbol{c}_v$ for the decoder by performing the linear transformations $\boldsymbol{c}_s = \boldsymbol{w}_s\boldsymbol{c}$ and $\boldsymbol{c}_v = \boldsymbol{w}_v\boldsymbol{c}$, where $\boldsymbol{c}_s$ comprises the clean speech information and $\boldsymbol{c}_v$ comprises the noise information, and where $\boldsymbol{w}_s$ and $\boldsymbol{w}_c$ are the weights of two fully connected layers.

- Step 3. Simultaneously perform the speech decoder operation $\hat{\boldsymbol{s}} = \boldsymbol{\Psi}_s(\boldsymbol{c}_s, \boldsymbol{z}_s)$ to denoise the speech and the noise decoder operation $\hat{\boldsymbol{v}} = \boldsymbol{\Psi}_v(\boldsymbol{c}_v, \boldsymbol{z}_v)$.

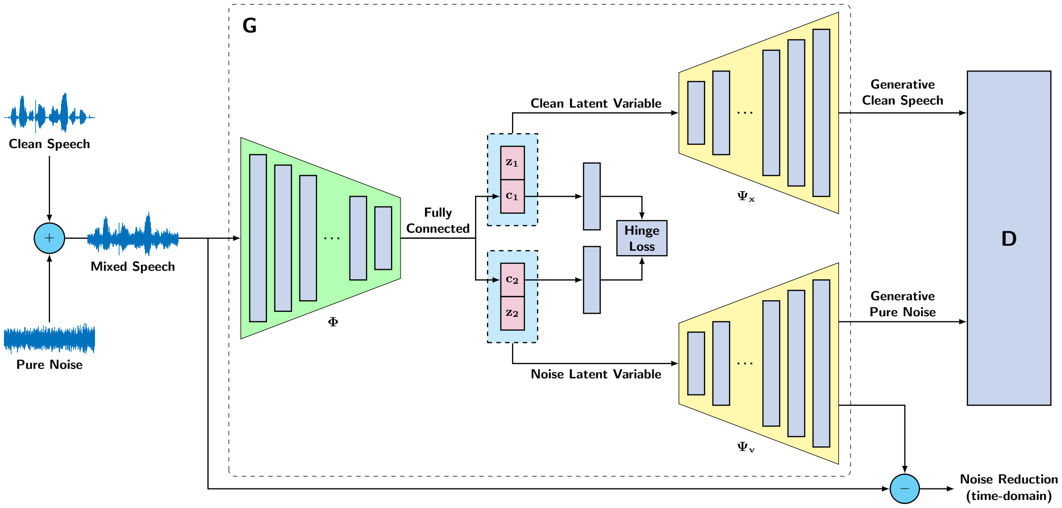


Figure 2.1: ForkGAN architecture with forked decoders for speech enhancement.

The speech decoder $\boldsymbol{\Psi}_s(\cdot)$ and the noise decoder $\boldsymbol{\Psi}_v(\cdot)$ have the same architecture and their objective is to generate the speech signal and the additive noise signal, respectively. Each decoder input concatenates an encoder-latent representation $\boldsymbol{c}$ with a random vector $\boldsymbol{z}$ that is sampled from a normal distribution $\mathcal{N}(0, I)$, and that outputs the predicted signal $\hat{\boldsymbol{s}} \in \mathbb{R}^n$, referred to as the clean speech prediction, and signal $\hat{\boldsymbol{v}} \in \mathbb{R}^n$, referred to as the noise prediction. Generator $\mathbf{G}$ has an architecture that uses convolutional operations, and both decoder layers are the inverse structure of the encoder with the same configurations. Note that each layer input is concatenated with skip connections from the encoder.

In the training phase, the aim is to minimize the difference between the enhanced signal pair ($\hat{\boldsymbol{s}}$, $\hat{\boldsymbol{v}}$) and the ground truth signal pair $(\boldsymbol{s}, \boldsymbol{v})$ by optimizing the encoder and decoder functions. Similar to SEGAN, the training procedure combines adversarial learning regularized with regression loss. We feed ForkGAN noisy speech $\boldsymbol{x}$, clean speech $\boldsymbol{s}$, and additive noise signal $\boldsymbol{v}$. During the training process, we sample two pairs of speech signal: the real pair of samples, which consists of a clean signal $\boldsymbol{s}$ and additive noise $\boldsymbol{v}$, and the fake pair of samples, which consists of the enhanced speech $\hat{\boldsymbol{s}}$ and predicted noise signal $\hat{\boldsymbol{v}}$, both conditioned on noisy speech $\boldsymbol{x}$. In adversarial learning, $\boldsymbol{s}$ and $\boldsymbol{v}$ are also used as ground truth for regression in the generator.

2.3.1 Loss Functions

In the following, we improve the basic ForkGAN framework by adding training objectives that are based on the characteristics of the proposed architecture.

ForkGAN-M – Margin-Based Loss. A max-margin-based loss function is introduced to regularize the loss model. The basic idea is to maximize the distance between the speech signal and the noise signal. The objective is to ensure that the distance between the embedded clean speech signal and the noise speech signal is larger than a predefined margin, so that the generated speech and noise are as dissimilar as possible. This is accomplished by using the Euclidean distance between the normalized embeddings and $\boldsymbol{c}_s \leftarrow \boldsymbol{c}_s/\|\boldsymbol{c}_s\|_2$ and $\boldsymbol{c}_v \leftarrow \boldsymbol{c}_v/\|\boldsymbol{c}_v\|_2$. The margin-loss function $\mathcal{L}_{\mathrm{M}}$ for each pair of a clean speech embedding and a noise embedding is specified by

$$\mathcal{L}_{\mathrm{M}} = \mathbb{E}_{\boldsymbol{x}\sim p_{\mathrm{data}}} \max(0, \Delta - \mathcal{D}(\boldsymbol{x})), \tag{2.4}$$

where Δ denotes the margin hyperparameter and

$$\mathcal{D}(\boldsymbol{x}) = \frac{1}{d}\sum_{d} \|\boldsymbol{c}_v - \boldsymbol{c}_s\|_2 \tag{2.5}$$

represents the Euclidean distance between the two normalized embeddings $\boldsymbol{c}_v$ and $\boldsymbol{c}_s$. The generator loss $\mathcal{L}_{\mathbf{G\,M}}$ incorporates the margin-based loss $\mathcal{L}_{\mathrm{M}}$ and original generator loss $\mathcal{L}_{\mathrm{G}}$, and is expressed as

$$\mathcal{L}_{\mathbf{G\,M}} = \mathcal{L}_{\mathbf{G}} + \alpha\mathcal{L}_{\mathrm{M}}, \tag{2.6}$$

where α denotes the coefficient that controls the strength of the auxiliary loss function.

ForkGAN-M-NR – Margin-Based and Noise-Reduction Loss. Spectral subtraction is historically one of the techniques used for enhancing single-channel speech. Since the noisy signal $\boldsymbol{x}$ is the sum of the desired signal value $\boldsymbol{s}$ and the noise value $\boldsymbol{v}$, the standard spectral subtraction is defined in the frequency domain by

$$\boldsymbol{S} = \boldsymbol{X} - \boldsymbol{V} \tag{2.7}$$

where $\boldsymbol{X}$, $\boldsymbol{S}$ and $\boldsymbol{V}$ are Fourier transforms of $\boldsymbol{x}$, $\boldsymbol{s}$, and $\boldsymbol{v}$, respectively. It follows from (2.7) that the accuracy of the spectral subtraction heavily depends on accurate noise spectrum estimation. Unlike conventional noise spectrum estimation, ForkGAN uses a neural network as a function approximator to estimate the noise signal. As many noise patterns are time-varying, it is easier to cancel the noise in the time domain for these non-stationary noise patterns. Since ForkGAN directly operates on time-domain waveforms, this results in a time-domain noise reduction loss of ForkGAN's training objectives. Since ForkGAN uses a neural-network-based approach to estimate the additive noise, the noise reduction can be derived by subtracting a noise prediction from the generator, i.e.,

$$\mathcal{L}_{\mathrm{NR}} = \mathbb{E}_{\boldsymbol{x}\sim p_{\mathrm{data}}} \|\boldsymbol{x} - \hat{\boldsymbol{v}} - \boldsymbol{s}\|_1. \tag{2.8}$$

The generator loss of noise and speech can be expressed as:

$$\begin{aligned}\mathcal{L}_{\mathbf{G}}^{(s)} &= \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z}),\boldsymbol{x}\sim p_{\mathrm{data}}(\boldsymbol{x})}[(\mathbf{D}(\boldsymbol{\Psi}_s(\boldsymbol{\Phi}(\boldsymbol{x}),\boldsymbol{z}_s),\boldsymbol{x})-1)^2] \\ &\quad + \lambda\,\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z}),\boldsymbol{s},\boldsymbol{x}\sim p_{\mathrm{data}}(\boldsymbol{x},\boldsymbol{s})}\|\boldsymbol{\Psi}_s(\boldsymbol{\Phi}(\boldsymbol{x}),\boldsymbol{z}_s))-\boldsymbol{s}\|_1 \qquad (2.9) \\ \mathcal{L}_{\mathbf{G}}^{(v)} &= \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z}),\boldsymbol{x}\sim p_{\mathrm{data}}(\boldsymbol{x})}[(\mathbf{D}(\boldsymbol{\Psi}_v(\boldsymbol{\Phi}(\boldsymbol{x}),\boldsymbol{z}_v),\boldsymbol{x})-1)^2] \\ &\quad + \lambda\,\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z}),\boldsymbol{v},\boldsymbol{x}\sim p_{\mathrm{data}}(\boldsymbol{x},\boldsymbol{v})}\|\boldsymbol{\Psi}_v(\boldsymbol{\Phi}(\boldsymbol{x}),\boldsymbol{z}_v)-\boldsymbol{v}\|_1, \qquad (2.10)\end{aligned}$$

The generator loss $\mathcal{L}_{\mathbf{G\,MNR}}$ incorporates the margin-based loss $\mathcal{L}_{\mathrm{M}}$ and the noise-reduction loss factor $\mathcal{L}_{\mathrm{NR}}$, and is expressed as

$$\mathcal{L}_{\mathbf{G\,MNR}} = \mathcal{L}_{\mathbf{G}}^{(s)} + \mathcal{L}_{\mathbf{G}}^{(v)} + \beta\mathcal{L}_{\mathrm{M}} + \gamma\mathcal{L}_{\mathrm{NR}}, \tag{2.11}$$

where β and γ denote coefficients that control the strength of each auxiliary loss function.

This discriminator loss function can be expressed as:

$$\begin{aligned}\mathcal{L}_{\mathbf{D}} = \frac{1}{2}\mathbb{E}_{\boldsymbol{x},\boldsymbol{s}\sim p_{\mathrm{data}}(\boldsymbol{s},\boldsymbol{x})}[(\mathbf{D}(\boldsymbol{s},\boldsymbol{x})-1)^2] + \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z}),\boldsymbol{x}\sim p_{\mathrm{data}}}[\mathbf{D}(\boldsymbol{\Psi}_s(\boldsymbol{\Phi}(\boldsymbol{x}),\boldsymbol{z}_s),\boldsymbol{x})^2] \\ + \frac{1}{2}\mathbb{E}_{\boldsymbol{v},\boldsymbol{x}\sim p_{\mathrm{data}}(\boldsymbol{v},\boldsymbol{x})}[(\mathbf{D}(\boldsymbol{v},\boldsymbol{x})-1)^2] + \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z}),\boldsymbol{v}\sim p_{\mathrm{data}}}[\mathbf{D}(\boldsymbol{\Psi}_v(\boldsymbol{\Phi}(\boldsymbol{x}),\boldsymbol{z}_v),\boldsymbol{x})^2]\end{aligned} \tag{2.12}$$

2.4 M-ForkGAN Architecture

The proposed M-ForkGAN uses a similar architecture to the ForkGAN as illustrated in Fig. 2.2. It takes a raw waveform as input, and the output of the encoder $\boldsymbol{\Phi}(\boldsymbol{x})$ is fed into two separate fully-connected layers that generate the speech latent representation $\boldsymbol{c}_s$ and the noise latent

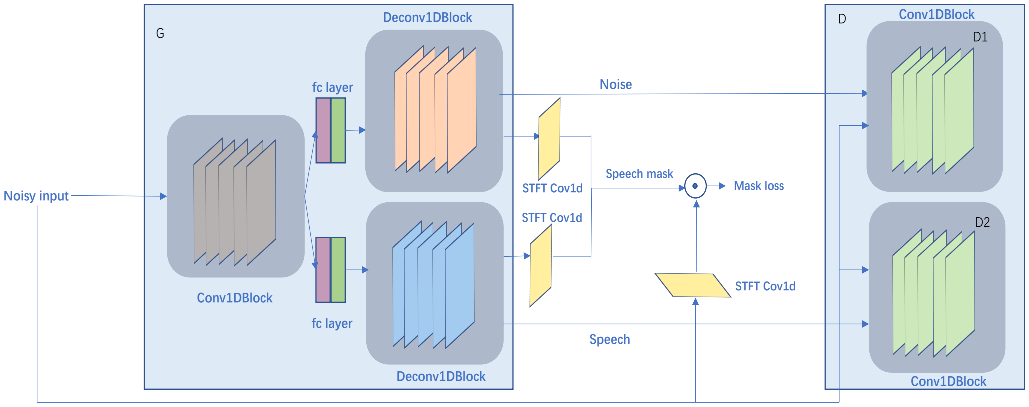


Figure 2.2: Proposed M-ForkGAN architecture.

representation $\boldsymbol{c}_v$, respectively. Each decoder input concatenates an encoder-latent representation with a random vector $\boldsymbol{z}$ that is sampled from a normal distribution $\mathcal{N}(0, I)$, and outputs the predicted time-domain speech signals $\hat{\boldsymbol{s}} = \boldsymbol{\Psi}_s([\boldsymbol{c}_s, \boldsymbol{z}_s])$ and noise signals $\hat{\boldsymbol{v}} = \boldsymbol{\Psi}_v([\boldsymbol{c}_v, \boldsymbol{z}_v])$, where $\boldsymbol{\Psi}(\cdot)$ denotes the decoding operation. The generator network also includes skip connections among encoder layers and its homologous decoding layer to avoid losing many low-level details.

Two STFT convolution 1-D layers are used to map the generated speech and noise waveforms to complex spectrograms that include both magnitude and phase components. The magnitude component will be used only. Given a window function ω of length N, the speech complex spectrogram $\hat{\boldsymbol{S}}_{t,f}$ and the noise complex spectrogram $\hat{\boldsymbol{V}}_{t,f}$ obtained by STFT can be written as

$$\hat{\boldsymbol{s}} \xrightarrow{\text{STFT}} \hat{\boldsymbol{S}}_{t,f} = \sum_{n=0}^{N-1} \hat{\boldsymbol{s}}\omega\left[n-t\right]\exp\left(-i\frac{2\pi n}{N}f\right) \tag{2.13}$$

$$\hat{\boldsymbol{v}} \xrightarrow{\text{STFT}} \hat{\boldsymbol{V}}_{t,f} = \sum_{n=0}^{N-1} \hat{\boldsymbol{v}}\omega\left[n-t\right]\exp\left(-i\frac{2\pi n}{N}f\right). \tag{2.14}$$

After having obtained the T-F representation of the enhanced speech and noise, the ideal ratio mask (IRM) and a modified signal approximation (SA) are calculated using

$$\text{IRM} = \sqrt{\frac{|\hat{\boldsymbol{S}}(t,f)|^2}{|\hat{\boldsymbol{S}}(t,f)|^2 + |\hat{\boldsymbol{V}}(t,f)|^2}}, \tag{2.15}$$

where $\hat{\boldsymbol{S}}(t,f)^2$ and $\hat{\boldsymbol{V}}(t,f)^2$ represent the generated speech energy and noise energy with a T-F unit, respectively. Then a signal approximation method is used to train a ratio mask estimator that minimizes the difference between the spectral magnitude of the clean speech and the estimated speech. The mask loss $\mathcal{L}_{\text{mask}}$ is defined as

$$\mathcal{L}_{\text{mask}} = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} \| \text{IRM} \odot \boldsymbol{X} - \boldsymbol{S} \|_2, \tag{2.16}$$

where $\boldsymbol{X}$ and $\boldsymbol{S}$ are noisy speech and clean speech magnitudes, respectively.

During the training phase, similar to ForkGAN, the goal is to minimize the difference between the estimated signal pair $(\hat{\boldsymbol{s}},\ \hat{\boldsymbol{v}})$ and the ground truth signal pair $(\boldsymbol{s}, \boldsymbol{v})$ by optimizing the encoder and decoder functions. We feed the noisy speech $\boldsymbol{x}$, the clean speech $\boldsymbol{s}$, and the additive noise signal $\boldsymbol{v}$ into the proposed framework. In adversarial learning, $\boldsymbol{s}$ and $\boldsymbol{v}$ are also used as ground truth for regression in the generator. As such, the generator loss $\mathcal{L}_{\mathbf{G}}$ is the weighted sum of the mask loss, L_1 regular loss and original adversarial loss, which can be written as

$$\mathcal{L}_G = \mathcal{L}_{\mathbf{G}}^{(s)} + \mathcal{L}_{\mathbf{G}}^{(v)} + \alpha \cdot \mathcal{L}_{\text{mask}} \tag{2.17}$$

where α denotes the coefficient that controls the contribution of the mask loss function. When $\alpha = 0$, the proposed model is similar to ForkGAN, but in the time domain and without noise reduction loss and margin loss. When α is large, the proposed model becomes similar to mask-learning.

Two separate discriminators are adopted in the proposed framework to distinguish between real and fake speech and noise, respectively. During the training process, we sample two pairs of the speech signal: *1)* the real pair of samples, which consists of a clean signal $\boldsymbol{s}$ and additive noise $\boldsymbol{v}$; *2)* the fake pair of samples, which consists of the enhanced clean speech $\hat{\boldsymbol{s}}$ and the predicted noise signal $\hat{\boldsymbol{v}}$. Both signals are conditioned on the noisy speech $\boldsymbol{x}$. Two separate discriminator loss terms are then computed using (2.2) to update the parameters of the generator.

2.5 S-ForkGAN Architecture

We extend time-domain ForkGAN to frequency-domain ForkGAN, referred to S-ForkGAN. Spectral processing has three major advantages: 1) speech enhancement can seamlessly interface with a post-ASR system, since state-of-the-art ASRs widely use acoustic features in the frequency domain; 2) the input dimensions of the raw time-domain noisy speech signals are typically much higher than for the spectral features; and 3) by performing speech enhancement in the frequency domain, one reinforces ASR robustness. The proposed S-ForkGAN architecture is adapted from ForkGAN, which is shown in Fig. 2.3. It takes a noisy speech signal $\boldsymbol{x}$ as input and extracts its LPS features using a Fast Fourier transform (FFT). The training procedures are same as ForkGAN.

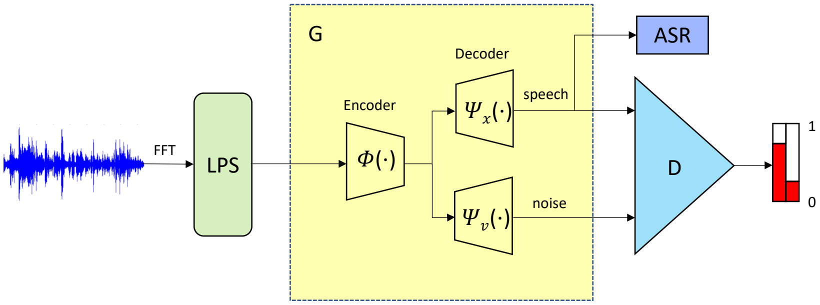


Figure 2.3: Proposed S-ForkGAN architecture which consists of forked GAN networks for simultaneous speech enhancement and noise identification.

2.6 Experiments Setup and Results

2.6.1 ForkGAN

The experimental setup comprises the selection of data sets, the training procedure of the proposed ForkGAN model, and the configuration of existing DNN-based and SEGAN systems that serve as a baseline for performance comparisons.

Data Set Selection. The data set that is used is derived from three sources. The TIMIT corpus [20] is used for clean speech references, and the noise is extracted from the NOISEX-92 corpus [112] and the NOISE-100 corpus [33]. The TIMIT corpus includes eight major dialects of American-English recordings from 630 speakers, where each speaker reads ten phonetically rich sentences. This corpus is partitioned into test and training subsets. The NOISEX-92 corpus includes 15 different noise sounds, including babble, factory, and white noise. The NOISE-100 includes 100 different noise sounds, e.g., animal sounds, water sounds, and bells. For the training set, a randomly-selected noise sound from NOISEX-92 or NOISE-100 was attached to every silenced-added segment with five different signal-to-noise ratios (SNRs): -5 dB, 0 dB, 5 dB, and 10 dB. The test set was generated by adding noise from the NOISEX-92 or the NOISE-100 corpus using the same settings as for the training set. To make the data set more realistic, another scenario is to add different noise types for each sentence. We designed a data set, which selects three different noise types from NOISEX-92 for each sentence, and refer to this data set as a multi-noise source data set. The clean speech from the TIMIT corpus is used as the target.

ForkGAN Setup. For the experiments, the ForkGAN model is trained for 86 epochs with the RMSprop algorithm [106], and the learning rate is set to 0.0002. The batch size is set to 32. During training, ForkGAN operates directly on raw audio. Every 500 ms, ForkGAN uses a 1-second sliding window with a 50-percent overlap to extract chunks of noisy speech waveforms of 16,384 samples each. A high-frequency pre-emphasis filter with filter coefficient 0.95 is applied to all input samples during the training and test stages. ForkGAN uses a shared encoder that is composed of 11 one-dimensional strided convolutional filter layers of width 31 and stride length 2. The number of filters per convolutional layer increases so that the depth increases as the width, i.e., the duration of the signal in time, gets shorter. The resulting dimensions d per layer ℓ, denoted as $d \times \ell$, where d corresponds to the number of samples and ℓ to the number of feature maps, are 16384×1, 8192×16, 4096×32, 2048×32, 1024×64, 512×64, 256×128, 128×128, 64×256, 32×256, 16×512, and 8×1024. In order to estimate the margin-based loss, a flattening operation is used to convert a 8×1024 vector to two 8192-dimensional vectors through a fully connected layer. The noise samples $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ from a prior 8×1024-dimensional normal distribution $\mathcal{N}(0, I)$ are concatenated with the two latent vectors ($\boldsymbol{c}_1$ and c_2) that were produced by the encoder. The network parameters of the decoder are symmetric to the encoder.

The discriminator utilizes a one-dimensional convolution similar to the generator's encoding stage and is adapted to behave as a classification network. The configuration of the discriminator is 16384×1, 8192×16, 4096×32, 2048×32, 1024×64. For the activation, a virtual batch-norm as presented in [93] was adopted before LeakyReLU nonlinearities with $\alpha = 0.3$. To reduce the number of parameters, the final layer is merged from 8×1024 to 8 using convolution.

Baseline Setup. The baseline systems used for performance comparison are the SEGAN system and the DNN-based system presented in [124]. The SEGAN set-up is very similar to ForkGAN, since the generator is an auto-encoder architecture with the same configuration as for ForkGAN. However, SEGAN uses only one decoder to generate clean speech. The DNN-based speech enhancement [124] applies log-spectral features that are spliced in time by taking a context size of seven frames, i.e., three preceding frames, the current frame and three succeeding frames. In the training stage, the regression DNN model, which uses a mean absolute error (MAE) loss function, is trained with samples from the TIMIT corpus. This corpus consists of pairs of noisy and clean speech represented by its log-spectral features. The full network topology consists of three hidden layers and 2048 hidden units. The network was trained for 10,000 iterations using the Adam optimizer with a mini-batch size of 500 and 20 % drop-out in the hidden layers.

Evaluation Metrics. Speech enhancement is commonly measured in terms of the *perceptual evaluation of speech quality* (PESQ) score [39] and the *short-time objective intelligibility* (STOI) score [99]. The PESQ score has a high correlation with subjective evaluation scores, and is mostly used as a compressive objective measure. The PESQ score is computed by comparing the enhanced speech with the clean reference speech, and it ranges from -0.5 to 4.5. The STOI score is highly relevant to human speech intelligibility and the score ranges from 0 to 1.

ForkGAN vs SEGAN vs DNN. The results of the experiments with the TIMIT corpus and different noise conditions ranging from -5 dB to 10 dB are shown in Table 2.1 and Table 2.2. It shows that ForkGAN outperforms both SEGAN and the DNN-based system for most SNR conditions. In comparison with the DNN-base system, ForkGAN significantly improves the average PESQ score from 2.493 to 2.788 and the average STOI score from 0.7634 to 0.8134 when using the TIMIT corpus with NOISEX-92. ForkGAN slightly outperforms SEGAN, with PESQ and STOI score improvements of 0.07 and 0.01, respectively. This suggests that the additional decoder for noise generation helps to purify the speech signal prediction. We also observe that for low SNR conditions, ForkGAN performs better than SEGAN and DNN, which suggests that ForkGAN is more robust for high noise conditions.

Effectiveness of Margin-Based Loss. The ForkGAN-M model incorporates the margin-based loss $\mathcal{L}_{\mathrm{M}}$ that maximizes the distance between noise and speech latent variables. Table 2.1 and Table 2.2 show that margin-based loss improves the speech enhancement performance slightly. For instance, relative to ForkGAN, the PESQ and STOI scores improve from 2.788 to 2.815 and from 0.8134 to 0.8137, respectively. The results indicate that ForkGAN-M outperforms all baseline models. This result shows that the extra margin-based loss effectively decouples the noisy speech to noise and clean speech.

Effectiveness of Noise Reduction Loss. ForkGAN-M-NR, which integrates time-domain noise subtraction loss and margin-based loss, achieves the best overall performance. Note that at a high SNR of 10 dB, the performance decreases slightly when compared with ForkGAN-M. This is reasonable because the noise reduction loss mainly optimizes the noise-reduced speech as in (2.8), whereas the distortion of enhanced speech depends on the quality of predicted noise. This causes the performance error to accumulate when the generated noise is inaccurate, especially for high SNR. It shows that using the noise reduction loss function improves performance mainly for low SNR conditions.

Model	-5 dB		0 dB		5 dB		10 dB		average	
Evaluation	PESQ	STOI	PESQ	STOI	PESQ	STOI	PESQ	STOI	PESQ	STOI
Noisy speech ($\boldsymbol{x}$)	1.51 ± 0.51	0.5863	1.84 ± 0.52	0.6721	2.20 ± 0.50	0.7524	2.55 ± 0.48	0.8180	2.025	0.7072
DNN	2.01 ± 0.36	0.6741	2.36 ± 0.33	0.7479	2.67 ± 0.29	0.7982	2.93 ± 0.26	0.8333	2.493	0.7634
SEGAN	2.37 ± 0.54	0.7511	2.56 ± 0.50	0.7860	2.91 ± 0.40	0.8342	3.01 ± 0.40	0.8525	2.713	0.8059
ForkGAN	2.46 ± 0.53	0.7575	2.66 ± 0.50	0.8019	2.88 ± 0.41	0.8340	3.15 ± 0.33	0.8603	2.788	0.8134
ForkGAN-M	2.43 ± 0.53	0.7565	2.73 ± 0.46	0.8006	2.92 ± 0.38	0.8352	**3.18 ± 0.34**	**0.8627**	2.815	0.8137
ForkGAN-M-NR	**2.50 ± 0.50**	**0.7593**	**2.77 ± 0.45**	**0.8083**	**2.98 ± 0.41**	**0.8395**	3.15 ± 0.34	0.8622	**2.850**	**0.8173**

Table 2.1: Single noise source mixture: performance of the three proposed ForkGAN models and existing DNN-based and SEGAN models for noisy speech ($\boldsymbol{x}$) from the TIMIT and NOISEX-92 data sets. The best values in each column are printed in boldface.

Model	-5 dB		0 dB		5 dB		10 dB		Average	
Evaluation	PESQ	STOI	PESQ	STOI	PESQ	STOI	PESQ	STOI	PESQ	STOI
Noisy speech ($\boldsymbol{x}$)	1.52 ± 0.45	0.6184	1.82 ± 0.43	0.7037	2.13 ± 0.40	0.7776	2.46 ± 0.38	0.8349	1.982	0.7336
DNN	1.99 ± 0.41	0.6832	2.33 ± 0.36	0.7481	2.61 ± 0.31	0.7964	2.87 ± 0.29	0.8289	2.450	0.7641
SEGAN	2.14 ± 0.50	0.7330	2.49 ± 0.46	0.7979	2.85 ± 0.39	0.8416	3.15 ± 0.30	0.8696	2.658	0.8105
ForkGAN	2.12 ± 0.51	0.7403	2.50 ± 0.44	0.7998	2.91 ± 0.35	0.8437	**3.20 ± 0.31**	0.8721	2.682	0.8140
ForkGAN-M	2.15 ± 0.48	0.7292	2.56 ± 0.43	0.8042	2.90 ± 0.36	0.8413	**3.20 ± 0.31**	0.8618	2.702	0.8116
ForkGAN-M-NR	**2.20 ± 0.50**	**0.7382**	**2.61 ± 0.42**	**0.8073**	**2.92 ± 0.36**	**0.8455**	3.19 ± 0.31	**0.8724**	**2.730**	**0.8159**

Table 2.2: Single noise source mixture: performance of the three proposed ForkGAN models and the baseline DNN-based and SEGAN models for noisy speech samples ($\boldsymbol{x}$) from the TIMIT and NOISE-100 data sets. The best values in each column are printed in boldface.

Model	-5 dB		0 dB		5 dB		10 dB		Average	
Evaluation	PESQ	STOI	PESQ	STOI	PESQ	STOI	PESQ	STOI	PESQ	STOI
Noisy speech ($\boldsymbol{x}$)	1.41 ± 0.34	0.5836	1.78 ± 0.33	0.6719	2.16 ± 0.30	0.7524	2.53 ± 0.28	0.8176	1.97	0.7064
DNN	2.04 ± 0.29	0.6732	2.42 ± 0.23	0.7452	2.69 ± 0.22	0.7917	2.90 ± 0.26	0.8352	2.51	0.7613
SEGAN	2.03 ± 0.33	0.7152	2.41 ± 0.31	0.7879	2.78 ± 0.27	0.8347	3.09 ± 0.24	0.8568	2.57	0.7886
ForkGAN-M-NR	**2.17 ± 0.30**	**0.7218**	**2.48 ± 0.30**	**0.7914**	**2.86 ± 0.26**	**0.8343**	**3.17 ± 0.23**	**0.8643**	**2.67**	**0.8030**

Table 2.3: Multiple noise sources: performance of the proposed ForkGAN models and existing DNN-based and SEGAN models for noisy speech ($\boldsymbol{x}$) from the TIMIT and NOISEX-92 data sets. In this scenario, each sentence is with three different noise type, which is more realistic. The best values in each column are printed in bold.

Multi-noise sources mixture. In a practical speech environment, the speech is often mixed with multiple noise types. We consider a multi-noise sources mixture environment to emulate more realistic scenario. In each dataset we randomly sample three different noise types and mix these with the speech signal. The results are presented in Table 2.3. It shows that ForkGAN outperforms both DNN and SEGAN in multiple noise conditions. ForkGAN's average PESQ score is 2.67, whereas the DNN-based system and SEGAN achieve 2.51 and 2.57, respectively.

Visualization. The network latent representation can be visualized by plotting the features of the noise decoder using t-SNE as in [111]. The 8192-dimensional noise features are projected down to 2 dimensions. Eight noise types are sampled, where each noise type contains about 400 samples. As shown in Fig. 2.4, ForkGAN makes use of latent variables and is capable of learning an interpretative noise representation in latent space. Furthermore, different types of noises are discriminated in large margin, indicating that learning noise representation is a relatively easy task when compared with with speech signal enhancement.

The differences in performance are illustrated by spectrograms of an utterance sample as shown in Fig. 2.5. The noisy speech mixes clean speech with additive pink noise at an SNR of 0 dB. We highlight several spots in Fig. 2.5, demonstrating that: DNN enhanced speech is distorted heavily, and high frequency noise is not properly suppressed; the SEGAN based model inhibits noises in a more aggressive way such that the low frequency speech information is cut sharply on the spectrogram; ForkGAN maintains most of the speech patterns, while suppressing noise smoothly.

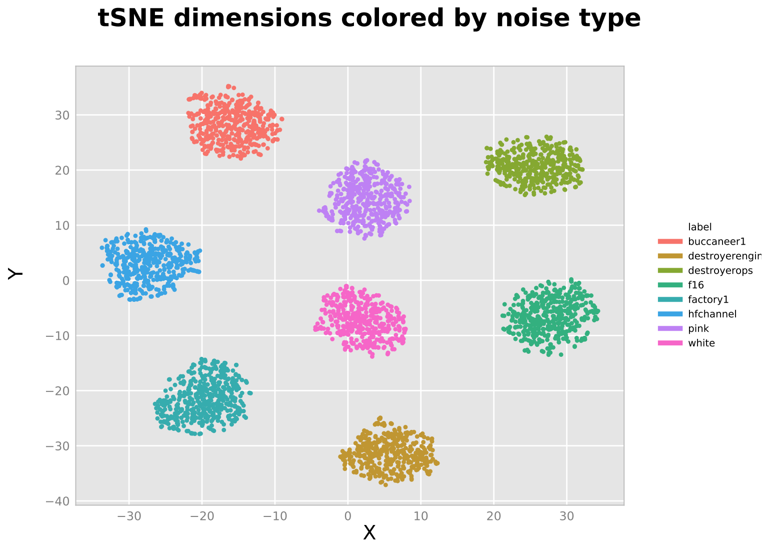


Figure 2.4: Examples of t-SNE visualizations of the noise latent representation c_n generated from the encoder when eight noise types from NOISEX-92 are used: pink, F16, destroyer engine, destroyer ops, white, buccaneer, and an HF channel.

ASR for Noisy Speech. In order to examine how the proposed ForkGAN improves ASR performance as a whole, an ASR is pre-trained for the speech recognition task with noisy TIMIT data set after which the speech enhancement is embedded prior to feeding it to the ASR model. The phone error rate (PER) is used as the evaluation metric.

Model	NOISEX-92					NOISE-100				
Evaluation	-5 dB	0 dB	5 dB	10 dB	average	-5 dB	0 dB	5 dB	10 dB	average
Noisy speech ($\boldsymbol{x}$)	84.0	76.8	65.5	49.8	68.98	79.2	72.0	61.4	48.5	65.28
DNN	57.8	48.0	38.6	32.1	44.13	55.4	46.1	37.9	31.4	42.7
SEGAN	48.6	44.9	35.2	34.4	40.78	53.6	43.5	36.1	29.1	40.58
ForkGAN	**45.7**	39.2	36.3	30.2	37.85	52.6	43.9	36.1	28.9	40.3
ForkGAN-M	46.7	38.7	34.7	**30.1**	37.55	52.7	43.4	34.7	29.0	39.95
ForkGAN-M-NR	47.1	**37.2**	**33.5**	30.4	**37.05**	**51.4**	**42.9**	**33.4**	**28.6**	**39.07**

Table 2.4: Phone error rate (PER), as a percentage, at the output of the ASR system. Note that baseline systems are DNN-based and SEGAN systems. (a) Noisy speech from TIMIT with NOISEX-92 (b) Noisy speech from TIMIT with NOISE-100. The best values in each column are boldfaced.

Pre-Trained ASR. A Deep Neural Network Hidden Markov Model (DNN-HMM) acoustic model was used to test the ASR performance of enhanced speech. A Gaussian Mixture Model-Hidden Markov Model(GMM-HMM) is first trained to obtain senones (tied triphone states) and the corresponding

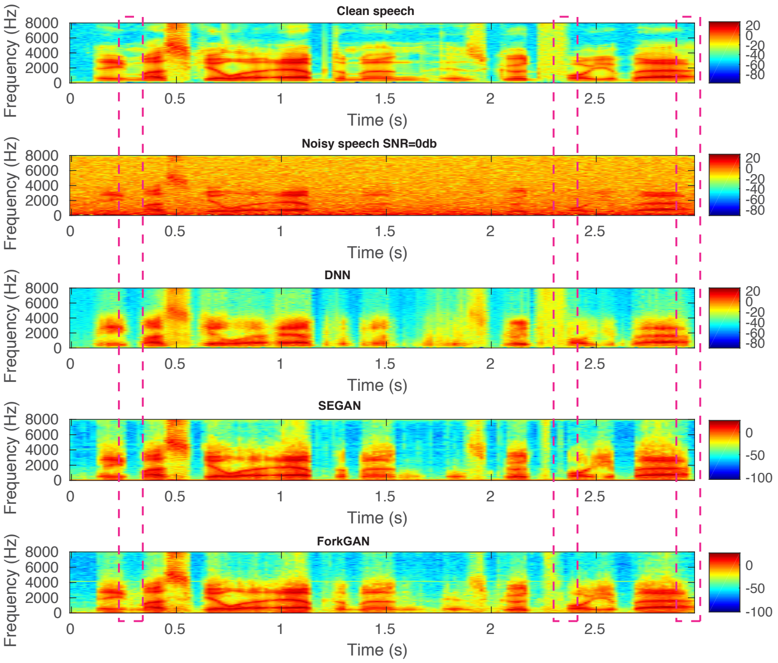


Figure 2.5: Single noise source mixture: Spectrograms of a sample input mixed with pink noise (SNR = 0 dB) and for the speech enhancement methods (DNN, SEGAN, ForkGAN-M-NR).

aligned frames for DNN training. The input feature vectors are used to train the GMM-HMM contain 13-dimensional Mel-frequency spectral coefficients (MFCCs) and their first and second derivatives. Context-dependent phones, tri-phones, are modeled by 3-state HMMs. The splices of 9 frames (4 on each side of the current frame) are projected down to 40-dimensional vectors by linear discriminant analysis (LDA), together with maximum likelihood linear transform (MLLT), and then used to train the GMM-HMM using maximum likelihood estimation.

The MFCC features stacked over an 11-frame window are used as the input layer of the DNN. The DNN itself has six hidden layers, and each layer contains $1,024$ nodes. Since TIMIT is a small corpus, the DNN acoustic model was first initialized with stacked restricted Boltzmann machines (RBMs) that were pre-trained in a greedy layer-wise fashion as in [30]. After pre-training, all weights and biases were discriminatory trained by optimizing the crossentropy between the target (corresponding to context-dependent HMM states) probability and actual output of soft-max output with the Back-Propagation (BP) algorithm, see [91].

As shown in Table 2.4 and Table 2.5, ForkGAN outperforms all baseline system (DNN and SEGAN). More specifically, for TIMIT with NOISEX-92, ForkGAN achieves absolute average gains of 7.1 %and 3.7 % relative to DNN and SEGAN, respectively, and for TIMIT with NOISE-100 it achieves gains of 3.6 % and 1.5 %, respectively.

At each SNR condition, the best performance was obtained by ForkGAN and its variants. The proposed use of two auxiliary loss terms are also effective in the context of ASR. ForkGAN-M-NR provides an additional improvement of the ASR performance relative to ForkGAN when using the TIMIT corpus of 0.8 % and 1.23 % for NOISEX-92 and NOISE-100, respectively.

Discussion. We compared the performance for two widely used noise data sets: NOISEX-92 and NOISE-100. As shown in Table 2.1 and Table 2.2, the proposed method works better on TIMIT with NOISEX-92 than with NOISE-100. For perceptive tests, the average PESQ score improvements

SNR	-5 dB	0 dB	5 dB	10 dB	average
Speech affected by noise ($\boldsymbol{x}$)	79.7	72.6	62.5	49.2	66.0
DNN	57.9	46.8	39.1	33.7	44.38
SEGAN	54.5	45.8	37.4	31.2	42.23
ForkGAN-M-NR	**53.2**	**44.6**	**37.4**	**29.4**	**41.15**

Table 2.5: Multiple noise sources mixture: The Phone Error Rate (PER), in percents, at the output of the ASR system.

relative to SEGAN are 0.07 and 0.137 for ForkGAN-M-NR with NOISEX-92 and NOISE-100, respectively. Similarly, the ASR experiments showed that the average PER improved by relative 9.1 % and 3.7 % with NOISEX-92 and NOISE-100, respectively (Table 2.4). As NOISE-100 includes more noise variations than NOISEX-92, this may indicate that our proposed method is sensitive to noise variations, but it still works. This is reasonable since with the increase of noise variation, the noise will become increasingly harder to estimate. ForkGAN-M-NR aims to use the additional noise information to improve the speech enhancement performance, the underestimated noise information would impact the performance of ForkGAN-M-NR. The multi-noise-source experiments, which may be closer to real-life situations, show that the proposed ForkGAN methods perform better than the SEGAN and DNN-based systems, and as such, are likely to be more robust for the arbitrary combinations of known noise types that were used in the training sets.

Our experiments use the same noise types in the training and the test set. Since ForkGAN captures additional noise information, the proposed method may encounter some generalization issues when the noise in the test set has never been seen during training. In other words, ForkGAN is more robust for handle known noise types in the test set. This is due to the fact that ForkGAN has two objectives that decouple noisy speech onto a noise signal and speech signal. In conditions where the test set contains unseen noise, inaccurate noise estimation would bias the generation of the clean speech. SEGAN or DNN systems, which do not take noise information into account in the loss function, will not be sensitive to new noise types.

2.6.2 M-ForkGAN

Data Sets. The data sets used for the experiments are the TIMIT corpus [20] and the NOISE-100 corpus [33]. The TIMIT corpus is used for clean speech references, and it includes eight major dialects of American-English recorded from 630 speakers, each reading ten phonetically-rich sentences, and partitioned into test and training subsets. The NOISE-100 corpus includes 100 different noise sounds, e.g., animal sounds, and the sound of water. For the training set, a randomly selected noise sound from the NOISE-100 corpus was attached to every silence-added segment with signal-to-noise ratios (SNRs): -3, 0, 3, 6, 9, 12 and 15 dB. In total, this yields 32,340 training samples. We selected 50 sentences from the TIMIT core test and mixed the noise from the NOISE-100 corpus with five SNR conditions (250 sentences in total). For the unseen scenario, we use five unseen SNR conditions at

-5, -2, 1, 4, and 7 dB. Note that both seen and unseen conditions were mixed with the same noise from the NOISE-100 corpus.

Baseline Setup. The proposed method is compared with the DNN-based speech enhancement [124], SEGAN [83], and SEGAN+ (`https://github.com/santi-pdp/segan_pytorch`).

DNN-based speech enhancement. Log-spectral features were applied for DNN-based speech enhancement spliced in time by taking a context size of seven frames. In the training stage, a regression DNN model using the mean absolute error (MAE) loss function is trained. The full network topology consists of three hidden layers and 2048 hidden units. The network was trained for 100 epochs using the Adam optimizer with a mini-batch size of 500 and a 20% drop-out in the hidden layers.

SEGAN and SEGAN+. The default parameter settings of the original SEGAN experiments are used, except for the batch size, which is set to 32. Both SEGAN and SEGAN+ take a raw 16,384-sample waveform as input. In SEGAN, **G** is composed of 22 1-D strided convolutional layers with filter-width 31 and stride 2. For SEGAN+, this is replaced by a generator comprising 10 1-D convolutional layers and stride 4. The virtual batch-norm (VBN) [93] that is used in SEGAN is replaced by a normal batch normalization in the discriminator.

Setup for the Proposed Method. The proposed model is trained for 100 epochs using an Adam optimizer [46] and a batch size of 32. The proposed approach operates directly on raw audio, which uses a 1-second sliding window with a 50-percent overlap to extract chunks of noisy speech waveforms of 16,384 samples each. A high-frequency pre-emphasis filter with a filter coefficient 0.95 is applied to all input samples during the training and test stages. The generator comprises one encoder and two decoders. Both the encoder and the two decoders consist of five 1-D convolutional layers as shown in Table 2.6. The speech decoder and the noise decoder have the same structure. Note that the decoder has the skip connections from the encoder part. Two separate fully-connected layers are used for generating speech and noise latent representations. For a short time Fourier transform (STFT) setting, we use a 20 ms Hann window, a 20 ms filter length and a 10 ms hop size. Thus, the input size of STFT is 16,384 and the output is 161×103. For the weight of $\mathcal{L}_{\text{mask}}$, we consider three settings, $\alpha \in \{0, 30, 50\}$, where $\alpha = 0$ means no $\mathcal{L}_{\text{mask}}$ for the training. Two discriminators are used to distinguish fake and real speech and noise, respectively. They both have the same model architecture, which is similar with the encoder in **G**. We use instance normalization (IN) [107] instead of batch normalization (BN) [35] in the discriminator as we found that IN is slightly better than BN. After convolutional layers, there are three fully connected layers (hidden layer size 256,128,1) with PReLU [29] for binary classification.

layer type		output size
input layer		1 × 16384
Encoder	conv-1-D	64 × 4096
	conv-1-D	128 × 1024
	conv-1-D	256 × 256
	conv-1-D	512 × 64
	conv-1-D	1024 × 16
Fully connected layer		8192
Fully connected layer		16384
Decoders	deconv-1-D	2048 × 16
	deconv-1-D	1024 × 64
	deconv-1-D	512 × 256
	deconv-1-D	256 × 1024
	deconv-1-D	128 × 4096
	deconv-1-D	1 × 16384
STFT conv-1-D		161 × 103

Table 2.6: Model Structures of the Proposed Method

Performance Results. Measurements were performed using the TIMIT corpus and NOISE-100 corpus to compare the proposed methods with $\alpha \in \{0, 30, 50\}$ and the baseline methods, i.e., the DNN-based method, SEGAN, and SEGAN+. The experimental results are detailed in Table 2.7. It is shown that the proposed method, with $\alpha = 30$, consistently outperforms the baseline methods for both seen and unseen conditions. The best baseline method is SEGAN+, and the DNN-based method outperforms SEGAN when using the PESQ metric, and SEGAN performs better than the DNN-based method when using the STOI metric.

Table 2.7 clearly shows the improvements obtained when applying the mask $\mathcal{L}_{\text{mask}}$ with $\alpha = 30$ relative to the situation where the mask is not used, i.e., for $\alpha = 0$. This shows that the additional mask-based loss helps purify speech signal prediction. For instance, the average PESQ and STOI scores were improved from 2.698 to 2.856 and from 0.9061 to 0.9353 on unseen SNR conditions, respectively. Furthermore, the proposed method outperforms all the baseline systems on both seen and unseen SNR conditions. When compared with SEGAN+, the proposed approach improves the average PESQ from 2.772 to 2.856, and the average STOI from 0.9132 to 0.9353 on

Metrics			w/o SE	DNN	SEGAN	SEGAN+	Proposed method		
							$\alpha = 0$	$\alpha = 30$	$\alpha = 50$
PESQ	seen	-3 dB	1.50 ± 0.33	2.27 ± 0.47	2.15 ± 0.48	2.52 ± 0.38	2.47 ± 0.36	$\mathbf{2.67 \pm 0.33}$	2.55 ± 0.33
		0 dB	1.69 ± 0.32	2.47 ± 0.41	2.37 ± 0.43	2.70 ± 0.38	2.66 ± 0.35	$\mathbf{2.83 \pm 0.34}$	2.73 ± 0.34
		3 dB	1.89 ± 0.31	2.64 ± 0.36	2.56 ± 0.40	2.86 ± 0.38	2.80 ± 0.37	$\mathbf{2.97 \pm 0.36}$	2.89 ± 0.35
		6 dB	2.11 ± 0.30	2.80 ± 0.31	2.75 ± 0.37	3.00 ± 0.38	2.94 ± 0.37	$\mathbf{3.09 \pm 0.40}$	3.01 ± 0.37
		9 dB	2.32 ± 0.29	2.94 ± 0.28	2.93 ± 0.33	3.12 ± 0.36	3.06 ± 0.38	$\mathbf{3.17 \pm 0.46}$	3.11 ± 0.38
		average	1.902	2.624	2.552	2.840	2.786	**2.946**	2.858
	unseen	-5 dB	1.40 ± 0.53	2.25 ± 0.58	2.05 ± 0.50	2.44 ± 0.42	2.34 ± 0.41	$\mathbf{2.54 \pm 0.42}$	2.45 ± 0.39
		-2 dB	1.62 ± 0.48	2.45 ± 0.50	2.26 ± 0.46	2.63 ± 0.40	2.55 ± 0.38	$\mathbf{2.71 \pm 0.41}$	2.64 ± 0.38
		1 dB	1.82 ± 0.48	2.63 ± 0.42	2.46 ± 0.42	2.79 ± 0.40	2.73 ± 0.37	$\mathbf{2.88 \pm 0.39}$	2.81 ± 0.37
		4 dB	2.02 ± 0.47	2.78 ± 0.36	2.64 ± 0.39	2.94 ± 0.38	2.87 ± 0.38	$\mathbf{3.03 \pm 0.37}$	2.95 ± 0.38
		7 dB	2.22 ± 0.46	2.91 ± 0.32	2.82 ± 0.35	3.06 ± 0.36	3.00 ± 0.38	$\mathbf{3.12 \pm 0.38}$	3.06 ± 0.38
		average	1.816	2.604	2.446	2.772	2.698	**2.856**	2.782
STOI	seen	-3 dB	0.7086	0.7576	0.8179	0.8791	0.8634	**0.9056**	0.8886
		0 dB	0.7608	0.7883	0.8577	0.9062	0.8956	**0.9287**	0.9145
		3 dB	0.8094	0.8132	0.8896	0.9261	0.9181	**0.9447**	0.9329
		6 dB	0.8535	0.8331	0.9143	0.9407	0.9351	**0.9549**	0.9464
		9 dB	0.8918	0.8482	0.9336	0.9519	0.947	**0.9577**	0.9559
		average	0.8048	0.8081	0.8826	0.9208	0.9118	**0.9383**	0.9277
	unseen	-5 dB	0.6653	0.7504	0.8037	0.8646	0.8492	**0.8971**	0.8757
		-2 dB	0.7213	0.7845	0.8468	0.8967	0.8872	**0.9235**	0.9067
		1 dB	0.7751	0.8103	0.8809	0.9198	0.9146	**0.9418**	0.9285
		4 dB	0.8245	0.8301	0.9082	0.9367	0.9330	**0.9534**	0.9440
		7 dB	0.8677	0.8450	0.9290	0.9484	0.9463	**0.9605**	0.9546
		average	0.7708	0.8041	0.8737	0.9132	0.9061	**0.9353**	0.9219

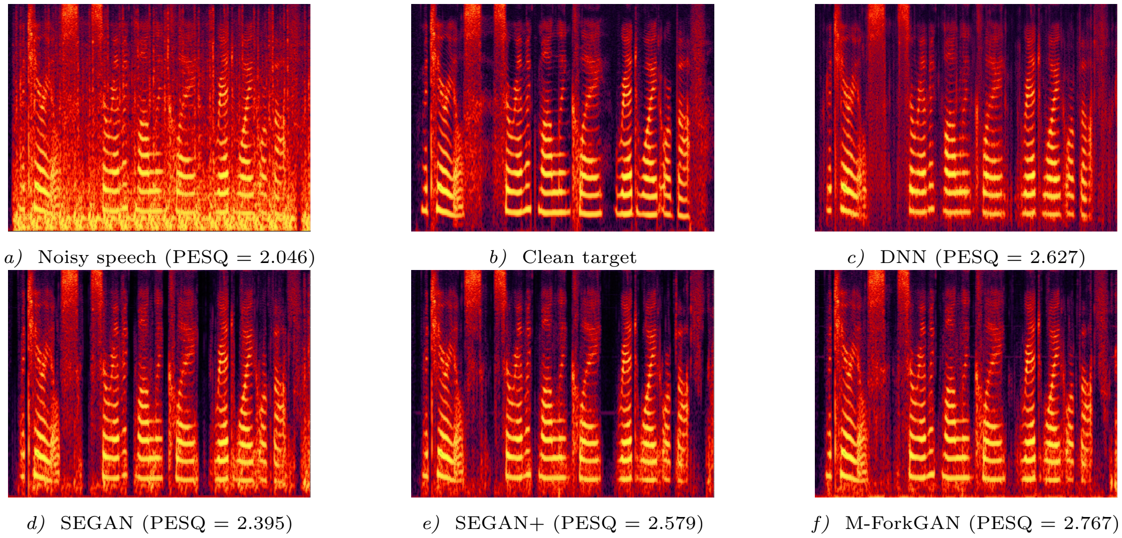
Table 2.7: Performance of three baseline models and the proposed model. The best values in each column are printed in boldface.

unseen SNR conditions. We also notice that if we use a large value for α, corresponding to a strong contribution of $\mathcal{L}_{\text{mask}}$, the performance degrades slightly, because mask-based learning introduces inaccurate information during training due to inaccuracies in the mask estimator. Thus, the loss of mask-based learning and time-domain feature mapping need to be calibrated. Fig 2.6 illustrates the effectiveness of the proposed approach by using spectrum.

2.6.3 S-ForkGAN

The performance of the proposed S-ForkGAN method is evaluated using extensive simulations.

Data Setup. The data set for the experiments is generated from two sources: the DARPA TIMIT corpus [20] is used for clean speech references, whereas the noise is extracted from the NOISEX-92 corpus [112]. The TIMIT corpus includes eight major American-English dialects recorded from 630 speakers, each reading ten phonetically rich sentences, and this corpus is partitioned into test and training subsets. The training set includes 4620 sentences; 192 sentences are selected for the testing set. The NOISEX-92 corpus includes 15 different noise types, ranging from machinery noise to machine gun noise. For the training set, a randomly-selected noise sound from NOISEX-92 is added



a) Noisy speech (PESQ = 2.046)

b) Clean target

c) DNN (PESQ = 2.627)

d) SEGAN (PESQ = 2.395)

e) SEGAN+ (PESQ = 2.579)

f) M-ForkGAN (PESQ = 2.767)

Figure 2.6: Spectrograms of a sample input mixed with N21 noise, where the SNR is equal to 1 dB.

to every silenced-added segment with a signal-to-noise ratio (SNR) of -5 dB, 0 dB, 5 dB and 10 dB. The test set was generated by adding noise from the NOISEX-92 corpus, using the same settings as the training set.

S-ForkGAN Setup. The proposed technique and architecture can be summarized as follows: the model is trained for 20 epochs with the RMSprop [106] method. It operates directly on spectral domain features, LPS, instead of on raw audio, and it aims to learn a mapping from the LPS feature input to the LPS feature output. The input and target LPS features are normalized by using zero mean and unit variance, respectively. The input feature contains a context window of 11 frames (±5), thus it is a 2827-to-257 mapping relation. Further experiments with ForkGAN use exactly the same settings. In S-ForkGAN, the shared encoder consists of 11 one-dimensional strided convolutional layers of filter width-31 and stride length 2. The number of filters per convolutional layer increases so that the depth increases as the width gets shorter. The resulting dimensions per layer in terms of the number of samples times the number of feature maps is 2827×1, 1414×16, 707×32, 354×32, 177×64, 89×64, 45×128, 23×128, 12×256, 6×256, 3×512 and 2×1024. A flattening operation is used for converting a 2×1024 vector to two length-2048 vectors via fully-connected layers. After that two encoded latent variables are obtained, which are used for speech and noise respectively. Also, the margin-based loss is calculated by these two vectors. The two encoded latent vectors are concatenated with two noise samples, which are from an a prior 2×1024-

dimensional normal distribution $\mathcal{N}(0, I)$. The concatenated vectors are the input of each decoder. The network parameters of the decoder are symmetric to the encoder. The discriminator also utilizes a one-dimensional convolution similar to the generator's encoding stage and is adapted to behave as a classification network.

Baseline Setup. Several GAN-based method with different enhancement networks are used as baseline systems, e.g., a DNN and long short-term memory (LSTM). Note that GAN-DNN and GAN-LSTM were originally used for speech de-reverberation; they are adjusted here for speech enhancement. The setup is similar to [117]. If the generator is an auto-encoder, the suffix AE is used.

GAN-DNN. The feed-forward DNN includes four hidden layers, each of which contains 1024 ReLU neurons. The input feature consists of a stacked 11-frame LPS feature. The mode is trained for 20 epochs using the learning rate 0.001 with a mini-batch size of length-8. Batch normalization is performed for this model.

GAN-LSTM. Instead of using a plain-vanilla LSTM, an LSTM with recurrent projection layer (LSTMP) [92] was adopted here. The LSTM includes four LSTMP layers followed by a linear output layer. Each LSTMP layer has 760 memory cells and 257 projection units and the input to the LSTM is a single acoustic frame with 257-dimensional LPS features. The learning rate was set to $3.0 \cdot 10^{-4}$ and the model was trained with eight full-length utterances parallel processing.

GAN-AE. The setup for GAN-AE is similar to S-ForkGAN. The only difference is that the generator is an auto-encoder architecture with the same configuration as the proposed method, using one decoder to generate clean speech.

ASR Setup. A Deep Neural Network-Hidden Markov Model (DNN-HMM) acoustic model is developed to evaluate the enhanced LPS features. A Gaussian Mixture Model-Hidden Markov Model(GMM-HMM) is first trained to obtain senones (tied tri-phone states) and the corresponding aligned frames for DNN training. The input feature vectors that are used to train the GMM-HMM contain 257-dimensional LPS and their first and second derivatives. The splices of nine frames (four on each side of the current frame) are projected down to 40-dimensional vectors by linear discriminant analysis (LDA), together with maximum likelihood linear transform (MLLT), and then used to train the GMM-HMM using maximum likelihood estimation.

The LPS features take a context size of 11 frames (±5), as the input of the DNN. The DNN topology consists of six hidden layers, and each layer contains $1,024$ nodes. Since TIMIT is a small corpus, the DNN acoustic model was first initialized with stacked restricted Boltzmann machines (RBMs) that were pre-trained in a greedy layered fashion [30]. After pre-training, all weights and biases were discriminator-trained by optimizing the cross-entropy between the target probability, corresponding to context-dependent HMM states, and the actual soft-max output with the Back-Propagation (BP) algorithm [91]. The weights are refined using sequence-discriminative training, state-level minimum Bayes risk (sMBR).

Performance Results. Using the experimental set-up presented in the previous section, the acoustic model was trained using clean data, and it was determined that the phone error rate (PER) on the TIMIT test set equals 18.0 %. The PER values were determined for several existing GAN-based speech enhancement approaches. It can be observed from Table 2.8 that all methods reduce the noise and improve the ASR performance. It is shown that GAN-LSTM achieves better results than GAN-DNN for all SNR values. For example, a GAN-LSTM reduces the PER from 32.6 % to 29.4 %. This indicates that LSTM's ability to model long-term contextual information is essential for speech enhancement.

SNR	-5 dB	0 dB	5 dB	10 dB
LPS w/o SE	87.2	81.3	70.1	56.0
GAN-DNN (LPS)	54.6	48.0	39.4	32.6
GAN-LSTM (LPS)	51.7	42.5	34.8	29.4
GAN-AE (raw audio)	48.6	44.9	35.2	34.4
GAN-AE (LPS)	45.7	38.3	34.1	32.4
ForkGAN	47.1	37.2	33.5	30.4
S-ForkGAN	45.1	37.8	30.5	26.8

Table 2.8: Phone error rate (as a percentage) for S-ForkGAN and prior methods; the best results for each SNR value are bold-faced.

The measurements show that GAN-AE with LPS inputs can further improve the performance, especially for high SNR. In contrast to GAN-LSTM, the PER drops from 51.7 % to 45.7 % and from 42.5 % to 38.3 % for an SNR of -5 dB and 0 dB, respectively. This means that the convolution layers in the auto-encoder can also provide additional useful information for speech enhancement. The proposed S-ForkGAN method with LPS achieved the best performance, which shows the effectiveness of the additional decoder and two auxiliary loss functions.

Given that S-ForkGAN and GAN-AE are auto-encoder-based methods, the performance is determined for different input features. The raw audio input is set to be the same as for the original SEGAN [83], where each chunk of waveform was extracted with a sliding window of approximately one second of speech (16,384 samples) every 500 ms. A high-frequency pre-emphasis filter coefficient of 0.95 was applied to all input samples during the training and test stages. From Table 2.8, one can see that both GAN-AE and S-ForkGAN with LPS features outperform systems with raw audio as input. The results show that directly operating on LPS is more helpful for the ASR tasks. Note that S-ForkGAN outperforms GAN-AE with respect to these two features.

To visualize the performance of the GAN-based methods, a single sentence was selected and mixed with destroyer noise at 0 dB. The spectrograms for each method are shown in Fig. 2.7. The S-ForkGAN and GAN-AE methods are clearly better than the GAN-DNN and GAN-LSTM methods.

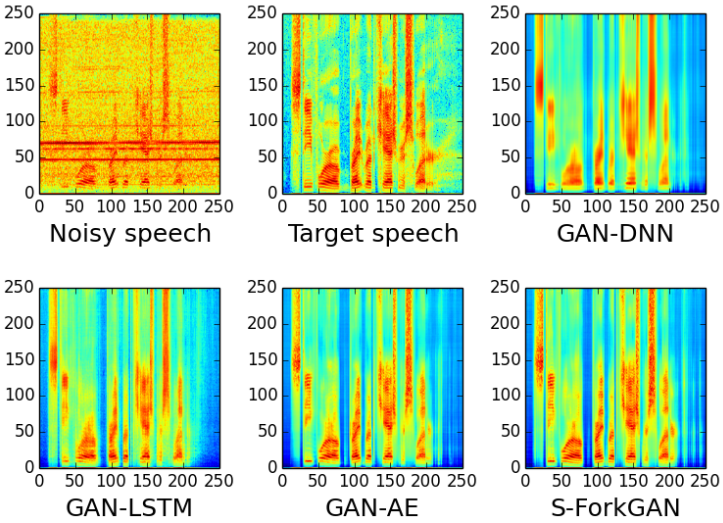


Figure 2.7: Spectrograms for GAN-DNN, GAN-LSTM, GAN-AE and the proposed S-ForkGAN method for a sample input mixture with destroyer noise and an SNR of 0 dB.

2.7 Summary

In this chapter, we have presented a novel GAN based approach for speech enhancement, named ForkGAN. The experiment results show that our proposed method outperforms the state-of-the-art speech enhancement methods in terms of perceptual evaluation of speech quality (PESQ) and Short-Time Objective Intelligibility (STOI) scores, and it also improve the speech recognition performance. Two variants of ForkGAN are further proposed, M-ForkGAN and S-ForkGAN, the effectiveness of them is also verified by a series of comprehensive experiments.

Chapter 3

Speech Enhancement Using Multi-Stage Self-Attentive Temporal Convolutional Networks

This chapter presents the Multi-Stage Self-Attentive Temporal Convolutional Networks (MS-SA-TCN) based speech enhancement approaches. The work presented in this chapter is to appear in the IEEE/ACM Transactions on Audio Speech and Language Processing [62].

3.1 Introduction

Recently, multi-stage learning has been successfully applied for a wide variety of tasks, including human pose estimation [76], action segmentation [16], speech enhancement [28, 58, 57] and speech separation [15]. A multi-stage architecture consists of stages that sequentially use the same model or a combination of different models, and each model operates directly on the output of the previous stage. The effect of such an arrangement is that the model used in a given stage takes the predictions from prior stages as input and incrementally refines these predictions.

Multi-stage learning systems that perform the same task in each stage typically use the same supervision principles in each intermediate stage [76, 16, 57]. In [16], multiple stacked TCN networks are proposed for action segmentation. In [57], a multi-stage network with dynamic attention was introduced, where the intermediate output in each stage is corrected with a memory mechanism. To reduce the model parameters, each stage uses a shared network. It is shown that this multi-stage approach typically performs better than systems with a larger and deeper network.

Multi-stage learning systems where each stage performs a different task are considered in [28, 58, 15]. Here, each stage has a different task and a different target. The performance can be improved by aggregating different stages if the nature of each stage is complementary. For instance, a two-stage speech enhancement approach is presented in [28], where the first stage uses a model to predict a binary mask to remove frequency bins that are dominated by severe noise, and where the second stage performs in-painting of the masked spectrogram from the first stage to recover the speech spectrogram that was removed in the first stage. In [58], a two-stage algorithm is proposed to optimize the magnitude and phase separately. The magnitude is optimized in the first stage and the enhanced magnitude and phase are then further refined jointly.

This chapter details a novel multi-stage speech enhancement system, where each stage comprises a self-attention (SA) block [113] followed by stacks of dilated temporal convolutional network (TCN) blocks. The system is referred to as a multi-stage SA-TCN speech enhancement system. Each stage generates a prediction in the form of a soft mask that is refined in each subsequent stage. Each self-attention block produces a dynamic representation for different noise environments and their relevance across frequency bins, as such enhancing the features, and the stacks of TCN blocks perform sequential refinement processing. A fusion block is inserted at the input of later stages to re-inject original speech information to mitigate possible speech information loss in earlier stages.

This chapter is organized as follows. Section 3.2 details the proposed multi-stage SA-TCN speech enhancement system and the underlying SA, TCN, and fusion blocks. Section 3.3 details the comprehensive experiments using the LibriSpeech [78] and VCTK [109] corpus. Section 3.4 first presents the experiments that were performed to fine-tune the multi-stage SA-TCN system's hyperparameters, to determine the optimum number of stages, and to quantify the impact of the SA block and the fusion block on the performance. The use of the proposed multi-stage SA-TCN system as a front-end for automatic speech recognition (ASR) systems is investigated as well. Extensive experiments with the LibriSpeech [78] and VCTK [109] corpus show that multi-stage SA-TCN systems achieve significantly better speech enhancement and speech recognition scores than other state-of-the-art speech enhancement systems. Section 3.5 concludes the paper and discusses further research directions.

3.2 Multi-Stage SA-TCN Systems

The proposed multi-stage SA-TCN speech enhancement system consists of K stages. Fig. 3.1 illustrates a 4-stage SA-TCN system. Each stage comprises a self-attention (SA) block followed by R stacks of L TCN blocks. For K-stage SA-TCN systems where $K \geq 3$, a feature fusion block is inserted prior to each stage k, where $3 \leq k \leq K$.

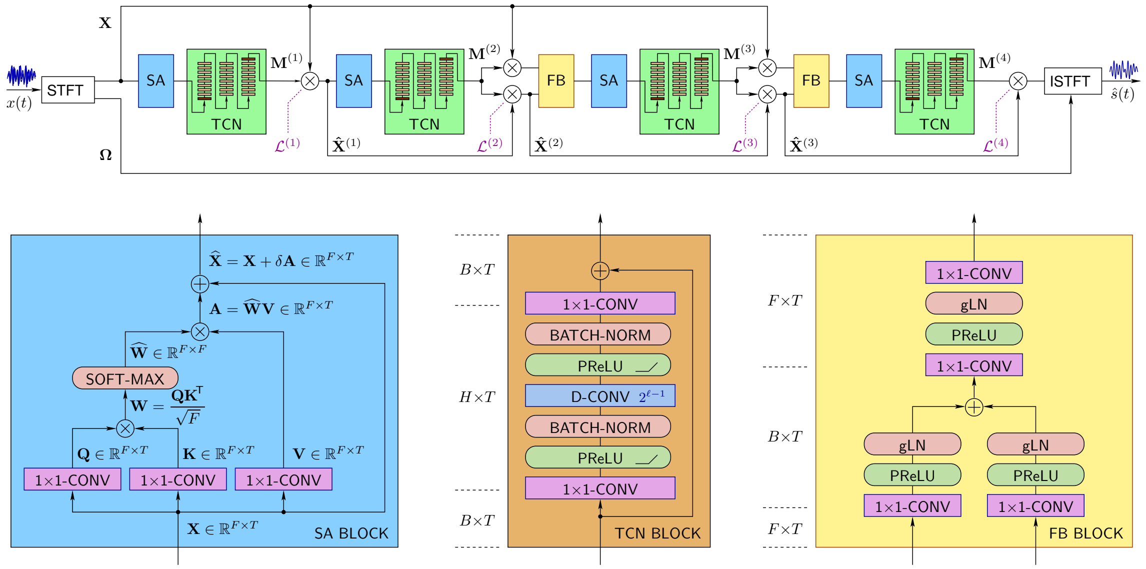


Figure 3.1: Block diagram of a multi-stage SA-TCN speech enhancement system with $K = 4$ stages, where each stage consists of a self-attention (SA) block, followed by $R = 3$ stacks of $L = 8$ TCN blocks, where the dilation factor of the ℓ-th TCN block in the stack equals $\Delta_\ell = 2^{\ell-1}$, i.e., the dilation doubles for each next TCN block in the stack. A fusion block is used as of Stage 3 to re-inject the original STFT magnitude. The figure also shows the detailed structure of the self-attention module, with frequency and time dimensions F and T, a single TCN block with hyperparameters B and H and P, and the proposed fusion block.

Each of the blocks have specific purposes that are particularly suited for speech enhancement. The self-attention mechanism aggregates context information across channels, which is particularly helpful in obtaining a dynamic representation when the noise is non-stationary, and this is the case for many speech enhancement scenarios.

The TCN consists of R stacks of L non-causal TCN blocks, where the dilation factor of the ℓ-th TCN block in the stack is given by $\Delta_\ell = 2^{\ell-1}$. As such, each stack has a large receptive field, which makes it particularly suited for temporal sequence modeling. Each TCN block has a skip connection between the input and output to reduce the loss of low-level details and to provide hooks for optimization.

The multi-stage architecture iteratively refines the initial predictions. It should be noted that the prediction of a previous stage may include some errors. For instance, the frequency bins dominated by speech may be masked and the resulting magnitude spectrogram may have lost some of the speech information. A fusion block is inserted prior to each stage k, where $3 \leq k \leq K$, that combines the predicted magnitude $\hat{\mathbf{X}}^{(k-1)}$ at the output of stage $k-1$ and the original magnitude $\mathbf{X}$ as input, in order to re-inject the original speech information.

The first stage consists of a self-attention (SA) block that takes $\mathbf{X}$ as input and that uses three 1×1-convolutions to form the query $\mathbf{Q}$ and the key-value pair $(\mathbf{K}, \mathbf{V})$, where $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{F \times T}$. In order to compute the attention component $\mathbf{A}$, we first compute the weight $\mathbf{W}$, given by

$$\mathbf{W} = \frac{\mathbf{Q}\mathbf{K}^{\mathsf{T}}}{\sqrt{F}}, \tag{3.1}$$

and then use the soft-max function $\sigma(\cdot)$ to obtain $\widehat{\mathbf{W}} = \{\widehat{W}_{i,j}\} = \sigma\left(\mathbf{W}\right)$, i.e.,

$$\widehat{W}_{i,j} = \frac{\exp\left(W_{i,j}\right)}{w_j}, \text{ where } \; w_j = \sum_{i=1}^{F} \exp\left(W_{i,j}\right). \tag{3.2}$$

The attention component $\mathbf{A} \in \mathbb{R}^{F \times T}$ is now determined using

$$\mathbf{A} = \widehat{\mathbf{W}}\mathbf{V}, \tag{3.3}$$

The SA block outputs $\widehat{\mathbf{X}} = \mathbf{X} + \delta\mathbf{A}$, where δ is a scalar with initial value zero that is used to allow the network to first rely on the cues in the local channels $\mathbf{X}$ and then gradually assign more weight to the non-local channels using back-propagation to reach its optimal value.

The output $\widehat{\mathbf{X}} \in \mathbb{R}^{F \times T}$ is fed into a TCN with input feature dimension B and network feature map dimension H by using a bottleneck layer to reduce the number of channels from F to B. The TCN consists of R identical stacks of L TCN blocks. Each TCN block comprises an 1×1 convolution at its input to match the input feature dimension B to the TCN block's internal feature map dimension H, a dilated depth-wise convolution (D-conv) layer with kernel size P and dilation factor $\Delta_\ell = 2^{\ell-1}$, where ℓ denotes the order of the TCN block in the stack of L TCN blocks, and a 1×1 convolution layer to reduce the number of channels at the output from H to B. This output is then recombined with the input using a skip connection to avoid losing low-level details.

A parametric rectified linear unit (PReLU) activation layer [29] and a batch normalization layer [35] are inserted prior to and after the depth-wise convolution layer to accelerate training and improve performance. A sigmoid function is applied at the output of the last TCN block of the last stack to obtain a [0-1] mask $\mathbf{M}^{(1)}$ that minimizes the mean absolute error loss

$$\mathcal{L}^{(1)} = \|\mathbf{M}^{(1)} \odot \mathbf{X} - \mathbf{S}\|, \tag{3.4}$$

where the operator $\odot$ denotes the Hadamard product and $\mathbf{S}$ denotes the STFT magnitude of the clean speech signal $s(t)$.

The stack of L TCN blocks with kernel P and dilation factor $\Delta_\ell = 2^{\ell-1}$ create a receptive field of size $R^{(P,L)}$, given by

$$R^{(P,L)} = 1 + \sum_{\ell=1}^{L} (P-1) \cdot 2^{\ell-1}. \tag{3.5}$$

As such, a stack of L TCN blocks creates a large temporal receptive field with fewer parameters than other models.

This chapter considers multi-stage SA-TCN systems with kernel size $P = 3$. An illustration of the receptive field for a stack of $L = 5$ TCN blocks with kernel size $P = 3$ is shown in Fig. 3.2. The multi-stage SA-TCN system's hyperparameters (B, H, R, L) will be optimized using experiments.

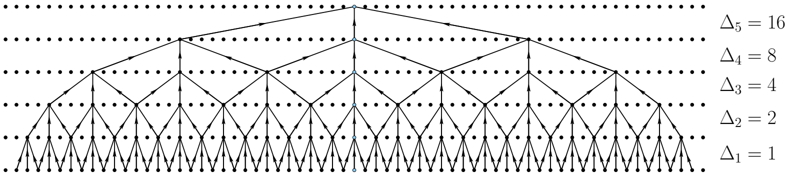


Figure 3.2: Example of the connections formed by a non-causal D-convolution for a stack of $L = 5$ TCN blocks, each with kernel $P = 3$ and dilation $\Delta_\ell = 2^{\ell-1}$, where $1 \leq \ell \leq L$.

As indicated, the same SA-TCN structure is used for subsequent stages, and an additional element, a fusion block, is inserted prior to each stage if there are three or more stages.

For notational convenience, let $\mathbf{\Psi}_k^{(R,L)}(\cdot)$ denote the mapping performed by the R stacks of L TCN blocks in stage k, and let $\mathbf{\Upsilon}_k(\cdot)$ denote the self-attention operation at stage k. It follows that $\mathbf{M}^{(1)}$ can now be expressed as

$$\mathbf{M}^{(1)} = S(\mathbf{\Psi}_1^{(R,L)}(\mathbf{\Upsilon}_1(\mathbf{X}))), \tag{3.6}$$

where $S(\cdot)$ denotes the sigmoid function. As such, $\mathbf{M}^{(1)}$ is the predicted mask at the output of the first stage. The enhanced speech STFT magnitude $\hat{\mathbf{X}}^{(1)}$ at the output of stage 1 is given by $\hat{\mathbf{X}}^{(1)} = \mathbf{M}^{(1)} \odot \mathbf{X}$.

In a similar fashion, the predicted mask $\mathbf{M}^{(2)}$ at the output of the second stage can be obtained by evaluating

$$\mathbf{M}^{(2)} = S(\mathbf{\Psi}_2^{(R,L)}(\mathbf{\Upsilon}_2(\hat{\mathbf{X}}^{(1)}))) \tag{3.7}$$

and the estimated STFT magnitude $\hat{\mathbf{X}}^{(2)} = \mathbf{M}^{(2)} \odot \hat{\mathbf{X}}^{(1)}$.

A multi-stage SA-TCN speech enhancement system with three or more stages ($K \geq 3$) is constructed by inserting a fusion block that performs operation $\mathbf{\Phi}(\cdot)$ prior to each stage k, where $3 \leq k \leq K$, taking the masked STFT magnitude $\hat{\mathbf{X}}^{(k-1)}$ and STFT magnitude $\mathbf{X}$ as inputs. Each input is passed through a 1×1-convolution and a PReLU operation, after which a global layer normalization (gLN) is performed [69]. The operation $gLN(\mathbf{Y})$ is given by

$$gLN(\mathbf{Y}) = \frac{\mathbf{Y} - E\left[\mathbf{Y}\right]}{\sqrt{\mathrm{var}(\mathbf{Y}) + \epsilon}} \odot \boldsymbol{\gamma} + \boldsymbol{\beta}, \tag{3.8}$$

where $\mathbf{Y} \in \mathbb{R}^{F \times T}$ is the input feature with mean $E\left[\mathbf{Y}\right]$ and variance $\mathrm{var}(\mathbf{Y})$, $\boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^{F \times 1}$ are trainable parameters, and ϵ is a small constant for numerical stability.

The outputs of the two gLN are added, and the result is again sent through a 1×1-convolution, a PReLU, another gLN, another 1×1-convolution and another PReLU. The output, denoted as $\breve{\mathbf{X}}^{(k-1)}$, is given by

$$\breve{\mathbf{X}}_{k-1} = \mathbf{\Phi}_k\left(\mathbf{M}^{(k-1)} \odot \mathbf{X}, \hat{\mathbf{X}}^{(k-1)}\right). \tag{3.9}$$

The output $\breve{\mathbf{X}}_{k-1}$ is then used as an input to the next stage, and the expression for the mask $\mathbf{M}^{(k)}$ at the output of the k-th stage is now given by

$$\mathbf{M}^{(k)} = S\big(\mathbf{\Psi_k}^{(R,L)}\big(\mathbf{\Upsilon_k}(\breve{\mathbf{X}}_k)\big)\big). \tag{3.10}$$

The enhanced magnitude $\hat{\mathbf{X}}^{(k)}$ at stage k is given by

$$\hat{\mathbf{X}}^{(k)} = \mathbf{M}^{(k)} \odot \hat{\mathbf{X}}^{(k-1)}. \tag{3.11}$$

Each next stage k, where $k > 1$, computes mask $\mathbf{M}^{(k)}$ that minimizes the mask-based signal approximation mean absolute error loss $\mathcal{L}^{(k)}$ using

$$\mathcal{L}^{(k)} = \big\|\mathbf{M}^{(k)} \odot \hat{\mathbf{X}}^{(k-1)} - \mathbf{S}\big\|, \tag{3.12}$$

where $\hat{\mathbf{X}}^{(k-1)}$ denotes the estimated STFT magnitude at stage $k-1$.

At the output of the last stage of the multi-stage SA-TCN system, the time-domain waveform $\hat{\boldsymbol{s}}$ is computed using the processed STFT magnitude $\hat{\mathbf{X}}^{(k)}$ and the original STFT phase Ω by applying the inverse STFT, in short ISTFT, denoted as

$$\hat{\boldsymbol{s}} = \mathrm{ISTFT}(\hat{\mathbf{X}}^{(K)}, \boldsymbol{\Omega}). \tag{3.13}$$

The proposed multi-stage SA-TCN system provides a mean absolute error loss $\mathcal{L}^{(k)}$ at the output of each stage. Since each stage provides an equal contribution during the training process, we use the accumulated mask-based signal approximation training objective function

$$\mathcal{L} = \sum_{k=1}^{K} \mathcal{L}^{(k)}. \tag{3.14}$$

The use of the mean absolute error loss is motivated by recent observations that it achieves better objective quality scores when using spectral mapping techniques [79, 80].

3.3 Experimental Setup

In the following, the data set, model set up and the evaluation metrics are detailed.

3.3.1 Data Set

To verify the effectiveness of the proposed multi-stage SA-TCN system, we conduct experiments using the LibriSpeech and VCTK data sets. The detailed set-up for each data set is detailed below.

LibriSpeech is an open-source corpus that contains 960 hours of speech derived from audio books in the LibriVox project. The sampling frequency is 16 kHz. The clean source is trained using 100 hours of speech data from the “train-clean” data set. The validation set uses 800 sentences from the “dev-clean” data set, and the test set uses 500 sentences from the “test-clean” data set. The training set uses 10,000 randomly selected noise sample sequences from the DNS Challenge [87]. The training clean speech has been cut to 75,206 4-second segments. The training and validation sets distort the clean segments with a randomly-selected noise sound from the DNS Challenge noise set with an SNR in the set $\{-5, -4, \cdots, 9, 10\}$ (in dB), The test set uses three distinct noise types: "babble noise" from the NOISEX-92 corpus [112], and "office noise" and "kitchen noise" from the DEMAND noise corpus [105]. The first channel signal of the corpus is used for data generation. Each clean utterance is distorted by a randomly selected noise type at a randomly selected SNR from the set $\{-5, 0, 5, 10, 15\}$ (in dB).

The **VCTK** database used here is derived from the Valentini-Botinhao corpus [109]. Each speaker fragment contains about 10 different sentences. The training set uses 28 speakers, and the test set uses two speakers. The training set used here uses 40 noise conditions: eight noise types and two artificial noise types from the Demand database [105]) are used at a randomly selected SNR from the set $\{0, 5, 10, 5\}$ (in dB). The test set uses 20 noise conditions: five noise types from the Demand database at a randomly selected SNR from the set $\{2.5, 7.5, 12.5, 17.5\}$ (in dB). There are about 20 different sentences in each condition for each test speaker. The test set conditions are different from the training set, as the test set uses different speakers and noise conditions.

3.3.2 Model Setup

The baseline systems used for performance comparison are a CRN system [100], a complex-CNN system that is based on concepts proposed in [119] and that was adapted for speech enhancement, and a multi-stage system DARCN [57]. The setup of the baseline systems and the proposed multi-stage SA-TCN systems are detailed below.

CRN: The CRN-based approach takes the magnitude as input. Instead of directly mapping the noisy magnitude to the clean magnitude, we adapted the CRN to predict the ratio mask and as such improve its performance. The CRN-based method consists of five 2D convolution layers with filters of size 3×2 each and [16, 32, 64, 128, 256] output channels, respectively. This output is post-processed by two LSTM layers with 1024 nodes each, and five 2D deconvolution layers with filter size 3×2 each and output channels [128, 64, 32, 16, 1], respectively.

Complex-CNN. The complex-CNN performs a complex spectral mapping [17, 101], where the real and imaginary spectrograms of the noisy speech signal are treated as two different input channels. An STFT is used with a 20 ms Hanning window, a 20 ms filter length and a 10 ms hop size. The architecture uses eight convolutional layers, one LSTM layer and two fully-connected layers, each with ReLU activations except for the last layer, which has a sigmoid activation. The parameters used here are similar to the ones used in [119], but now both the input and the output have two channels with real and imaginary components, respectively. The prediction serves as a complex mask, consisting of a real and imaginary mask. The training stage uses a multi-resolution STFT loss function [12], which is the sum of all STFT loss functions using different STFT parameters.

DARCN. DARCN [57] is a recently proposed monaural speech enhancement technique that uses multiple stages and that combines dynamic attention and recursive learning. Experiments are conducted with the open-source code (`https://github.com/Andong-Li-speech/DARCN`) using a non-causal, 3-stage configuration.

Proposed multi-stage SA-TCN Systems. The proposed multi-stage SA-TCN systems are characterized by the number of stages K and the hyperparameters (H, B, R, L). Each K-stage SA-TCN system uses an STFT with a 32 ms Hanning-window, a 32 ms filter length and a 16 ms hop size. As such, $F = 257$. The multi-stage SA-TCN systems are trained using 80 epochs of 4-second utterances from the LibriSpeech corpus and using 100 epochs of variable-length utterances from the VCTK corpus. The proposed multi-stage SA-TCN systems are trained using the Adam optimizer [47] with an initial learning rate of 0.0002. All models use a mini-batch of 16 utterances. For each mini-batch of 16 utterances from the VCTK corpus, the longest utterance is determined and the other utterances are zero-padded to obtain equal-length utterances.

3.3.3 ASR Setup.

The automatic speech recognition (ASR) experiments use a time-delay neural network-hidden Markov model (TDNN-HMM) hybrid chain model [84]. The TDNN models long-term temporal dependencies with training times that are comparable to standard feed-forward DNNs. The data is represented at different time points by adding a set of delays to the input, which allows the TDNN to have a finite dynamic response to the time series input data. This acoustic model is trained using the Kaldi toolkit [85] with the standard recipe (`https://github.com/kaldi-asr/kaldi/tree/master/egs/librispeech/s5`). The ASR acoustic models were trained using 960 hours from the LibriSpeech training set. The word error rate (WER) was measured using the LibriSpeech "test-clean" set.

3.3.4 Evaluation Metrics

The speech enhancement systems are evaluated using the commonly used wide-band *perceptual evaluation of speech quality* (PESQ) score [90, 37, 38], the *short-time objective intelligibility* (STOI) score [99], the scale-invariant signal-to-distortion ratio (SI-SDR) [54], and the CSIG, CBAK and COVL scores. The CSIG score is a signal distortion mean opinion score, the CBAK score measures background intrusiveness, and the COVL score measures the speech quality. The automatic speech recognition performance is measured by determining the word error rate (WER).

3.4 Experimental Performance Results

Extensive experiments have been performed to determine the performance of the proposed multi-stage SA-TCN speech enhancement systems, This section first details the findings of the ablation studies, and then presents the performance results for the multi-stage SA-TCN systems.

3.4.1 Ablation Studies

Ablation studies were performed to fine-tune the multi-stage SA-TCN system's hyperparameters (H, B, R, L), and to analyze the effectiveness of the self-attention and fusion blocks.

The performance of 5-stage SA-TCN systems is measured in terms of PESQ and STOI scores for several hyperparameter configurations. The results are listed in Table 3.1. We observe that it is more effective to increase the number of channels (hyperparameters B and H) in each TCN block than to increase the number of TCN blocks per stack (L). For instance, when $R = 2$ and H and B are doubled, the PESQ score improves from 2.59 to 2.65 and the STOI score improves from 92.36 to 93.02. At the same time, using $L = 8$ instead of $L = 5$ causes a slight degradation of the PESQ score. The performance can also be improved significantly by increasing the number of stacks R. We determined the model size for the larger TCN with $R = 3$ stacks and $L = 8$ TCN blocks per stack, which accounts for about 1.68 M parameters. Each SA block has about 0.2 M parameters and each fusion block has about 0.17 M parameters. If we only consider models with less than 10 million parameters, the model where $(H, B, R, L) = (256, 128, 3, 8)$ performs best. We should also note that there is a trade-off between the performance and the model size.

Next, we investigate the impact of the number of stages K on the performance of a multi-stage SA-TCN speech enhancement system. The motivation for employing multi-stage learning is

R	L	H	B	P	model size	PESQ	STOI
2	5	128	64	3	2.38 M	2.59	92.36
2	5	256	128	3	5.19 M	2.65	93.02
2	8	128	64	3	2.90 M	2.53	92.32
2	8	256	128	3	7.21 M	2.64	93.05
3	5	128	64	3	2.81 M	2.61	92.67
3	5	256	128	3	6.88 M	2.71	**93.40**
3	8	128	64	3	3.59 M	2.60	92.20
3	8	256	128	3	9.91 M	**2.73**	93.37

The best score in a column is bold-faced, the second best is navy blue and the third best is dark pink.

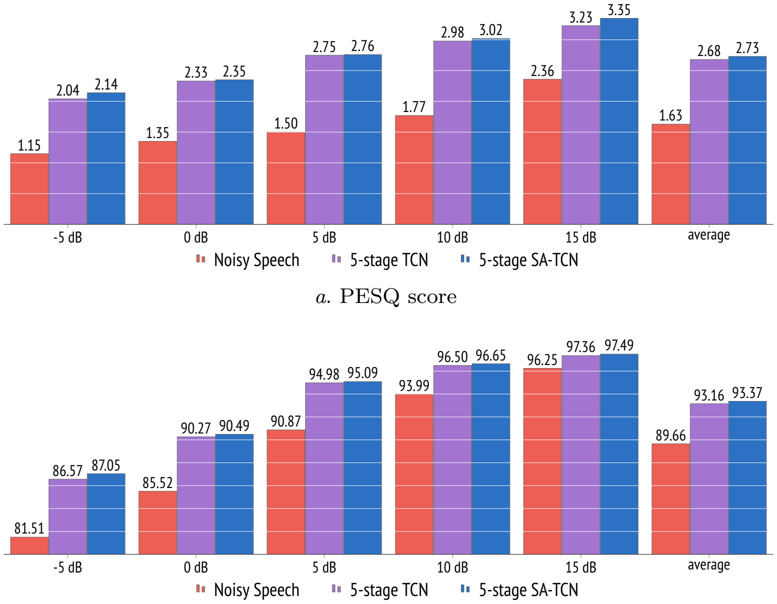
Table 3.1: Performance for Several 5-stage SA-TCN Configurations

that the initial prediction is refined by the next stage. The results in Table 3.2 show that the performance improves step-wise after each stage. For instance, when comparing the first and the fifth stage, it shows that the PESQ score improves from 2.60 to 2.73, and the STOI score improves from 93.08 % to 93.37 %. We also observe that the PESQ score's rate of improvement gradually decreases from 0.5 to 0.1, which suggests that adding further stages has diminishing returns in terms of performance and that a 5-stage SA-TCN system is likely close to the upper bound on performance for this multi-stage TCN-based approach.

Stage	PESQ	STOI
stage 1	2.60	93.08
stage 2	2.65	93.10
stage 3	2.70	93.22
stage 4	2.72	93.33
stage 5	2.73	93.37

Table 3.2: Per-Stage PESQ and STOI Scores for a 5-Stage SA-TCN System

The performance impact of using self-attention was determined using PESQ and STOI scores. The results are shown in Fig. 3.3. On average, a 5-stage SA-TCN system provides a STOI score improvement of 3.5 % and a PESQ score improvement of 1.05 relative to unprocessed noisy speech. The insertion of the SA block prior to the stacked layers of TCN blocks consistently improves PESQ and STOI scores for all SNR conditions: the average PESQ score improves from 2.68 to 2.73 and the average STOI score improves from 93.16 % to 93.37 %. This indicates that the SA block is able to aggregate the frequency context, which is helpful for TCN-based speech enhancement. We also observe that the use of SA blocks show more significant performance gains at low SNR, e.g., at -5 dB, the PESQ score improves from 2.04 to 2.14 and the STOI score improves from 86.57 % to 87.05 %. This also indicates that multi-stage SA-TCN systems are more robust for lower SNR.



a. PESQ score

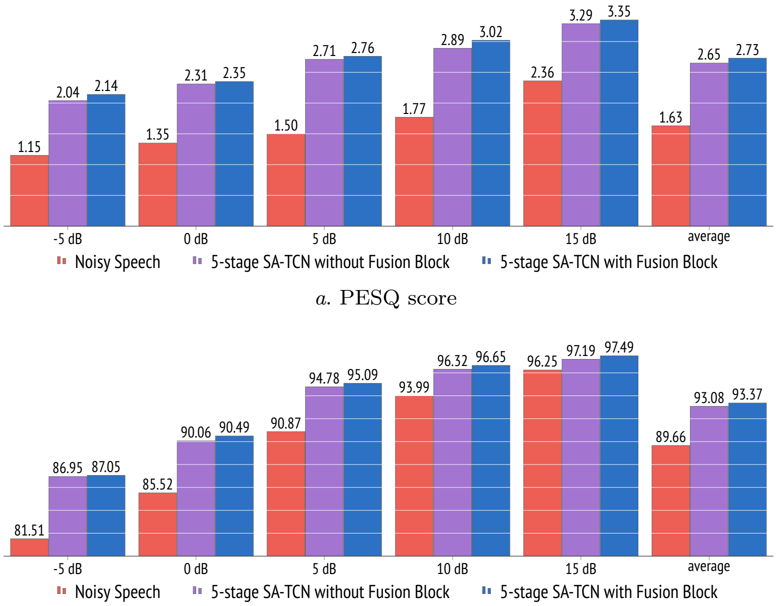
b. STOI score [%]

Figure 3.3: Impact of using a self-attention module in a 5-stage SA-TCN system, where $(H, B, R, L) = (256, 128, 3, 8)$.

The effectiveness of the proposed fusion block, which re-injects original information in stages 3–5 in a 5-stage SA-TCN system to alleviate any speech signal loss, is considered next. The PESQ and STOI scores are shown in Fig. 3.4. It shows that both scores improve for all SNR scenarios. The average PESQ score improves from 2.65 to 2.73, and the average STOI score improves from 93.08 % to 93.37 %. The impact of the fusion block is, as expected, more prominent at lower SNR, when the model not only removes the noise, but can also easily partly remove the speech signal itself.

3.4.2 Baseline System Comparison

Extensive experiments with the proposed multi-stage SA-TCN system and the CRN-based, complex-CNN and DARCN systems were conducted using the LibriSpeech data set. All multi-stage SA-TCN systems use hyperparameters $(H, B, R, L) = (256, 128, 3, 8)$. Table 3.3 shows that all multi-stage SA-TCN systems outperform the baseline systems in terms of the PESQ score for the different noise types and SNR conditions. The results also show that multi-stage SA-TCN systems



a. PESQ score

b. STOI score [%]

Figure 3.4: Impact of using a fusion block in a 5-stage SA-TCN system, where $(H, B, R, L) = (256, 128, 3, 8)$.

with more stages have a better PESQ score. Similarly, Table 3.4.2 shows that the STOI scores of the multi-stage SA-TCN systems are generally better than the baseline systems, and that the best STOI scores are generally obtained for 4-stage and 5-stage SA-TCN systems. Interestingly, even the single-stage SA-TCN system outperforms all baseline systems in terms of PESQ score. Adding more stages improves the overall performance significantly. For instance, the single-stage SA-TCN and the 2-stage SA-TCN have average PESQ scores of 2.47 and 2.67, respectively, and average STOI scores of 92.52% and 92.88%. The best performance is achieved with $K = 5$ stages, with an average PESQ score of 2.73 and an average STOI score of 93.37 %. The proposed 5-stage SA-TCN system has much better PESQ and STOI scores than the baseline systems, which demonstrates the effectiveness of the proposed approach. We also observe that multi-stage learning is more effective at a low SNR. For example, the 5-stage SA-TCN system achieves much better performance for Office and Kitchen Noise at -5 dB, and it also performs well for Babble Noise at low SNR.

Finally, we determined the SI-SDR metrics that quantify speech distortion. Table 3.5 shows that the proposed multi-stage SA-TCN sytems generally outperform the baseline systems. We also observe that the SI-SDR performance for multi-stage SA-TCN systems with $K > 3$ stages decreases slightly, which indicates that the additional stages not only mask the noise, but also distort the speech signal. However, it will be shown next that these speech distortions do not impact the ASR performance.

Noise type	Office Noise					Babble Noise					Kitchen Noise					Average
SNR	-5	0	5	10	15	-5	0	5	10	15	-5	0	5	10	15	
Noisy speech	1.30	1.68	2.01	2.60	3.39	1.06	1.09	1.22	1.37	1.80	1.07	1.18	1.30	1.62	2.10	1.63
CRN	1.90	2.16	2.58	3.07	3.45	1.08	1.20	1.46	1.75	2.24	1.43	1.81	2.19	2.44	2.78	2.09
Complex-CNN	2.14	2.40	2.84	3.01	3.24	1.13	1.30	1.70	2.06	2.54	1.77	2.20	2.46	2.67	2.95	2.30
DARCN	2.23	2.48	3.02	3.35	3.62	1.14	1.33	1.72	1.97	2.52	1.80	2.16	2.45	2.64	2.82	2.35
1-stage SA-TCN	2.29	2.60	3.11	3.38	3.64	1.15	1.38	1.85	2.28	2.81	1.83	2.27	2.69	2.84	2.87	2.47
2-stage SA-TCN	2.55	2.85	**3.46**	**3.65**	**3.84**	1.17	1.44	1.94	2.37	2.89	1.93	2.47	2.94	3.09	**3.39**	2.67
3-stage SA-TCN	2.67	**2.93**	3.43	3.58	3.83	1.17	1.42	1.97	2.39	2.92	1.95	2.47	2.94	3.18	3.30	2.69
4-stage SA-TCN	2.61	2.90	3.42	3.56	3.81	**1.19**	1.46	2.03	2.48	2.95	1.97	2.49	**2.98**	3.11	3.35	2.70
5-stage SA-TCN	**2.74**	2.90	3.38	3.57	3.82	1.18	**1.47**	**2.05**	**2.51**	**3.02**	**2.10**	**2.53**	2.94	**3.19**	3.30	**2.73**

All multi-stage SA-TCN models use hyperparameters $(H, B, R, L) = (256, 128, 3, 8)$. The best score in a column is bold-faced, the second best is navy blue and the third best is dark pink.

Table 3.3: PESQ Scores for Multi-Stage SA-TCN and Baseline Systems Using Samples from the LibriSpeech Corpus

Noise type	Office					Babble					Kitchen					Average
SNR [dB]	-5	0	5	10	15	-5	0	5	10	15	-5	0	5	10	15	
Noisy speech	92.91	96.81	98.50	98.62	99.25	54.94	64.93	80.04	87.10	91.08	84.59	91.75	95.40	98.21	98.86	89.66
CRN	93.02	96.15	97.29	97.55	98.07	54.44	68.15	83.75	90.04	91.98	87.14	92.60	95.57	97.15	97.23	90.23
Complex-CNN	93.23	95.78	97.25	96.91	97.01	60.70	72.92	87.01	91.85	93.26	87.38	92.63	95.11	96.83	96.83	91.09
DARCN	95.08	97.04	98.46	98.44	98.82	62.60	75.20	88.90	91.72	94.41	90.31	93.71	96.60	**98.31**	98.31	92.64
1-stage SA-TCN	94.81	97.09	98.55	98.52	98.76	61.51	74.96	88.62	92.89	94.26	89.37	94.11	96.37	98.08	98.27	92.52
2-stage SA-TCN	94.85	96.98	98.35	98.24	98.89	64.91	76.59	89.89	93.29	94.47	89.45	93.72	96.76	97.67	98.65	92.88
3-stage SA-TCN	95.02	97.23	98.48	98.32	98.92	63.30	75.75	89.92	93.63	94.64	88.71	93.68	96.79	98.12	98.60	92.82
4-stage SA-TCN	94.93	97.14	98.52	98.22	98.75	**65.47**	76.94	**90.33**	93.70	94.73	89.41	94.47	96.74	98.10	98.62	93.10
5-stage SA-TCN	**95.43**	**97.24**	**98.62**	**98.60**	**98.96**	64.26	**77.57**	**90.33**	**93.94**	**94.94**	**90.82**	**94.84**	**96.88**	98.22	**98.78**	**93.37**

All multi-stage SA-TCN models use hyperparameters $(H, B, R, L) = (256, 128, 3, 8)$. The best score in a column is bold-faced, the second best is navy blue and the third best is dark pink.

Table 3.4: STOI Scores for Multi-Stage SA-TCN and Baseline Systems Using Samples from the LibriSpeech Corpus

3.4.3 Automatic Speech Recognition

We conducted automatic speech recognition (ASR) experiments using LibriSpeech to assess the performance of multi-stage SA-TCN systems with up to five stages and determined the word error rate (WER) as well as the WER reduction. The baseline systems are the CRN-based method, and the complex-CNN and DARCN methods. The results are shown in Table 3.6. Our 1-stage SA-TCN system performs slightly worse than the best baseline systems, but the multi-stage SA-TCN methods perform better, and the the proposed 5-stage SA-TCN achieves an absolute improvement of 18.8 %, 8.4 % and 4.6 % relative to CRN, complex-CNN and the DARCN methods, respectively.

SNR [dB]	-5	0	5	10	15	Average
CRN	1.43	5.72	9.63	13.07	16.39	9.28
Complex-CNN	9.66	11.32	13.35	14.49	16.86	13.15
DARCN	11.00	12.65	15.84	17.60	19.87	15.41
1-stage SA-TCN	11.13	**13.40**	16.77	18.05	19.77	15.84
2-stage SA-TCN	10.80	12.71	16.60	17.42	19.44	15.42
3-stage SA-TCN	11.41	13.03	**17.01**	**18.34**	**20.33**	**16.04**
4-stage SA-TCN	10.48	12.69	16.25	17.10	19.51	15.23
5-stage SA-TCN	**11.43**	12.82	16.29	17.41	19.67	15.55

The best score in a column is bold-faced, the second best is navy blue and the third best is dark pink.

Table 3.5: SI-SDR scores for Multi-Stage SA-TCN and Baseline Systems Using Samples from the LibriSpeech Corpus

The ASR results are similar to the STOI performance.

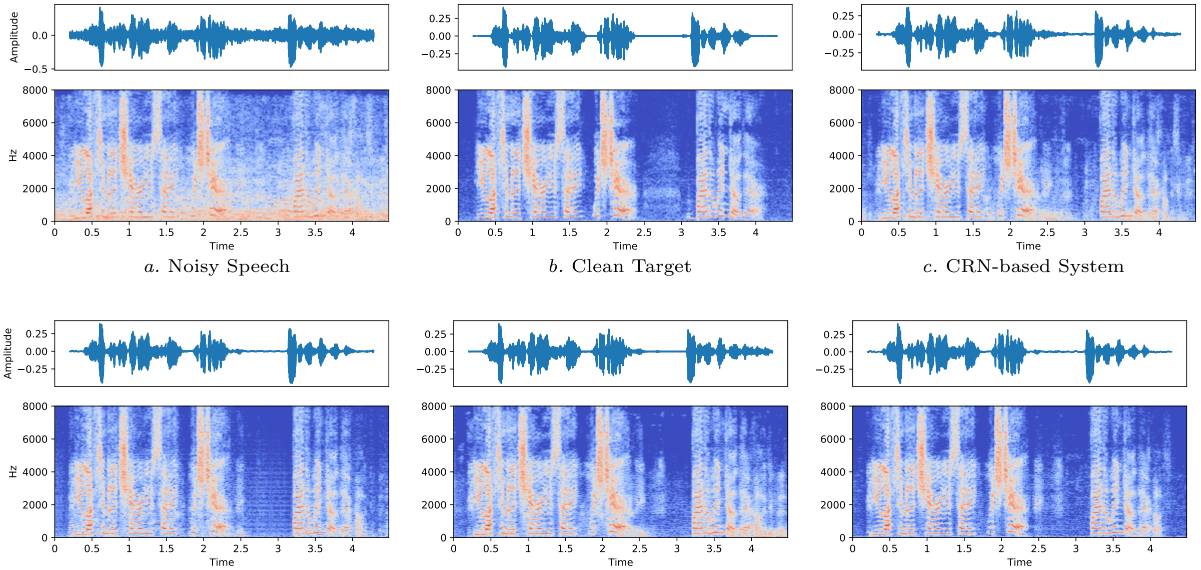
Method	WER [%]	WER reduction [%]
Noisy Speech	32.94	–
CRN	30.86	6.3
Complex-CNN	27.44	16.7
DARCN	26.18	20.5
1-stage SA-TCN	27.86	15.4
2-stage SA-TCN	25.32	23.1
3-stage SA-TCN	26.11	20.7
4-stage SA-TCN	25.27	23.3
5-stage SA-TCN	**24.67**	**25.1**

The best score in a column is bold-faced, the second best is navy blue and the third best is dark pink.

Table 3.6: Speech Recognition Performance of Multi-Stage SA-TCN and Baseline Systems

3.4.4 Spectrogram-Based Visualization

Speech enhancement performance can be assessed using spectrograms. Consider the situation where clean speech is perturbed by Babble noise at an SNR of 5 dB. Fig. 3.5 shows spectrograms of the noisy speech signal, the clean speech target, as well as the CRN-based and complex CNN-based systems, the DARCN system, and the proposed 5-stage SA-TCN enhanced speech system. The spectrograms clearly show that the proposed system is best at suppressing residual noise while preserving the speech patterns.



a. Noisy Speech *b.* Clean Target *c.* CRN-based System

d. Complex-CNN System *e.* DARCN System *f.* 5-Stage SA-TCN System

Figure 3.5: Spectrograms of a sample input mixed with Babble Noise at an SNR of 5 dB and the speech-enhanced signals at the output of SA-TCN and baseline systems.

3.4.5 Speech-Enhancement Benchmark Results

The proposed multi-stage SA-TCN speech enhancement systems are compared with state-of-the-art methods using the publicly available benchmark data set VCTK. As shown in Table 3.7, the proposed multi-stage SA-TCN systems outperform methods that use T-F frequency features, including magnitude, gamma-tone spectral and complex STFT in terms of all the speech enhancement metrics used in this paper. Compared with the recently proposed time-domain method DEMUCS, our proposed method uses fewer parameters and achieves better performance in terms of CBAK and COVL metrics, while the PESQ, STOI and CSIG are slightly worse. The experiments with the VCTK corpus show that adding more stages still provides some incremental performance improvements.

3.5 Summary

In this chapter, we have presented novel multi-stage SA-TCN speech enhancement systems, where each stage consists of a self-attention block followed by R stacks of L temporal convolutional network blocks with doubling dilation factors. The stacks of L TCN blocks effectively perform se-

	model size	feature type	PESQ	STOI	CSIG	CBAK	COVL	SI-SDR
noisy speech	–	–	1.97	0.921	3.35	2.44	2.63	8.45
SEGAN [83] (2017)	43.2 M	Waveform	2.16	0.93	3.48	2.94	2.80	–
Wave-U-Net [71] (2018)	10.2 M	Waveform	2.40	–	3.52	3.24	2.96	–
DFL [21] (2018)	0.64 M	Waveform	–	–	3.86	3.33	3.22	–
MMSE-GAN [97] (2018)	0.79 M	Gamma-tone spectral	2.53	0.93	3.80	3.12	3.14	–
MetricGAN [18] (2019)	1.89 M	Magnitude	2.86		3.99	3.18	3.42	–
MB-TCN [127] (2019)	1.66 M	Magnitude	2.94	0.9364	4.21	3.41	3.59	–
DeepMMSE [128] (2020)	–	Magnitude	2.95	0.94	4.28	3.46	3.64	–
MHSA-SPK [51] (2020)	–	STFT	2.99	–	4.15	3.42	3.57	–
STFT-TCN [52] (2020)	–	STFT	2.89	–	4.24	3.40	3.56	–
DEMUCS [12] (2020)	127.9 M	Waveform	**3.07**	**0.95**	**4.31**	3.40	3.63	–
1-stage SA-TCN	1.88 M	Magnitude	2.84	0.9402	4.16	3.37	3.50	17.98
2-stage SA-TCN	3.76 M	Magnitude	2.96	0.9422	4.25	3.45	3.62	18.19
3-stage SA-TCN	5.81 M	Magnitude	2.99	0.9423	4.27	3.48	3.64	18.38
4-stage SA-TCN	7.86 M	Magnitude	3.01	0.9428	4.27	3.49	3.66	18.40
5-stage SA-TCN	9.91 M	Magnitude	3.02	0.9439	4.29	**3.50**	**3.67**	**18.48**

The best score in a column is bold-faced, the second best is navy blue and the third best is dark pink.

Table 3.7: Performance Evaluation Scores of Multi-Stage SA-TCN and Baseline Systems Using Samples from the VCTK Corpus

quential refinement processing. Multi-stage SA-TCN systems with three or more stages use a fusion block as of the third stage to mitigate any possible loss of the original speech information loss in later stages. The proposed self-attention module is used to provide a dynamic representation by aggregating the frequency context. Extensive experiments were used to fine-tune the hyperparameters. It was shown that both the addition of the self-attention modules and the fusion blocks resulted in better performance. We noted that even the basic 1-stage SA-TCN system performs well and that adding stages improves the speech enhancement scores. The model size increases almost linearly with the number of stages. The relative improvement when adding an additional stage reduces when more stages are added and as such one approaches an implicit upper bound for this approach. The best overall performance with a reasonable model size was obtained with a 5-stage SA-TCN system.

Extensive experiments were conducted using the LibriSpeech and VCTK data sets to determine the performance of the multi-stage SA-TCN speech enhancement systems and to compare the proposed system with other state-of-the-art deep-learning speech enhancement systems. It was shown that the proposed multi-stage SA-TCN methods achieve better performance in terms of widely used objective metrics while having fewer parameters. Speech enhancement, especially in mobile applications, requires computational- and parameter-efficient models. The proposed methods meet this requirement and at the same time provide excellent performance. Spectrograms were used to visualize that the proposed 5-stage SA-TCN systems can remove noise effectively while preserving the speech patterns. The proposed multi-stage SA-TCN systems predict a soft mask at each stage, which can be viewed as an implicit ideal ratio mask (IRM). For speech signals that are dominated by noise, the noise is suppressed gradually in each stage, which is a main reason for the excellent performance. The proposed multi-stage SA-TCN systems are also shown to have excellent ASR performance.

The focus of this chapter is to process and enhance the spectrum magnitude, and the unaltered noisy phase is used when reconstructing the waveforms in the time domain. Recently, several studies have shown that phase information is also important for improving the perceptual quality [77, 58]. Thus, incorporating phase information into the proposed approach may lead to further improvements.

Chapter 4

On Loss Functions for Multi-scale Temporal Convolutional Network-Based Speech Enhancement

In this chapter we investigate multi-scale temporal convolutional network (TCN) for speech denoising. We propose three multi-scale architectures to improve the speech enhancement performance, which consist of 1) TCN-dual, which has two different dilation factors in one dilated convolutional block; 2) TCN-flatten, which uses a fixed number of dilation factors in each dilated convolutional block and the output of each dilation factor are concatenated; 3) TCN-pyramid, which is similar to TCN-flatten, adopts an additional hierarchical feature fusion mechanism.

4.1 Introduction

Recently deep learning based speech enhancement approaches have become the mainstream. Early methods include a recurrent neural network(RNN) [70], a deep auto-encoder [66], a deep neural networks (DNN) [124] and so on. Recent studies consider the use of temporal convolutional network (TCN) for speech enhancement [49, 52] and speech separation [69] as its ability to model longer context dependencies with fewer parameters. For example, in [49], the TCN used in [69] for speech separation was adapted for speech enhancement and integrated in a multi-layer encoder-decoder architecture. [52] proposed to use complex Short-Time Fourier transform (STFT) features for TCN-based speech enhancement. TCN are typically stacked multiple layers to model a longer temporal contextual field, in which the dilation rate in each block is exponentially increased. As the number of layers increases, the resulting large dilation rate makes the model pay more attention to long term dependency. However, the corresponding local information may be neglected in the higher layers. To mitigate these limitations, several multi-scale TCN architectures were proposed. For instance, FurcaNeXt was proposed in [126], which includes several multi-scale gated TCN for speech separation. In [127], a speech enhancement system was proposed that uses a multi-branch TCN, which adopts different dilation rates in each branch and enables the model to learn useful representation by aggregating the information from each branch. In [58], a dual gated TCN was proposed for speech denoising and showed improved performance.

4.2 Multi-Scale TCN Framework

The objective of a speech enhancement module is to filter a received *noisy speech signal* and to generate an *enhanced signal* that is as close as possible to the original speech signal. Let $\boldsymbol{x} \in \mathbb{R}^n$ denote a received n-sample noisy speech signal that is fed to the speech enhancement module, and let $\boldsymbol{y}$ denote the clean speech waveform. The quantitative objective of the speech enhancement module is to output a signal $\hat{\boldsymbol{y}}$ that is as close as possible to the original speech signal $\boldsymbol{y}$.

4.2.1 Overview of the proposed framework

As shown in Fig. 4.1, the proposed framework is adapted from Conv-tasnet architecture [69]. Instead of using a trainable encoder/decoder, Short-Time Fourier transform (STFT)/Inverse STFT is adopted for feature processing. The architecture consists of R stacks of K TCN blocks, where the dilation factor of the i-th TCN block in the stack is given by $\Delta_i = 2^{i-1}$. As such, each stack has a large receptive field, which makes it particularly suited for temporal sequence modeling. Each TCN block has a residual connection between the input and output to reduce the loss of low-level details and to provide hooks for optimization. A sigmoid are applied at the output to generate a

[0-1] mask. The enhanced magnitude is obtained by multiplying noisy magnitude and the predicted mask. When performing the inverse STFT to reconstruct the waveform, we use the phase of the original noisy speech. Next we detail the TCN used in the proposed framework.

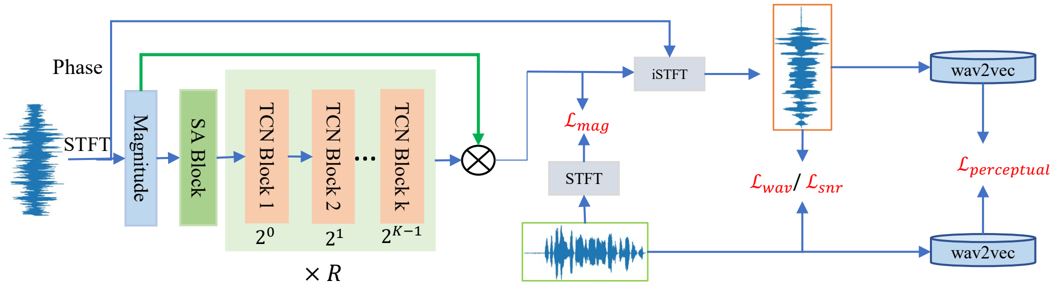


Figure 4.1: The overall framework of the proposed approach. The waveform with green box is the target waveform and the waveform with red box is the enhanced output.

4.2.2 Multi-Scale TCN Block

We explore three different multi-scale TCN blocks. Each TCN block comprises an 1×1 convolution at its input to match the input feature dimension B to the TCN block's internal feature map dimension H, a dilated depth-wise convolution (D-conv) layer with kernel size P and dilation factor $\Delta_i = 2^{i-1}$, where i denotes the order of the TCN block in the stack of K TCN blocks, and a 1×1 convolution layer to reduce the number of channels at the output from H to B. A parametric rectified linear unit (PReLU) activation layer [29] and a batch normalization layer [35] are inserted prior to and after the depth-wise convolution layer to make the training stable. To simplify the figure, we only describe the depth-wise convolution layer (D-Conv) in Fig. 4.2. As we can see in Fig. 4.2, **TCN-dual** consists of two parallel branches, which can model two different dilation rates. For example, if the the total number of TCN blocks is 8, then the first TCN block will have 1 (2^0) and 128 (2^7) dilation rates, where both local and longer context can be learned by the TCN-dual design. For the **TCN-flatten** design, each TCN block has a fixed number of dilation rates, which is the total number of the TCN blocks. Different dilation rates in each branch allow the TCN block to learn the representations from a large effective receptive field. We explore to merge the output of each branch using grouped 1×1 convolution that can reduce the parameters. **TCN-pyramid** is similar to TCN-flatten, the only difference is that a hierarchical feature fusion (HFF) [72] is adopted as shown in Fig. 4.2. The outputs of each dilation rates are hierarchically added before concatenating them, which does not increase the complexity of the TCN block.

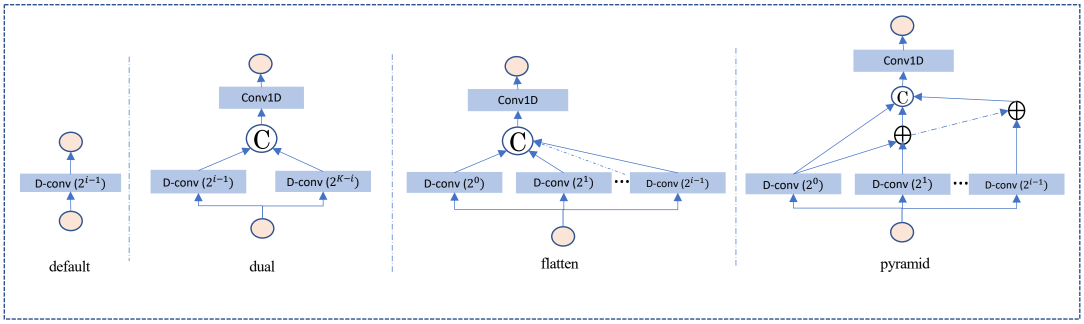


Figure 4.2: The architectures of TCN block.

4.2.3 Loss function

We investigate four loss functions for the proposed multi-scale TCN. The mask-based signal approximation mean absolute error loss $\mathcal{L}_{\text{mag}}$ is defined as:

$$\mathcal{L}_{\text{mag}} = ||\text{mask}\odot\mathbf{X} - \mathbf{Y}||_1, \tag{4.1}$$

where $\mathbf{X}$ and $\mathbf{Y}$ denote the noisy STFT magnitude and target magnitude, and the operator $\odot$ denotes the Hadamard product. The predicted waveform is defined as:

$$\hat{\boldsymbol{y}} = \text{ISTFT}(\text{mask}\odot\mathbf{X}, \Omega), \tag{4.2}$$

where Ω denotes original STFT phase. The L_1waveform loss can be obtained by:

$$\mathcal{L}_{\text{wav}} = ||\hat{\boldsymbol{y}} - \boldsymbol{y}||_1, \tag{4.3}$$

SI-SNR is commonly used as an loss function at the time-domain. $\mathcal{L}_{\text{SI-SNR}}$ is defined as:

$$\begin{cases} \boldsymbol{y}_{\text{target}} \coloneqq (< \hat{\boldsymbol{y}}, \boldsymbol{y} > \cdot \boldsymbol{y})/||\boldsymbol{y}||_2^2, \\ \boldsymbol{e}_{\text{noise}} \coloneqq \hat{\boldsymbol{y}} - \boldsymbol{y}_{\text{target}}, \\ \mathcal{L}_{\text{SI-SNR}} \coloneqq 10\log_{10}\frac{||\boldsymbol{y}_{\text{target}}||_2^2}{||\boldsymbol{e}_{\text{noise}}||_2^2}, \end{cases} \tag{4.4}$$

Recent studies have shown that additional perceptual loss can improve the speech enhancement performance [21, 43, 32]. In particular, wav2vec [95], which is a self-supervised that learns generic speech representations, performs well for speech denoising. We adopt it as one of the loss functions for the proposed framework. In addition, we also explore several combinations of these loss functions and more details can be found in the result section.

4.3 Experiments and results

4.3.1 Data set

To verify the effectiveness of the proposed multi-scale TCN system, we conduct experiments using the LibriSpeech and DNS data sets. The detailed set-up for each data set is detailed below.

LibriSpeech is an open-source corpus that contains 960 hours of speech derived from audio books in the LibriVox project. We select 100 hours of speech data from the "train-clean" data set, 800 sentences from the "dev-clean" data set, and 500 sentences from the "test-clean" data set for training set validation set and test set, respectively. The training set uses 10,000 randomly selected noise sample sequences from the DNS Challenge [87]. The speech data of training and validation sets are mixed with a randomly-selected noise sound from the DNS Challenge noise set with an SNR in the set $\{-5, -4, \cdots, 9, 10\}$ (in dB), three distinct noise types: "babble noise" from the NOISEX-92 corpus [112], and "office noise" and "kitchen noise" from the DEMAND noise corpus [105] are selected for test set. The first channel signal of the corpus is used for data generation. Each clean utterance is distorted by a randomly selected noise type at a randomly selected SNR from the set $\{-5, 0, 5, 10, 15\}$ (in dB).

We also conduct experiments on the **DNS-Challenge dataset** [87]. For DNS-Challenge dataset, we totally generate around 60000 noisy-clean pairs for training with the provided clean utterances and noise sets. The development test set with 150 pairs is adopted for model evaluation.

4.3.2 Model setup

Proposed Multi-scale TCN: The proposed framework takes magnitude as input that is extracted by an STFT with a 32 ms Hanning-window, a 32 ms filter length and a 8 ms hop size. As such, $F = 257$. The hyperparameters (H, B, R, K) are set to (256, 128, 3, 8) and kernel size P is 3.

All the proposed methods in this chapter are causal setting, thus it can be applied to real-time applications.

The multi-scale TCN systems are trained using 80 epochs of 4-second utterances from the LibriSpeech corpus and using 60 epochs of 4-second utterances from the DNS corpus. The proposed systems are trained using the Adam optimizer [47] with an initial learning rate of 0.0002. The proposed models use a mini-batch of 16 utterances for Librispeech and a mini-batch of 20 utterances for DNS corpus.

4.3.3 Evaluation Metrics

The speech enhancement systems are evaluated using the commonly used wide-band *perceptual evaluation of speech quality* (PESQ) score [90, 37, 38], the *short-time objective intelligibility* (STOI) score [99], the scale-invariant signal-to-distortion ratio (SI-SDR) [54], and the CSIG, CBAK and COVL scores. The CSIG score is a signal distortion mean opinion score, the CBAK score measures background intrusiveness, and the COVL score measures the speech quality.

4.3.4 Result

We first compare the performance of proposed approaches using different loss functions mentioned 4.2.3. We observe that using $\mathcal{L}_{\text{wav}}$ and $\mathcal{L}_{\text{SI-SNR}}$ can get better SI-SDR performance. This is due to SDR mainly reflects the similarity between the original and reconstructed waveforms in the time domain. Using $\mathcal{L}_{\text{mag}}$ and $\mathcal{L}_{\text{perceptual}}$ can achieve better PESQ and STOI metrics. Moreover, the combination of $\mathcal{L}_{\text{mag}}+\mathcal{L}_{\text{perceptual}}$ and $\mathcal{L}_{\text{mag}}+\mathcal{L}_{\text{wav}}$ can further improve the performance in terms of PESQ and STOI. For example, compared with the TCN-pyramid system only using $\mathcal{L}_{\text{mag}}$, the TCN-pyramid system using $\mathcal{L}_{\text{mag}}+\mathcal{L}_{\text{perceptual}}$ improves the PESQ and STOI from 2.79 to 2.85 and 93.77% to 93.80%, respectively. We also observe that TCN-pyramid achieves the best performance in contrast to TCN-dual and TCN-flatten. This indicates that using hierarchical feature fusion and parallel dilate rates are effective.

Next, we compare our proposed methods with other state of the art systems on DNS challenge dataset. The baseline systems list in Table 4.2 are the top systems in INTERSPEECH2020 DNS challenge [88]. NSnet [123] is the official baseline system. DCCRN [34] is a complex-domain neural network, where both CNN and RNN structures can handle complex-valued operation. It

Models		PESQ	STOI	SI-SDR
Noisy		1.63	89.66	5.03
TCN-dual	①mag_loss	2.72	93.54	11.56
	②snr_loss	2.50	92.75	16.14
	③wav_loss	2.45	92.83	14.73
	④perceptual_loss	2.65	93.17	11.40
	①+②	2.47	92.97	13.74
	①+③	2.80	93.49	13.74
	①+④	2.78	93.55	15.66
TCN-flatten	①mag_loss	2.77	93.58	15.18
	②snr_loss	2.56	92.98	14.67
	③wav_loss	2.48	92.81	15.53
	④perceptual_loss	2.75	93.42	12.38
	①+②	2.64	93.26	12.76
	①+③	2.80	93.64	15.55
	①+④	2.83	93.78	13.29
TCN-pyramid	①mag_loss	2.79	93.77	14.77
	②snr_loss	2.48	92.94	15.44
	③wav_loss	2.40	92.79	15.90
	④perceptual_loss	2.75	93.43	12.78
	①+②	2.64	93.04	13.81
	①+③	2.81	93.57	13.66
	①+④	2.85	93.80	14.60

Table 4.1: Performance comparison for proposed multi-scale TCN using samples from the LibriSpeech corpus.

ranked first for the real-time-track and second for the non-real-time track. PoCoNet [36] architecture proposed to use the frequency-positional embeddings, which is able to more efficiently build frequency-dependent features in the early layers. It ranked first for the non-real-time-track. DTLN [121] proposed a dual-signal transformation LSTM network (DTLN) for real-time speech enhancement, which also achieved the state of the art performance.

As shown in Table 4.2, all of our proposed methods can achieve competitive results compared with previous SOTA systems. In particular, the proposed TCN-pyramid is close to the top-1 performance in terms of WB-PESQ and NB-PESQ. This illustrates the effectiveness of our proposed multi-scale approach. We should also note that with large training data, the improvement of TCN-Pyramid over TCN-default becomes smaller. Compared TCN-flatten with TCN-pyramid, it further verified the effectiveness of the HFF techniques.

	WB-PESQ	NB-PESQ	CBAK	COVL	CSIG	SI-SDR	STOI
Noisy	1.58	2.16	2.53	2.35	3.19	9.07	91.52
NSnet [123]	1.81	2.68	2.00	2.24	2.78	12.47	90.56
DTLN [121]	-	3.04	-	-	-	-	-
DCCRN [34]	-	3.27	-	-	-	-	-
PoCoNet [36]	2.75	-	3.04	3.42	4.08	-	-
TCN-Default	2.71	3.24	3.53	3.43	4.21	16.43	96.02
TCN-Dual	2.72	3.25	3.51	3.47	4.19	16.08	96.07
TCN-Flatten	2.70	3.24	3.53	3.46	4.20	16.43	95.60
TCN-Pyramid	2.73	3.25	3.54	3.48	4.21	16.45	96.03

Table 4.2: Performance comparison for proposed multi-scale TCNs using magnitude loss and perceptual loss and other state of the art systems using samples from the DNS Corpus.

4.4 Summary

In this chapter, we have presented three multi-scale TCN architectures for speech enhancement. Multiple training targets are investigated in the proposed approaches. Experiments show that the combination of magnitude loss and perceptual loss can achieve the best performance in terms of the commonly used speech enhancement metrics. We also compare the proposed systems with the top systems on INTERSPEECH2020 DNS challenge. It was shown that the proposed approaches can achieve comparable performance in contrast to the top-1 system. The future work could be explored to combine the multi-stage and multi-scale into a unified framework to improve the performance.

Chapter 5

Speaker-Aware Speech Enhancement with Self-Attention

This chapter presents the speaker-aware speech enhancement. The work presented in this chapter has been accepted by European Signal Processing Conference, EUSIPCO 2021.

5.1 Introduction

Recently, modeling to learn the acoustic noisy-clean speech mapping has been enhanced by including auxiliary information such as visual cues [31], phonetic and linguistic information [59, 67], and speaker information [8]. In particular, the utilization of three kinds of broad phonetic class (BPC) information for speech enhancement can achieve notable improvements [67]. In [8], a speaker-aware deep denoising auto-encoder (SaDAE) extracts speaker representation from the noisy input using a DNN model. Target speaker extraction was investigated in [119, 40, 86].

In this chapter, we first visualize the impact of the quality of a clean speech reference signal on speaker representation. Given that it is generally possible to collect a few seconds of clean reference speech in applications, e.g., similar to a smart virtual assistant that needs a few-second clean speech record during its setup stage, or extracted from (prior) high-SNR recordings, it is worthwhile investigating how a few seconds of clean reference can be best used to improve speech enhancement performance. In this chapter, we propose a novel speaker-aware speech enhancement

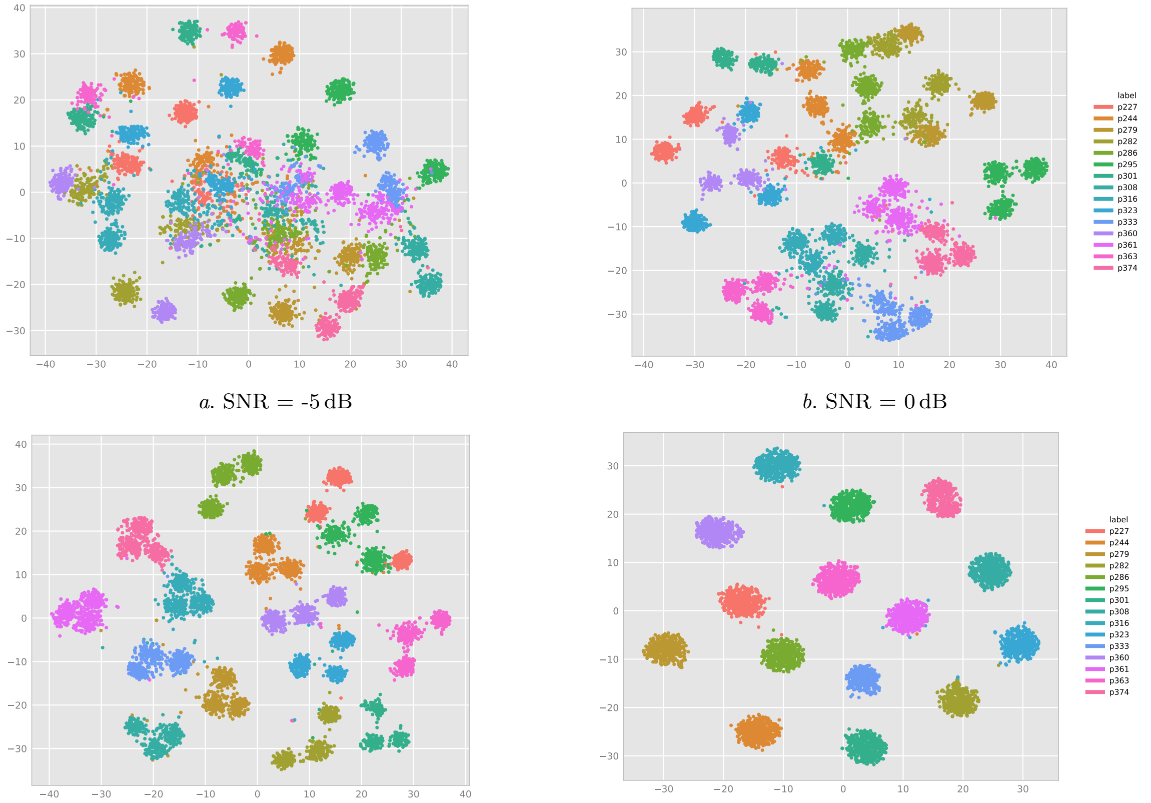
(SASE) method that extracts speaker information from a clean reference using long short-term memory (LSTM) layers, and then uses a convolutional recurrent neural network (CRN) to embed the extracted speaker information. The SASE framework is extended with a self-attention mechanism. Extensive simulations are performed using the Valentini-Botinhao corpus [109] to determine the performance of the proposed SASE method. It will be shown that a few seconds of clean reference speech is sufficient, and that the proposed SASE method performs well for a wide range of scenarios.

5.2 Speaker Embedding

The need for accurate speaker information is visualized by an experiment with fifteen speakers from the Valentini-Botinhao corpus [109], where two noise sources from the DEMAND corpus were added at an SNR of -5 dB, 0 dB, and 5 dB. Fig. 5.1 shows the t-distributed stochastic neighbor embedding (t-SNE) [111] of the speaker embedding information affected by noise. One clearly sees that speaker embedding information is very sensitive to noise. To mitigate the effects of noise, we propose to use clean reference speech, and show that a few seconds suffice to properly extract speaker embedding information. Given that it is often feasible to use a few seconds of clean reference speech in real applications, e.g., from pre-recorded training samples or from prior high SNR recordings, it is worth investigating how the availability of a few seconds of clean reference can be best used to improve speech enhancement performance.

5.3 Proposed SASE Framework

We propose a novel speaker-aware speech enhancement (SASE) system that uses a short clean-speech reference. The system consists of three components: a pre-trained speaker embedding extractor to process the reference clean speech, a CRN-based speech enhancement module, and a self-attention module. The CRN comprises a convolutional encoder-decoder structure which extracts high-level features with a 2-D convolution, and a long short-term memory (LSTM) layers to capture long-span dependencies in temporal sequences. A block diagram is shown in Fig. 5.2.



a. SNR = -5 dB

b. SNR = 0 dB

c. SNR = 5 dB

d. Clean Speech

Figure 5.1: Example of t-SNE visualization for speaker embedding of 15 speakers for various SNR conditions using two noise types from the DEMAND corpus [105].

5.3.1 Speaker embedding extractor

The speaker embedding extractor, proposed in [114], is shown to perform well and is used here. It consists of three LSTM layers with 768 nodes in each layer and one linear layer with a 256-dimensional output.

The pre-trained model (`https://github.com/mindslab-ai/voicefilter`) is trained using the VoxCeleb2 data set [9], which comprises records of thousands of speakers. The model takes as input a Mel-spectrogram, which is extracted using a Short Time Fourier Transform (STFT) with an 80 ms window and a 40 ms hop size. The model achieves a 7.4 % equal error rate on the VoxCeleb1 test data set (first eight speakers of the data set).

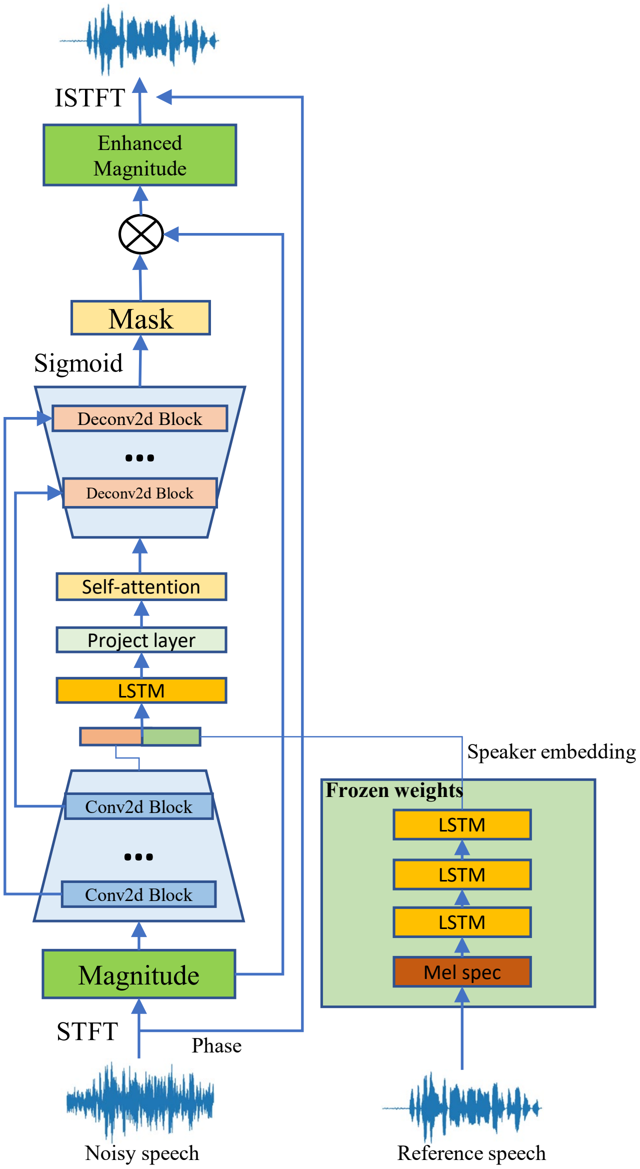


Figure 5.2: Proposed SASE framework.

5.3.2 Self-Attention Module

Self-attention [113] is an efficient context information aggregation mechanism that operates on the input sequence itself and that can be utilized for any task that has a sequential input and output. Consider an 4-dimensional input $\mathbf{X}$ of shape $[B, C, T, F]$, where B, C, T, and F denote the batch size, number of channels, and the time and frequency dimensions, respectively. The self-attention layer takes $\mathbf{X}$ as input and uses three 1×1-convolutions to form the query $\mathbf{Q}$ and the key-value pair $(\mathbf{K}, \mathbf{V})$, where $\mathbf{Q}$ and $\mathbf{K}$ have shape $[B, C', T, F]$, and $\mathbf{V}$ has shape $[B, C, T, F]$. To reduce memory requirements, we use $C' = C/8$. Next, $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are reshaped to form 3D matrices (including batch size).

In order to compute the attention component $\mathbf{A}$, we first compute the weight $\mathbf{W}$, given by

$$\mathbf{W} = \mathbf{Q}^{\mathsf{T}}\mathbf{K}, \tag{5.1}$$

and then use the soft-max function $\sigma(\cdot)$ to obtain $\widehat{\mathbf{W}} = \{\widehat{W}_{i,j}\} = \sigma(\mathbf{W})$, i.e.,

$$\widehat{W}_{i,j} = \frac{\exp(W_{i,j})}{w_j}, \text{ where } \; w_j = \sum_{i=1}^{T \cdot F} \exp(W_{i,j}). \tag{5.2}$$

The attention component $\mathbf{A} \in \mathbb{R}^{B \times C \times T \times F}$ is now determined using

$$\mathbf{A} = \widehat{\mathbf{W}}\mathbf{V}^{\mathsf{T}}, \tag{5.3}$$

The attention module outputs $\widehat{\mathbf{X}} = \mathbf{X} + \delta\mathbf{A}$, where δ is a learnable scalar with initial value zero.

5.3.3 Proposed SASE framework

The SASE framework, shown in Fig. 5.2, has three main components: an encoder-decoder based CRN, a LSTM-based speaker embedding extractor and a self-attention module. The encoder of the CRN consists of five 2-D convolutional blocks, each of which includes a 2-D convolutional layer, a batch normalization layer [35], and exponential linear units (ELUs) [10]. The decoder uses five 2-D deconvolutional blocks to convert the low-resolution features into high-resolution spectrograms. Each deconvolutional block consists of a 2-D transposed convolutional layer, followed by batch normalization and the ELU activation. We include skip connections from each encoder layer to its corresponding decoder layer, in order to avoid losing fine-resolution details and to facilitate optimization. There are two LSTM layers between the encoder and decoder to capture long-term temporal dependencies.

Training Flow. The proposed SASE method takes noisy speech and reference clean speech as input. The reference clean speech is fed into the speaker embedding extractor to obtain speaker information. The noisy speech is fed into the encoder to determine the low-resolution features. The concatenation of the speaker representation and the encoder output are then fed into LSTM layers. The LSTM output is also fed into the self-attention module. The attention output is then followed by the encoder. We apply a sigmoid at the encoder output to generate a [0-1] mask. The following loss function, referred to as SA-MSE, is used during the training stage:

$$\mathcal{L} = \|\mathbf{M}\odot\mathbf{X} - \mathbf{S}\|_{2.} \tag{5.4}$$

where **X** and **S** denote the magnitude of the noisy speech and clean speech signals, respectively, and the operator $\odot$ denotes the Hadamard product. The mean squared error (MSE) loss function that is determined using the clean and predicted magnitude directly is referred to as SM-MSE. When performing the inverse STFT to reconstruct the waveform, we use the phase of the original noisy speech.

5.4 Experiments and Results

In the following, the data set, model set up and the evaluation metrics are detailed. The results will be discussed at the end of this section.

5.4.1 Data Set

The database used here is derived from the Valentini-Botinhao corpus [109]: 84 speakers and two speakers in the original data set are used for training and test, respectively. Each speaker fragment consists of about 10 different sentences. The noisy training set used here considers 40 conditions: 10 noise types (two artificial noise types and eight noise types selected from the Demand database [105]), where each noise type is considered at an SNR of 0 dB, 5 dB, 10 dB, 15 dB. For the test set, a total of 20 different conditions are considered: five types of noise (all from the Demand database) with four SNRs each (2.5 dB, 7.5 dB, 12.5 dB and 17.5 dB). There are around 20 different sentences in each condition for each test speaker. The test set condition is totally different with the training set, as it uses different speakers and conditions. For each speaker, we generate a 60-second segment as clean reference speech. The clean reference is processed by removing the silence part. After holding out the utterance for clean reference, there are 722 sentences in total for testing. During the training stage or testing stage, we randomly choose a small segment from the clean reference for the given segment size, e.g., 2 s, 4 s, 6 s, and 8 s.

5.4.2 Model Setup

The baseline systems considered here are the LSTM- and CRN-based speech enhancement methods. The LSTM baseline model consists of two LSTM layers with 768 nodes each, followed by a fully-connected output layer that reduces the dimension to 161. The CRN-based method consists of five conv2d blocks with filters of size 3×2 each and [16, 32, 64, 128, 128] output channels, respectively. This output is post-processed by two LSTM layers with 512 nodes each, followed by five deconv2d blocks with filter size 3×2 each and output channels [128, 64, 32, 16, 1], respectively.

The proposed SASE method has a similar encoder-decoder as the CRN-based method. The speaker representation (256-D) and the encoder output (512-D) are concatenated and then fed into two LSTM layers of 768 nodes each. The output is then projected onto 512 feature dimensions and reshaped to match the encoder output, and then post-processed by the self-attention module and the decoder.

The feature input for all models is a spectral magnitude vector of length 161 of the noisy speech signal, which is computed using a STFT with a 20 ms Hamming window and a 10 ms window shift. All models are trained using the Adam optimizer [47] with an initial learning rate of 0.0006. A mini-batch size of 32 utterances is used for all models except for SASE with attention. SASE with attention uses mini-batch size of 16 utterances. We zero-pad all utterances to have the same length as the longest utterance within a mini-batch.

5.4.3 Evaluation Metrics

The speech enhancement systems are evaluated using the commonly used *perceptual evaluation of speech quality* (PESQ) score [90, 37, 38], the *short-time objective intelligibility* (STOI) score [99], the scale-invariant signal-to-distortion ratio (SI-SDR) [54], and the CSIG, CBAK and COVL scores. The CSIG score is a signal distortion mean opinion score, the CBAK score measures background intrusiveness, and the COVL score measures the speech quality.

	Loss	Model size	PESQ	STOI	SI-SDR	CSIG	CBAK	COVL
Noisy Speech	–	–	1.970	92.06	8.51	3.35	2.45	2.63
LSTM	SM-MSE	7.71 M	2.608	93.44	16.36	2.91	3.10	2.74
CRN		4.69 M	2.598	93.49	16.52	3.31	3.14	2.94
SASE (2s)		10.33 M (12.13 M)	2.636	93.67	16.80	3.42	3.18	3.02
SASE (4s)			2.627	93.72	16.73	3.44	3.18	3.02
SASE (6s)			2.649	93.80	16.93	3.48	3.20	3.05
SASE (8s)			2.651	93.72	16.84	3.52	3.19	3.07
LSTM	SA-MSE	7.71 M	2.614	93.65	16.70	3.96	3.19	3.29
CRN		4.69 M	2.658	93.87	16.67	4.02	3.22	3.34
SASE (2s)		10.33M (12.13 M)	2.702	93.95	16.86	4.08	3.26	3.40
SASE (4s)			2.699	**94.07**	16.97	**4.09**	3.27	3.40
SASE (6s)			2.696	94.00	16.92	4.08	3.26	3.40
SASE (8s)			2.693	93.98	17.05	4.08	3.27	3.39
SASE (2s) + attn	SA-MSE	10.35M (12.13 M)	2.670	93.92	17.14	4.05	3.26	3.36
SASE (4s) + attn			2.706	94.02	17.34	4.05	3.29	3.38
SASE (6s) + attn			**2.756**	94.05	**17.35**	**4.09**	**3.32**	**3.43**
SASE (8s) + attn			2.703	93.97	17.23	4.05	3.28	3.38

The acronyms SM-MSE and SA-MSE denote spectral mapping and mask-based signal approximation with MSE loss, respectively. The values in parenthesis specify the duration of the reference speech signals. The best score in a column is **bold-faced**, the second best is navy blue and the third best is dark pink.

Table 5.1: Performance Scores for the Proposed SASE and Baseline Systems

5.4.4 Experiments and Results

We first investigate the performance of all models by determining the mean squared error (MSE) loss on the predicted and clean magnitude directly, which is denoted as SM-MSE. The results are provided in Table 5.1. The performance metrics for the CRN-based method are better than the LSTM-based method, except in terms of PESQ. The proposed SASE approach outperforms the CRN baseline system, even with only 2 s reference clean speech. The best performance when applying the SM-MSE loss function is achieved by SASE with 8 s reference speech. This indicates that additional speaker information is useful to further improve speech enhancement performance. Next, we replace the SM-MSE loss function by a mask-based signal approximation loss function (SA-MSE) at the training stage. Table 5.1 shows that all SA-MSE-based loss models perform better than the models that use the SM-MSE loss function, in particular for the PESQ, CSIG and COVL metrics. For instance, relative to SASE with 2 s reference speech using SM-MSE loss, the PESQ score improves from 2.636 to 2.702 and the COVL score improves from 3.02 to 3.40.

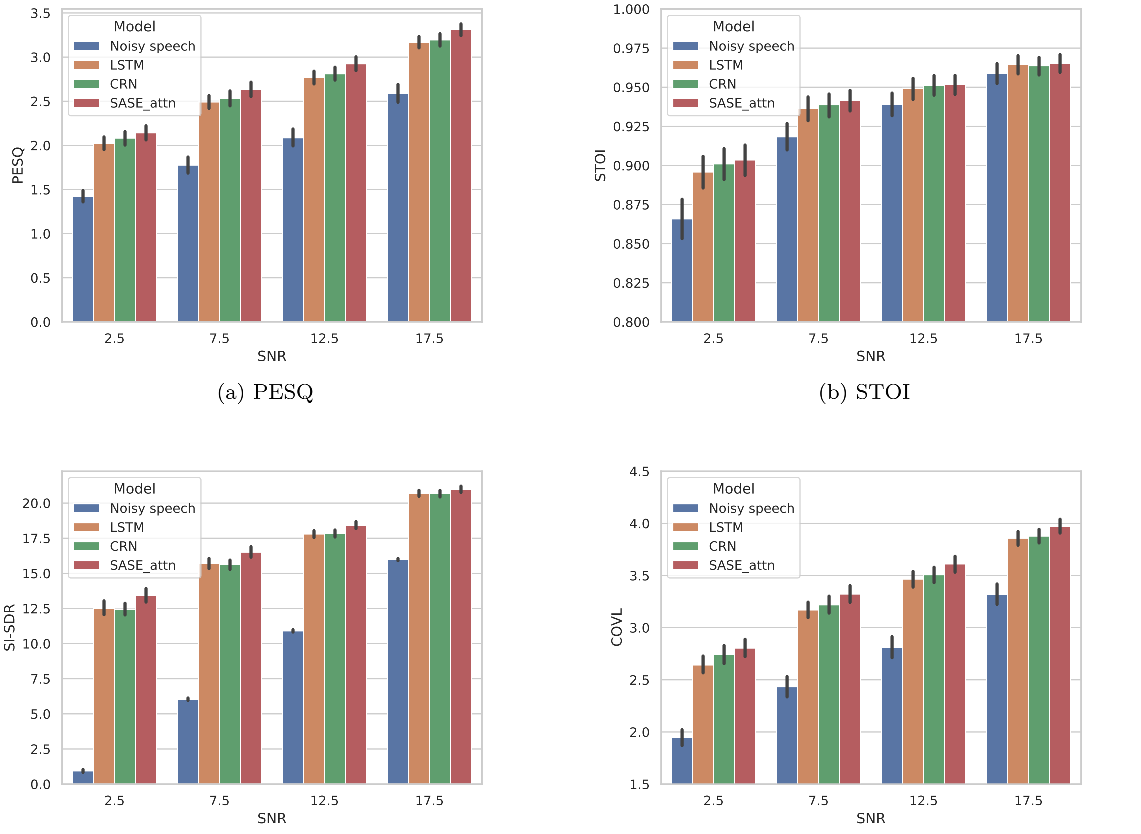
Further adding self-attention can boost the performance as well in terms of most metrics. We observe that adding self-attention improves the SI-SDR consistently for all SASE-based approaches. The best PESQ, SI-SDR, CISG, CBAK, and COVL scores are achieved by SASE with a 6-second reference speech signal.

The detailed PESQ, STOI, SI-SDR scores for the baseline systems and the proposed SASE method with self-attention (6s) with respect to the four SNR conditions considered here are shown in Fig. 5.3. One can clearly see that the proposed SASE method is more effective at lower SNR conditions. This suggests that additional speaker information provides important cues to distinguish the speech and noise at high noise conditions.

Model complexity. The proposed SASE method has more parameters than the baseline systems, because of the three LSTM layers of the speaker embedding extractor. It is planned to replace the LSTM-based speaker embedding extractor with an extractor that uses CNN models with fewer parameters.

5.5 Summary

In this chapter, we presented and validated a novel speaker-aware speech enhancement method that uses a few seconds of reference clean speech. We first compared the proposed SASE with baseline systems using spectral mapping-MSE and mask-based signal approximation-MSE loss, respectively. The experimental results indicate that the proposed SASE system outperforms the baseline systems using both loss functions. The results also show that using mask-based signal approximation loss is better than spectral mapping-MSE loss. Adding self-attention achieves the best performance in terms of most metrics, especially for SI-SDR metric. We tested the proposed SASE approach with four different reference-speech durations. All achieved better performance in comparison with the CRN baseline, which demonstrates the effectiveness of the proposed method.



(a) PESQ

(b) STOI

(c) SI-SDR

(d) COVL

Figure 5.3: The detailed scores for baseline systems and the proposed SASE framework with self-attention for four tests with different SNR conditions.

Chapter 6

N2N-SE: Noise2Noise for Speech Enhancement

This chapter presents a novel noise-to-noise speech enhancement method, referred as N2N-SE, that is trained with actual speech without the need for clean speech targets.

6.1 Related Work

Speech Enhancement In [124], the SEDNN architecture was introduced, which deploys a deep neural network (DNN) as a regression model and trains it using the log-power spectral features of noisy- and clean-speech data pairs. This method outperformed prior techniques such as such as global variance equalization and noise-aware training strategies in terms of perceptive and objective measures. In [118], a DNN-based architecture was proposed that uses a multi-objective learning and ensemble (MOLE) framework, which shows that one can improve the performance by combining two compact DNNs via boosting. Such DNN-based methods are commonly referred to as feature-mapping methods. Other speech enhancement methods are based on mask-learning [74, 120], where a DNN is used to estimate the ideal ratio mask or the ideal binary mask based on the noisy input features. The mask is used to filter noisy speech signals and recover clean speech signals.

In [83], a speech enhancement generative adversarial network (SEGAN) was introduced to train the model directly based upon receiving raw audio data in an end-to-end fashion, and it shows significant performance gains in terms of perceptual speech quality metrics compared with previous work. Similarly, [79] leverages GAN framework using a standard DNN, and they demonstrate the performance improvement via L_1 norm instead of using L_2 norm. In [13], the SEGAN concept is extended to the spectral domain by modeling noise patterns, and it is shown that this further improves performance.

Denoising and Restoration In [108], it is shown that the structure of a convolutional neural network (CNN) can be used for image restoration without requiring additional training data due to the prior distribution of natural images. The CNN was fed with a random but constant noise input and trained to approximate a single noisy image as output. The network could produce a cleanly denoised image if training processing is stopped at the right moment before convergence. The Noise-to-Noise (N2N) framework proposed in [56] attempts to learn a mapping between pairs of independently degraded versions of the same training image. The networks that are trained this way aim to learn the targets' invariant signals, and they could produce the denoised image when pairs of noisy images are available. However, the acquisition of such pairs with same signal only possible for static scenes. To address this limitation, a novel training scheme called NOISE2VOID (N2V) [53] was proposed. N2V is a self-supervised training approach, which can be used in the situations for which neither noisy images pairs nor clean target images are available.

6.2 Proposed N2N-SE Framework

A novel noise-to-noise speech enhancement (N2N-SE) method is proposed, which takes noisy speech as input and outputs the clean speech signal. The overall objective is to learn the expectations among different noisy speech samples so that the model can capture the invariant signal, which is the clean speech signal. The N2N-SE framework consists of a speech enhancement module and a noise conversion module as shown in Fig. 6.1.

6.2.1 Theoretical Analysis

The enhancement module takes noisy speech signals as input, and is tasked to reconstruct the signal from multiple corrupted speech signals. Let $\mathcal{P}^{(M)}$ denote a set of M different noise types or augmentations. We consider the input $\hat{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\epsilon}$, where $\boldsymbol{x}$ denotes the clean speech signal,

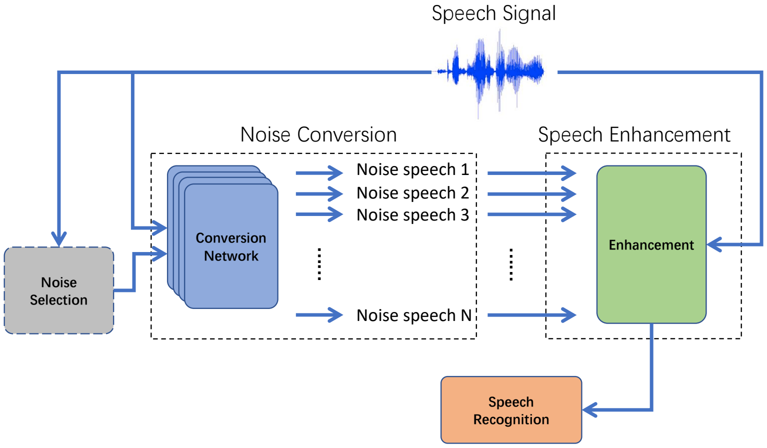


Figure 6.1: The proposed N2N-SE framework, which comprises a noise conversion module and a speech enhancement module. The noisy speech is first fed into noise conversion network to generate multiple new noisy speech as the targets for the speech enhancement network. The enhanced speech is then fetched into ASR for post-processing based on applications.

$\boldsymbol{\epsilon} \sim \mathcal{P}^{(M)}$ denotes the noise that is mixed with the original signal. Note that we only observe $\hat{\boldsymbol{x}}$ and that our objective is to recover $\boldsymbol{x}$.

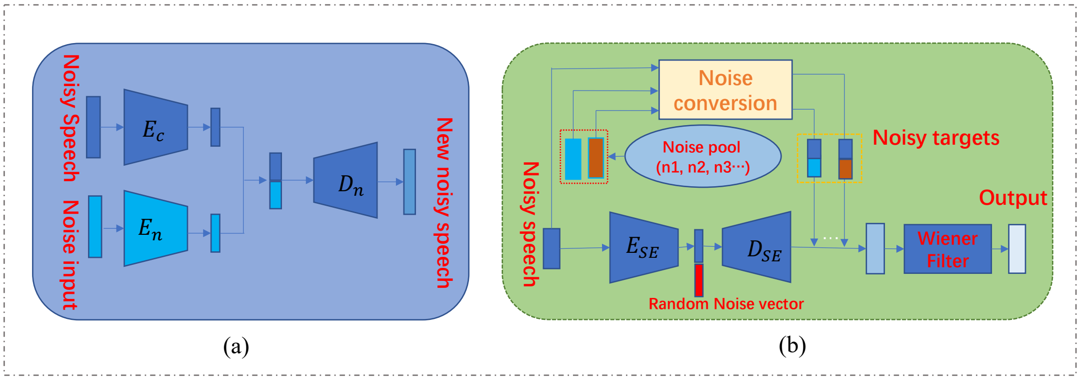
Recently, a novel approach to restore images by only processing noisy samples was introduced in [56, 53]. A similar idea was applied to speech enhancement [1]. These methods use the hypothesis that the noise has a zero mean, i.e., $\mathbb{E}(\boldsymbol{\epsilon}) = 0$, and therefore

$$\mathbb{E}(\hat{\boldsymbol{x}}) = \mathbb{E}(\boldsymbol{x}) + \mathbb{E}(\boldsymbol{\epsilon}) = \boldsymbol{x} \tag{6.1}$$

However, this is generally not true in the speech domain, since a speech signal can be non-stationary with an arbitrary distribution. In [1], it is assumed that dual mics are used to collect multiple noisy speech sources. Here, we rewrite (6.1) as a vector which operates on the raw audio signal:

$$\mathbb{E}(\hat{\boldsymbol{x}}) = \mathbb{E}(\boldsymbol{x}) + \mathbb{E}(\boldsymbol{\epsilon}) = \mathbb{E}(\boldsymbol{x}) + \frac{1}{M}\Sigma_i \epsilon_i \tag{6.2}$$

when we have a sufficient number of noise variants mixed with the clean speech signal, we can still reconstruct the clean speech $\boldsymbol{x}$ from corrupted samples. In this chapter, we consider M to be number of noise types. The second term goes to zero as the number of samples grows. Therefore, the model's target is the same source signal that is mixed with multiple noise types. The model g_θ can be simply



(a)

(b)

Figure 6.2: Model structure of the noise conversion module (a) and speech enhancement module (b).

trained in a batch:

$$\nabla_\theta \mathbb{E}_{\epsilon_i \sim \mathcal{P}^{(M)}}[\mathcal{L}(g_\theta(x+\epsilon_0), x+\epsilon_i)] = \\ \mathbb{E}_{\epsilon_i \sim \mathcal{P}^{(M)}}[\nabla_\theta \mathcal{L}(g_\theta(x+\epsilon_0), x+\epsilon_i)] \tag{6.3}$$

In practical experiments, we often have a limited number of noise types, and as such the second term in (6.2) could create a bias. Assuming that μ_ϵ and σ_ϵ denotes to the mean and variance of noises, based on the central limited theory, when M becomes efficiently large, this bias term (the mean of the ϵ values) would be normally distributed. Hence, we could approximate Equation 6.2 as $\mathbb{E}\left(\boldsymbol{x}\right)+\omega$, where $\omega \sim \mathcal{N}(\mu_\epsilon, \sigma_\epsilon/\sqrt{M})$. Then, we can simply use a linear filter such as a Wiener filter to remove this bias term. The upper bound of noise2noise based speech enhancement would be the mapping between noisy speech with clean target.

6.2.2 Noise Conversion Module

We propose to use a noise conversion module to generate multiple types of noisy speech as shown in the left part of Fig. 6.2. The noise conversion module takes a noisy speech signal $\hat{\boldsymbol{x}}$ and a noise source ϵ_i as input, and generates a new speech signal $\hat{\boldsymbol{x}}'_i$ with additive noise from ϵ_i, which can be represented as

$$f(\boldsymbol{x}+\epsilon_0, \epsilon_i) \rightarrow \boldsymbol{x}+\epsilon_i, \text{ where } \epsilon_i \sim \mathcal{P}^{(M)}, \tag{6.4}$$

where $f(\cdot)$ is an auto-encoder where both the encoder and decoder are a convolution neural network (CNN). The encoder consists of a content encoder E_c that captures phonetic information, and a noise encoder E_n that generates the noise embedding. The computational procedure is outlined in Algorithm 1. The noisy speech signal $\hat{\boldsymbol{x}}$ is fed to the content encoder and transformed to content embedding c_x; the noise signal ϵ_i, which is selected from the noise pool $\mathcal{P}^{(M)}$, feeds into the noise encoder to generate noise embedding c_n. Next, c_x and c_n are concatenated and then fed into the decoder to generate a new noisy speech signal. Note that we train the noise encoder first. During the training of the noise conversion module, the gradients do not update the weights of encoder E_n. We enumerate ϵ_i from $\mathcal{P}^{(M)}$, and store the generated $\hat{\boldsymbol{x}}'_i$ as speech enhancement targets in the generated speech set $\mathcal{D}$.

6.2.3 Speech Enhancement Module

The speech enhancement module, which is shown on the right-hand side of Fig. 6.2, takes the noisy speech signal $\hat{\boldsymbol{x}}$ as input and a number of generated noisy speech signals $\hat{\boldsymbol{x}}'_i$ as targets, and aims to recover the clean speech signal $\boldsymbol{x}$. Note that the output of encoder is concatenated with a random vector sampled from a normal distribution $\mathcal{N}(0, I)$. We empirically found that the random vector could help training process converge faster, see Fig 6.5 and Fig 6.4. The speech enhancement module g is a convolutional auto-encoder with objective function:

$$\arg\min_{\theta} \mathbb{E}_{\hat{\boldsymbol{x}}}\{\mathbb{E}_{\hat{\boldsymbol{x}}'_i|\hat{\boldsymbol{x}}}\{\mathcal{L}(g_\theta(\hat{\boldsymbol{x}}), \hat{\boldsymbol{x}}'_i)\}\} \tag{6.5}$$

It follows that the network parameter θ is sample-dependent. For each sample $\hat{\boldsymbol{x}}$, instead of performing a one-to-one mapping between the input and the output, (6.5) performs a one-to-many mapping. In addition, we attempt to learn the mapping from a noisy input to other noisy targets, and the model still converges. This is due to the fact that the expectation of the noisy targets is equal to the clean signal, as discussed earlier. The computation procedure is detailed in Algorithm 1, Phase II, below. Since we cannot collect an infinite number of noisy targets, it is inevitable that there is some residual noise in the enhanced speech signal. The performance can be further improved by using a Wiener Filter to remove this residual noise at the output of the auto-encoder.

6.2.4 Training

The noise conversion module is trained separately with the speech enhancement module. To train noise conversion, we pre-trained an auto-encoder first to generate the noise embedding, where both input and output of this auto-encoder is the same noise samples. After pre-training, we only use the encoder part (E_n) to generate latent noise embedding. The training pairs of noise conversion network ($< \boldsymbol{x} + \epsilon_0, \epsilon_i >$, $\boldsymbol{x} + \epsilon_i$) are fed into the network with a $\mathcal{L}_1$ loss functions.

$$\mathcal{L} = \|\mathcal{D}_n(E_c(\boldsymbol{x} + \epsilon_0), E_n(\epsilon_i)) - (\boldsymbol{x} + \epsilon_i)\|_1 \tag{6.6}$$

For specific noise, we train the separate network parameters respectively. After training of noise conversion module, the corresponding noise conversion network is selected according the noise type of input sample. The generated noisy targets then feed into the speech enhancement module. The training objective of speech enhancement is also L_1 loss as shown in Algorithm 1. The model would learn the invariant signal of noisy targets and converge after several epochs, See Fig 6.5 and 6.4 in Section 6.3.

6.3 Experiments

6.3.1 Dataset

The dataset for the experiments is generated from two sources: the DARPA TIMIT corpus [20] is used as clean speech references, whereas the collection of noise is from the 100 OSU noise corpus [33]. The TIMIT corpus includes eight major American-English dialects recorded from 630 speakers, each reading ten phonetically rich sentences, and this corpus is partitioned into test and training subsets. In this chapter, we only consider the situation where the signal-to-noise ratio (SNR) is 0 dB for speech enhancement, and we select the popular deep learning methods SEDNN [124] and SEGAN [83] as our baseline. We create training set for the deep learning models and N2N-SE's noise conversion module. To train noise conversion module, which is a one-to-many mapping function, we use the TIMIT training set (4620 sentences) and four specific noise types: the sound of an alarm, wind, car noise, and falling water. In other words, we train four noise-specific conversion modules, where each module can convert one specific noise type to the other 99 noise types from the OSU noise corpus. We use two randomly-selected noise types from the OSU corpus, and perform additive

Algorithm 1 Denoising Algorithm

Input: Noisy speech signal set $\mathcal{X}$, noise set $\mathcal{P}^{(M)}$
Output: Enhanced signal $\boldsymbol{x}$

```
Phase I: Noise Conversion Module
for each $\hat{\boldsymbol{x}} \in \mathcal{X}$ do
  initialize $\mathcal{D}_i$ as an empty set;
  for each $\epsilon_i \in \mathcal{P}^{(M)}$ do
    $\boldsymbol{c}_x \leftarrow E_c(\hat{\boldsymbol{x}})$; $\boldsymbol{c}_n \leftarrow E_n(\epsilon_i)$;
    $\boldsymbol{c} \leftarrow \{\text{concat}(\boldsymbol{c}_{\boldsymbol{x}}, \boldsymbol{c}_{\boldsymbol{n}})\}$;
    $\hat{\boldsymbol{x}}'_i \leftarrow D_n(\boldsymbol{c})$;
    add $\hat{\boldsymbol{x}}'_i$ to $\mathcal{D}_i$;
  end for
end for

Phase II: Speech Enhancement Module
initialize $\mathcal{R}$ as an empty set for results
for each $\langle \hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}'_i \rangle \in \langle \mathcal{X}, \mathcal{D}_i \rangle$ do
  while not converge do
    $\boldsymbol{c} \leftarrow E_{SE}(\hat{\boldsymbol{x}})$; $\boldsymbol{z} \leftarrow \mathcal{N}(0, I)$;
    $\boldsymbol{c} \leftarrow \{\text{concat}(\boldsymbol{c}, \boldsymbol{z})\}$;
    train and calculate loss:
    $\mathcal{L} = \|D_{SE}(\boldsymbol{c})) - \hat{\boldsymbol{x}}'_i\|_1$
    do the inference for $\boldsymbol{x}$: $\boldsymbol{x} = D_{SE}(E_{SE}(\hat{\boldsymbol{x}}))$
  end while
  add $\boldsymbol{x}$ to $\mathcal{R}$
end for
return $\mathcal{R}$
```

mixing with each sentence from the TIMIT training set for baseline systems. A total of 50 sentences from the core TIMIT test set are selected for the test set. Each sentence is combined with one of four noise types (alarm, wind, car noise, and falling water).

6.3.2 Model Setup

Baseline Setup Our baseline systems consist of a Wiener filter, SEDNN and SEGAN. For SEDNN, log-spectral features were applied for DNN-based speech enhancement spliced in time by taking a context size of seven frames, i.e., three preceding frames, the current frame, and the three next frames. The full network topology consists of three hidden layers and 2048 hidden units. The network was trained for 10,000 iterations using the Adam optimizer with a mini-batch size of 500 and 20% drop-out in the hidden layers. For SEGAN, we use the same setting as in [83], except for the batch size, which is set to 32.

Evaluations	Noise Type 1 (Alarm)		Noise Type 2 (Wind)		Noise Type 3 (Car Noise)		Noise Type 4 (Water)	
	PESQ	STOI	PESQ	STOI	PESQ	STOI	PESQ	STOI
w/o SE	1.59 ± 0.17	0.7525	1.52 ± 0.15	0.7578	1.51 ± 0.14	0.7031	1.26 ± 0.17	0.6418
Wiener Filter	2.16 ± 0.12	0.8012	1.32 ± 0.20	0.7303	1.46 ± 0.14	0.7041	1.10 ± 0.16	0.6062
SEDNN	2.46 ± 0.12	0.8342	2.05 ± 0.17	0.7835	2.29 ± 0.16	0.8338	1.89 ± 0.15	0.7255
SEGAN	2.14 ± 0.25	0.8298	1.88 ± 0.22	0.8205	2.02 ± 0.19	0.8306	1.98 ± 0.23	0.8139
N2N-SE	2.72 ± 0.18	**0.9375**	2.58 ± 0.22	**0.9014**	2.68 ± 0.21	0.9233	2.57 ± 0.21	**0.9124**
N2N-SE + WF	$\mathbf{2.97 \pm 0.18}$	0.9370	$\mathbf{2.59 \pm 0.23}$	0.8989	$\mathbf{2.69 \pm 0.21}$	**0.9236**	$\mathbf{2.69 \pm 0.23}$	0.9121

Table 6.1: Performance of the proposed models and existing DNN-based, SEGAN models and Wiener Filter methods using the evaluation metrics PESQ, STOI, and PER (the best values per column are printed in bold-face).

N2N-SE Setup The N2N-SE method directly operates on raw waveforms using a one-second sliding window, where each extract chunks of noisy speech waveforms of 16,384 samples with a 50-percent overlapped. Both noise conversion and speech enhancement modules use CNN structure. The speech enhancement is an auto-encoder, which encoder and the decoder has symmetric network configuration. The feature maps of the encoder are 16384×1, 8192×16, 4096×32, 2048×32, 1024×64, 512×64, 256×128, 128×128, 64×256, 32×256, 16×512, and 8×1024. In the speech enhancement module, we sample the noise z from our prior distribution $\mathcal{N}(0, I)$, which is 8×1024-dimensional. The noise conversion module has same structure except for an additional encoder for noise embedding. For each sentence, we use the noise conversion module to generate 99 noisy pairs. The noise embedding dimension of pre-trained encoder E_n is also set as 8×1024. The training process of noise conversion network and speech enhancement network is same, which uses RMSprop [106] optimizer and a learning rate of 0.0002, using an batch size of 8. We set a total number iterations to be 5000 for speech enhancement and 100 epochs for noise conversion module.

ASR Setup The Deep Neural Network-Hidden Markov Model (DNN-HMM) acoustic model is used to test the ASR performance of the enhanced speech signals. We first train a Gaussian Mixture Model-Hidden Markov Model (GMM-HMM) to obtain senones (tied triphone states) and the corresponding aligned frames for DNN training. The input feature vectors are used to train the GMM-HMM that contains 13-dimensional Mel-frequency spectral coefficients (MFCCs) and their first and second derivatives. Context-dependent phones, tri-phones, are modeled by 3-state HMMs. The splices of 9 frames (4 on each side of the current frame) are projected down to 40-dimensional vectors by linear discriminant analysis (LDA), together with maximum likelihood linear transform (MLLT), and then used to train the GMM-HMM using maximum likelihood estimation.

The MFCC features which is stacked over an 11-frame window are used as the input layer of the DNN. The DNN has six hidden layers, and each layer contains $1,024$ nodes. Since TIMIT is a small corpus, the DNN acoustic model was first initialized with stacked restricted Boltzmann machines (RBMs) that were pre-trained in a greedy layer-wise fashion as in [30]. After pre-training, all weights and biases were discriminatory trained by optimizing the cross entropy between the target (corresponding to context-dependent HMM states) probability and actual output of softmax output with the Back-Propagation (BP). We used the default TIMIT s5 recipe in Kaldi [85].

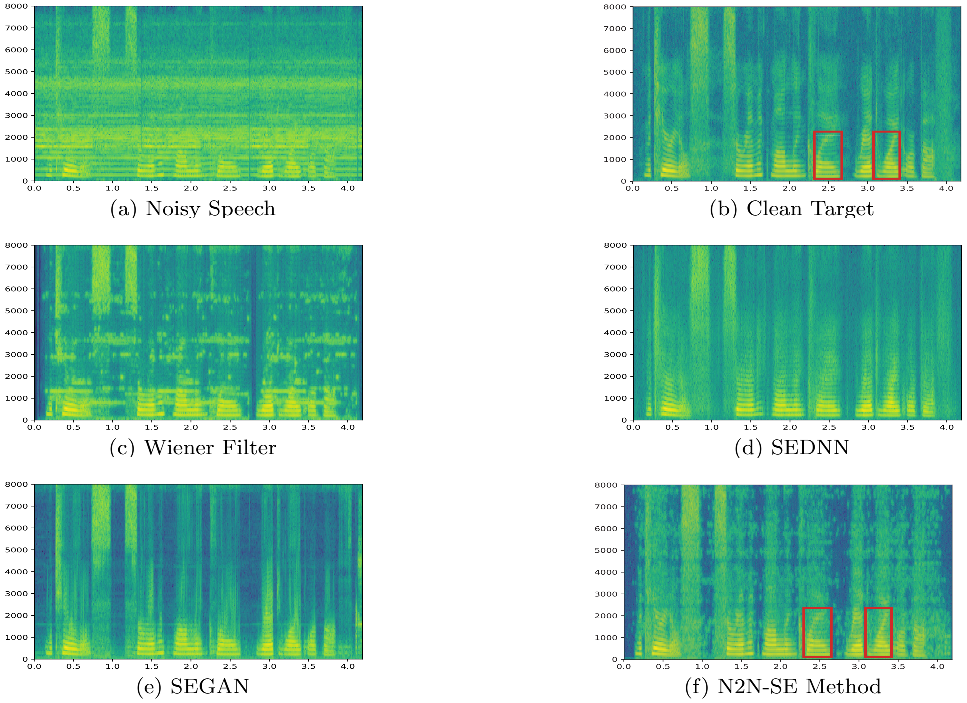
Evaluation Metrics We evaluate speech enhancement via the perceptual evaluation of speech quality (PESQ) score [39] and the Short-Time Objective Intelligibility (STOI) score [99]. The PESQ score has a high correlation with subjective evaluation scores, and is mostly used as a compressive objective measure. The PESQ score is computed by comparing the enhanced speech with the clean reference speech, and it ranges from -0.5 to 4.5. The STOI score is highly relevant to human speech intelligibility and the score ranges from 0 to 1. In order to evaluate the ASR performance, we use the phone error rate (PER) to demonstrate the effectiveness of enhanced speech signal, where we expect ASR to be robust with noisy speech signal.

6.3.3 Speech Enhancement Performance

N2N-SE vs. Baselines: As shown in the Table 6.1, the N2N-SE method outperforms all baseline in terms of perceptual quality and intelligibility with marginal gap. For example, compared with SEDNN, N2NSE significantly improves the average PESQ from 2.46 to 2.72 and average STOI from 0.8342 to 0.9375 on alarm noise condition. Also, N2N-SE clearly outperforms SEGAN with a 0.58 improvement of PESQ on alarm conditions. This demonstrates that the N2N-SE method effectively suppresses the noise signal, and it maintains the clean speech signal with only a minimal amount of distortion.

N2N-SE with Wiener Filter: The Table 6.1 shows that N2N-SE post-processed with a Wiener Filter (N2N-SE+WF) further improves the speech perceptual quality compare with pure N2N-SE. N2N-SE+WF outperforms N2NSE in most cases, indicating that the additional wiener filter can further purify the speech signal prediction. This is expected that when we do not have enough noisy targets, and the bias term in Equation 6.2 (the mean of the noise signal) would be a normal distribution, which can be effectively removed by wiener filter as discussed in Section 6.2.1.

Signal fidelity: As shown in Fig 6.3, we plot the spectrograms of a one-sentence speech



(a) Noisy Speech

(b) Clean Target

(c) Wiener Filter

(d) SEDNN

(e) SEGAN

(f) N2N-SE Method

Figure 6.3: Spectrograms of a) a sample input mixed with alarm noise, where the SNR = 0 dB, b) the clean target, c) Wiener Filter, d) SEDNN, e) SEGAN, and f) the proposed N2N-SE method.

sample, and list visualization of different methods. As shown in the plot, the proposed N2N-SE method not only effectively inhibit noise sources, but also it incurs only minimal distortions on the speech signal, which is illustrated by the red boxes. This indicates that N2N-SE can suppress noise, while preserving signal fidelity with minimal distortion.

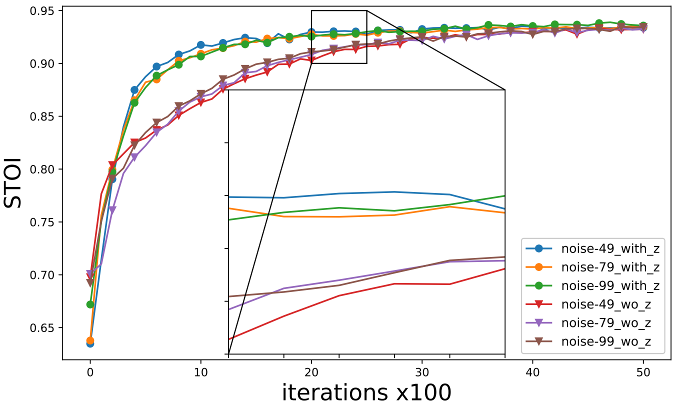


Figure 6.4: The STOI performance with and without random noise vector on different number of noisy targets (higher is better).

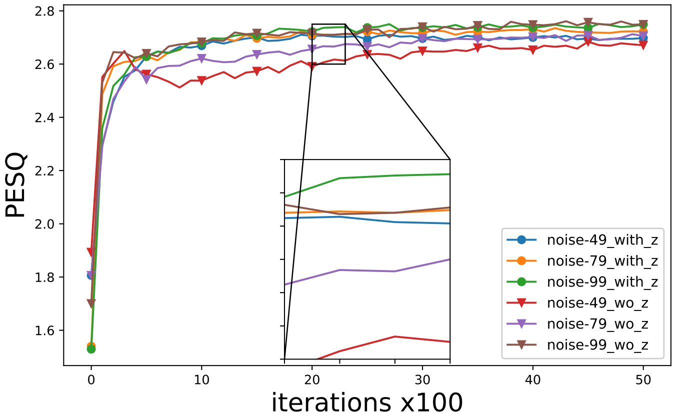


Figure 6.5: The PESQ performance with and without random noise vector on different number of noisy targets (higher is better).

Number of noise targets effectiveness: As discussed in Section 6.2, the noise term in Equation 6.2 can be suppressed when M increases. To verify the hypothesis, we adjust the number of noisy speech signals that are generated by the noise conversion module using the same input sample. As shown in Fig 6.7, both PESQ and STOI improves significantly as M increases for different noise types, which verify our assumption in Section 6.2.1.

Random noise vector effectiveness: In the speech enhancement module, we find that concatenating a random noise vector with the output of encoder would make training converge faster. As shown in Fig 6.5 and Fig 6.4, both PESQ and STOI of the model without random noise vector are slightly worse than that of the system with random noise vector. We speculate that adding these noise vector would avoid the speech enhancement network overfit to the noisy targets and learn the invariant signal as much as possible.

Convergence Speed: Similar as [56], we observe that the training loss of N2N-SE becomes diverge. Each loss value of one iteration is calculated from 16384 sample points (the input dimension). We demonstrate the training loss of two noise conditions as shown in Fig 6.6. In Fig 6.6, although the loss does not show any convergence at $3,000$ iterations, the enhancement performance has already converged to stable level, which is $3,000$ shown in Fig 6.7.

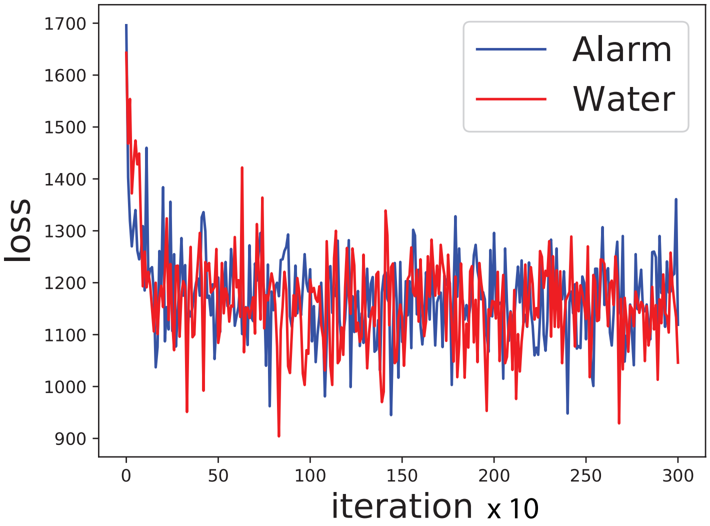


Figure 6.6: Training loss of N2N-SE would not converge

6.3.4 Noisy Speech Recognition Performance:

We further evaluate the how much ASR performance gain would be obtained by using our proposed framework. Many state-of-art speech recognition systems use multi-style training (MTR) [63] to achieve robustness to noise at inference time. In order to verify this strategy is also effective for the preprocessing stage, we examine the speech enhancement module with two scenarios: 1) the acoustic model is trained only using clean speech and 2) the acoustic model is trained by conventional MTR, where we use enhanced speech as ASR input. We use original TIMIT clean speech and mixed noisy speech for MTR. A randomly selected noise sound from OSU-100 sources was attached to every sentence of TIMIT training set with five different signal-to-noise ratios (SNRs): -5, 0, 5, 10 and 15dB.

ASR trained with clean speech: As we shown in Table 6.2, where the ASR is trained on clean speech while testing on the enhanced speech samples, the N2N-SE+WF outperform most of baselines. Specifically, the N2N-SE method achieves an absolute PER gain of 16.9 %, 5.8 % and 7.5 % when compared with the Wiener Filter, SEDNN and SEGAN on alarm noise condition, respectively. Additionally, we observe the downgraded ASR performance on car noise even though the corresponding perceptual quality and intelligibility metrics show best results. On the other hand, Table 6.2 indicate the closeness between enhanced speech and clean target: Since clean speech (training phase) and enhanced speech (prediction phase) would have mismatched distribution, the improved ASR performance shows that N2N-SE+WF output would be more close with clean speech compared with other alternatives.

ASR trained with MTR: As the ASR is trained with enhanced speech at both training and evaluation phase, the performance of acoustic model in Table 6.3 is more robust with noisy speech compared with Table 6.2. As shown in Table 6.3, N2N-SE outperform all previous baseline with large gap. However, when training ASR, we find that the extra Wiener Filter would downgrade the performance of our model. One possible reason is that the Wiener Filter would employ extra distortions on original speech signal. Another explanation is that the approximate Gaussian noise act as regularization role, such that it provides ASR better generalization.

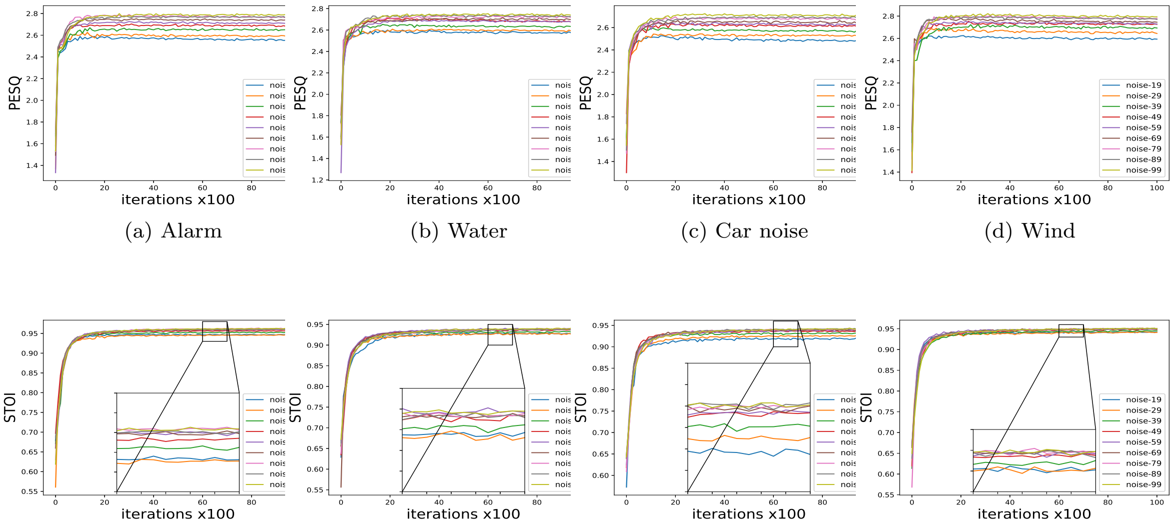
Evaluations	Alarm	Wind	Car Noise	Water
w/o SE	72.8	74.1	71.9	75.4
Wiener Filter	60.5	54.4	72	75.8
SEDNN	49.4	53.9	**42.5**	59.7
SEGAN	51.1	50.5	50.2	49.8
N2N-SE	43.6	52.4	45.2	49.4
N2N-SE + WF	**43.4**	**50.4**	44.9	**48.9**

Table 6.2: PER(%) with acoustic model using clean speech

Evaluations	alarm	wind	car noise	water
w/o SE	50.4	50.8	56.3	54
Wiener Filter	51.4	53.1	69.5	57.5
SEDNN	44.3	40.2	56.8	47.6
SEGAN	49.3	46.7	46.6	48.4
N2N-SE	**34.3**	**36.8**	**38.5**	**40.6**
N2N-SE + WF	38.4	36.9	42.7	44.7

Table 6.3: PER(%) with acoustic model using multi-style training

Number of noise targets on ASR: We also investigate the how ASR performance is influenced when the number of noisy targets varies. Based on trained ASR in both clean speech and MRT cases, we tune the number of generated noise targets from noise conversion module. As shown in Fig 6.8, we observe that PER decreases as the number of noisy target increased for both cases. Furthermore, the performance of MTR based acoustic model is significantly better than that of the acoustic model which only use clean speech.



(a) Alarm (b) Water (c) Car noise (d) Wind

(e) Alarm (f) Water (g) Car noise (h) Wind

Figure 6.7: The performance impact of using different number of noisy targets (higher is better). For each noise condition, we plot the PESQ and STOI, respectively. Note that noise-19 means that 19 noise types used in this case.

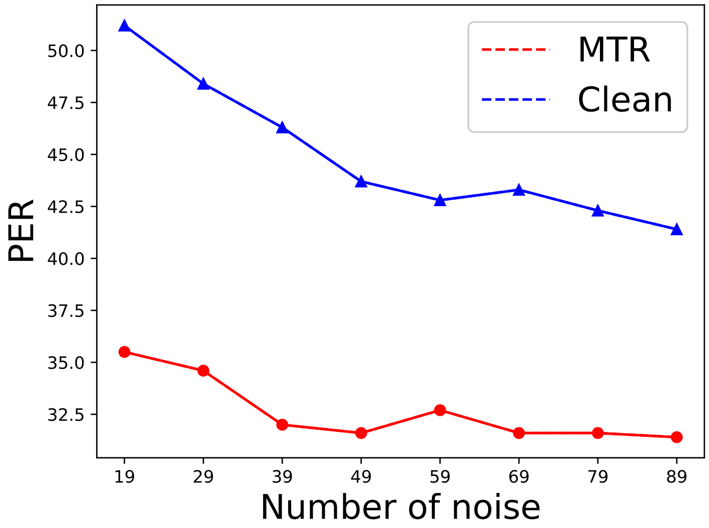


Figure 6.8: ASR performance on two different acoustic models with different number of noisy targets (lower is better).

6.4 Summary

In this chapter, we proposed a novel speech enhancement framework without the need to use clean speech. The proposed N2N-SE method comprises a noise conversion module to generate noisy targets and a speech enhancement module that reconstructs expectations of the corrupted signal. A series of experiments have been conducted for four specific noise types. The proposed N2N-SE method achieves the best performance on perceptual quality and intelligibility metrics, and also shows marginal ASR performance gains in most cases. The multi-style training ASR experiment results also show that MTR based acoustic model is also the effective for the enhanced speech.

In future work, the noise conversion module could be replaced by a GAN, such as styleGAN [41] or starGAN [7], to realize non-parallel noisy speech conversion. In addition, since the current speech enhancement parameters are sample-dependent, another potential direction is to explore a model that can infer clean signals without re-training. Furthermore, we would like to test the proposed approach on a large scale data set and a more diverse noise corpus.

Chapter 7

Speech Enhancement Using Cascaded Conformers to Suppress Noise and Reverberation

7.1 Introduction

During the last several decades, a wide variety of noisy-reverberant speech enhancement methods have been developed and refined to improve the quality and intelligibility of the degraded speech signal. In [27], a deep neural network (DNN) was used as a non-linear regression function that maps the log-magnitude spectrum of noisy-reverberant speech to that of clean anechoic speech. In [122], the complex ideal ratio mask was proposed to separate speech in reverberant and noisy environments. Normally, background noise is an additive signal to clean speech, while reverberation is a convolution of the speech signal with the room impulse response (RIR) [125]. In order to deal with the different natures of background noise and room reverberation, a step-by-step approach was proposed in [48], which first performs speech denoising by using spectral subtraction, and then removes the reverberation from the denoised reverberant signal by a multi-step linear prediction dereverberation algorithm. A DNN-based two-stage approach was proposed in [131], where denoising and dereverberation are performed in two separate stages. During the inference time, the time-domain signals are resynthesized using the Griffin-Lim phase enhancement algorithm [25]. In [89], a wide residual network (WRN) was used to leverage the residual connections in a very deep architecture. Tang et al. [104] proposed a long short-term memory (LSTM) network with progressive learning for noisy-reverberant speech enhancement. DenseUNet, proposed in [129], uses time-frequency (T-F) attention for noisy-reverberant speech enhancement in the complex domain [129]. Fan et al. [14] proposed to simultaneously denoise and dereverberate using deep embedding features, which are generated from the anechoic speech and residual reverberation signals. DESNet, proposed in [19], performs speech dereverberation, enhancement and separation simultaneously.

Recently, a conformer was introduced for speech recognition [26]. A conformer is composed of a feed-forward module, a self-attention module, a convolution module, and a second feed-forward module. It combines convolution neural networks and transformers to model both local and global dependencies of an audio sequence in a parameter-efficient way. Very recently, the conformer was adapted for speech separation [5] and speech denoising [42], and it was shown to be very effective.

In this chapter, we propose a speech enhancement method that uses cascaded conformers to denoise and dereverberate speech signals sequentially. We first investigate the conformer's effectiveness in removing noise and reverberation in a single processing stage, and then explore the use of two cascaded conformers for denoising and dereverberation, respectively. Specifically, we develop two conformer-based subsystems that are trained for denoising and dereverberation individually. Then, the two sub-systems are concatenated for testing. We also develop and explore several joint training strategies and weight initialization procedures. The effectiveness of the proposed systems is compared with LSTM-based systems using the LibriSpeech corpus in terms of the commonly used *perceptual evaluation of speech quality* (PESQ) score [37], the *short-time objective intelligibility* (STOI) score [99], and the word error rate for the automatic speech recognition (ASR) experiments.

7.2 Speech Enhancement Using Cascaded Conformers

Consider an anechoic speech signal $s(t)$ that is affected by background noise and reverberance. The noise-affected reverberant speech signal $y(t)$ is given by

$$y(t) = x(t) + v(t) = s(t)*h(t) + v(t). \tag{7.1}$$

where $v(t)$ denotes the background noise, $x(t)$ denotes the reverberant speech and $h(t)$ denotes the impulse response function that reflects the reverberation. The operator $*$ denotes the convolution operator.

The objective is to process the observed noise-affected reverberant speech signal $y(t)$ and recover the anechoic signal $s(t)$. It is natural to first remove the noise and then recover the anechoic speech. In this chapter, the estimation of the reverberant speech signal $\hat{x}(t)$ given the noise-affected reverberant speech signal $y(t)$ is referred to as the denoising operation and the estimation of the clean anechoic signal $\hat{s}(t)$ given $x(t)$ is referred to as the dereverberation operation. The estimation of $\hat{s}(t)$ given $y(t)$ is referred to as combined denoising and dereverberation. The model name with suffix "-N" denotes the denoising operation, e.g., LSTM-N, CONF-N, the suffix "-R" denotes the dereverberation operation, and the suffix "-N-R" denotes simultaneous denoising and dereverberation.

7.2.1 Conformer Architecture

A conformer, originally proposed in 2020 for ASR, increases the local information modeling capability of a traditional transformer [113] by inserting a convolution layer into the transformer block. In addition, it combines the power of relative position encoding (PE) and a half-step feed-forward Macaron net [68]. The architecture of a conformer block is shown in Fig. 7.1. It takes input $\boldsymbol{x}$ and first uses a feed-forward network (FFN), given by

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \frac{1}{2}\text{FFN}(\boldsymbol{x}). \tag{7.2}$$

Next, it uses a multi-headed self-attention (MHSA) module μ to obtain

$$\boldsymbol{x}' = \tilde{\boldsymbol{x}} + \mu(\tilde{\boldsymbol{x}}). \tag{7.3}$$

The MHSA module used in the conformer uses a variant of self-attention, which allows the model to learn from different representation sub-spaces. Self-attention is an efficient context information aggregation mechanism, which can be formulated as querying a dictionary with key-value pairs [113]. The self-attention function $\alpha(\boldsymbol{K}, \boldsymbol{V}, \boldsymbol{Q})$ is defined as

$$\alpha(\boldsymbol{K}, \boldsymbol{V}, \boldsymbol{Q}) = \sigma\left(\frac{\mathbf{QK}^\mathsf{T}}{\sqrt{d}}\right) \cdot \boldsymbol{V}, \tag{7.4}$$

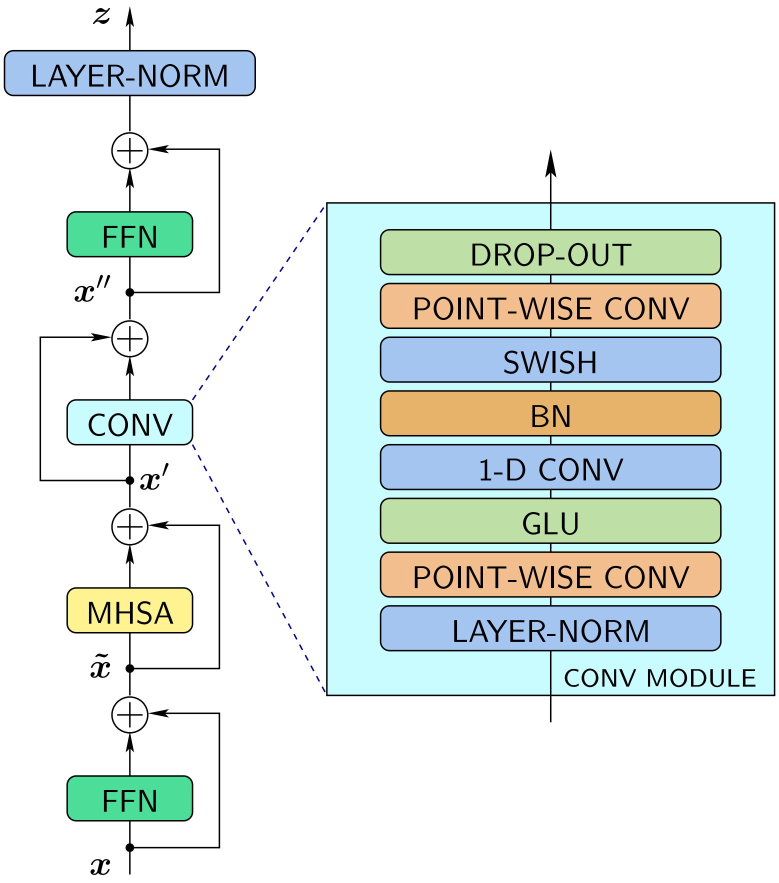


Figure 7.1: Conformer block architecture.

where $\sigma(\cdot)$ denotes the soft-max function, $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are hidden representations of the previous layer of dimension d, which are referred to as the query, and the key-value pair, respectively.

The MHSA used in the conformer is defined by

$$\mu(\boldsymbol{K},\boldsymbol{V},\boldsymbol{Q}) = [\mathbf{H}_1,\mathbf{H}_2,\cdots,\mathbf{H}_n]\boldsymbol{W}^h \tag{7.5}$$

where $\boldsymbol{W}^h \in \mathbb{R}^{d\times d}$ is an output linear projecting matrix, and

$$\mathbf{H}_i = \alpha(\boldsymbol{K_i},\boldsymbol{V_i},\boldsymbol{Q_i}) \tag{7.6}$$

The conformer also integrates a position encoding scheme from the TransformerXL [11] to generate better position information for the input sequence with various lengths, named relative positional encodings.

The output $\boldsymbol{x}'$ of the MHSA module is fed into a convolution (CONV) module, i.e.,

$$\boldsymbol{x}'' = \boldsymbol{x}' + \mathrm{CONV}(\boldsymbol{x}'). \tag{7.7}$$

Lastly, a second feed-forward network is used to obtain conformer output $\boldsymbol{z}$, given by

$$\boldsymbol{z} = \mathrm{LayerNorm}(\boldsymbol{x}'' + \mathrm{FFN}(\boldsymbol{x}'')) \tag{7.8}$$

7.2.2 Cascaded Conformers

The proposed speech enhancement system consists of two cascaded conformers that are used for speech denoising and speech dereverberation, respectively. We refer to the proposed approach as a two-stage approach. A block diagram of the proposed system is shown in Fig. 7.2. Each conformer stacks several conformer blocks. The proposed system takes noisy reverberant speech signal as input and uses its magnitude. Previous studies have shown that methods that are based on masked-learning for speech denoising usually perform better than feature mapping methods in terms of speech quality metrics [120, 75]. Guided by this observation, the first conformer is used to predict the ratio mask using a sigmoid function and as such attenuate background noise, and the second conformer performs spectral mapping for speech dereverberation. Ratio masking is a good choice for speech denoising as speech and noise are uncorrelated, whereas spectal mapping is effective in dealing with artifacts introduced by ratio masking and in suppressing dereverberation [116, 131, 103]. The estimated magnitude and the phase of the noisy reverberant signal are fed into an inverse STFT to reconstruct the anechoic speech.

7.2.3 Training

For notational convenience, let $\boldsymbol{\Phi}(\cdot)$ denote the conformer for speech denoising and let $\boldsymbol{\Psi}(\cdot)$ denote the conformer for speech dereverberation. The cascaded conformer takes the magnitude $\boldsymbol{Y}$ of the noisy-reverberant speech as input. Let $\hat{\boldsymbol{Y_1}}$ denotes the intermediate output of $\boldsymbol{\Phi}(\cdot)$ and $\hat{\boldsymbol{Y_2}}$ denotes the output of $\boldsymbol{\Psi}(\cdot)$. The training objective function of the proposed approach can be defined

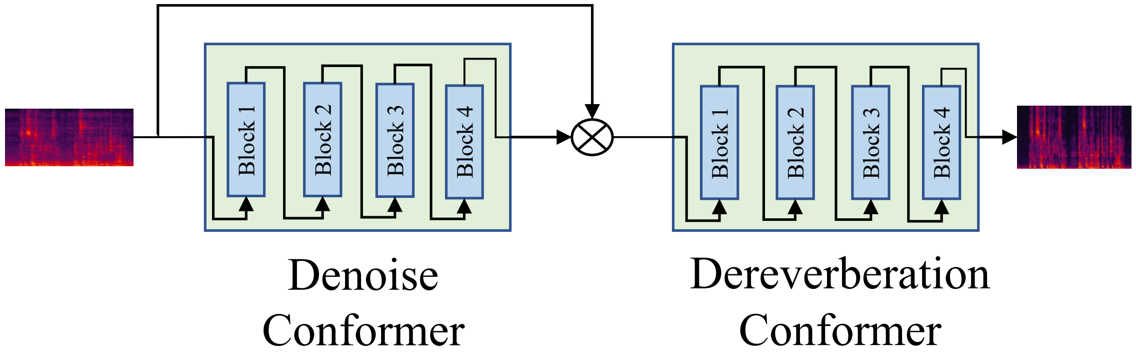


Figure 7.2: Proposed cascaded conformers for noisy-reverberant speech enhancement.

as

$$\begin{aligned}\mathcal{L} &= \|\boldsymbol{X} - \hat{\boldsymbol{Y}}_1\|_1 + \|\boldsymbol{S} - \hat{\boldsymbol{Y}}_2\|_1 \\ &= \|\boldsymbol{X} - \boldsymbol{\Phi}(Y)\|_1 + \|\boldsymbol{S} - \boldsymbol{\Psi}(\boldsymbol{\Phi}(Y))\|_1 \end{aligned} \tag{7.9}$$

where $\boldsymbol{S}$ and $\boldsymbol{X}$ are the magnitudes of the anechoic speech and reverberant speech, respectively. The use of the mean absolute error loss is motivated by recent observations that it achieves better objective quality scores when using spectral mapping techniques [79, 80].

We investigate four training strategies for the proposed two-stage approach: *i)* The denoising and dereverberation conformers are trained individually and then concatenated in the test stage (CONF-N – CONF-R (fixed-wgt)). *ii)* The cascaded conformer is trained jointly from scratch (CONF-N – CONF-R (scratch)). *iii)* The denoising conformer is loaded from the pre-trained model and then jointly trained with the dereverberation conformer (CONF-N – CONF-R (N-tuned)). *iv)* Both the denoising and the dereverberation conformer are fine-tuned using pre-trained models (CONF-N – CONF-R (N-R-tuned)).

7.3 Experiments

In the following, the data set, the model set up and the ASR set up are detailed.

Data Set. In the experiments, the clean source is obtained from LibriSpeech [78], and for training 100 hours of speech data from the "train-clean" data set is selected. The RIRs are obtained from [50] and include both recorded and synthesized RIRs for small, medium and large rooms. The training clean speech is cut into 4-second segments. The validation set uses 1400 sentences from the "dev-clean" data set, and the test set uses 500 sentences from the "test-clean" data set. The clean speech is convolved with recorded and synthesized RIRs to obtain the reverberant utterances. We formed two testing sets: one is the *recorded* set that contains 500 reverberant speech utterances convolved with recorded RIRs and the other test set, referred to as *synthesized*, contains 500 reverberant speech utterances convolved with synthesized RIRs. The RIRs used for training and testing do not overlap. The reverberant speech is generated by convolving the clean signal with a given RIR using Kaldi [85].

Next, we generate the noisy-reverberant speech. The training set uses 10,000 randomly selected noise sample sequences from the DNS Challenge [87]. The training and validation sets distort the reverberant clean segments with randomly-selected noise from the DNS Challenge noise set with an SNR in the set $\{-5, -4, \cdots, 9, 10\}$ (in dB). The test set uses three distinct noise types: "babble noise" from the NOISEX-92 corpus [112], and "office noise" and "kitchen noise" from the DEMAND noise corpus [105]. The first channel signal of the corpus is used for data generation. Each clean utterance is distorted by a randomly selected noise type at a randomly selected SNR from the set $\{-5, 0, 5, 10\}$ (in dB).

Model Setup. The proposed cascaded conformer-based systems are compared with LSTM-based baseline systems. All models take the magnitude as input. An STFT is used with a 32 ms Hanning window, a 32 ms filter length and a 16 ms hop size. As such, the number of frequency bins equals 257. The LSTM baseline model consists of three LSTM layers followed by a fully-connected output layer. Each LSTM layer has 1,024 memory cells. The output layer reduces the dimensionality from 1,024 to 257. The total number of parameters of the baseline system is 22.31 M.

Each conformer model consists of four conformer blocks with four attention heads and 128 attention dimensions, which amounts to 10.71 M parameters. The single-stage conformer used to predict the ratio mask employ a sigmoid function to simultaneously suppress the noise and the reverberation. The two-stage conformer-based systems use a denoising conformer to predict the mask, and its output is then multiplied with the noisy spectra and fed into the dereverberation conformer.

All models are trained using the Adam optimizer [47] with an initial learning rate of 0.0002 and use a mini-batch of 16 utterances.

ASR Setup. The automatic speech recognition (ASR) experiments use a time-delay neural network-hidden Markov model (TDNN-HMM) hybrid chain model [84] to account for long-term temporal dependencies with training times that are comparable to standard feed-forward DNNs. The data is represented at different time points by adding a set of delays to the input, which allows the model to have a finite dynamic response to the time series input data. This acoustic model is trained using the Kaldi toolkit [85] with the standard recipe (`https://github.com/kaldi-asr/kaldi/tree/master/egs/librispeech/s5`). The ASR acoustic models were trained using 960 hours from the LibriSpeech training set.

7.4 Results

The speech enhancement systems are evaluated using the commonly used *perceptual evaluation of speech quality* (PESQ) score [37] and the *short-time objective intelligibility* (STOI) score [99]. The automatic speech recognition performance is measured by determining the word error rate (WER). We first compare the conformer with an LSTM baseline system for speech denoising and dereverberation, respectively. The resulting PESQ and STOI scores are listed in Table 7.1. We can see that the conformer-based systems outperform LSTM systems significantly. Using recorded RIR signals, the conformer-based system achieves denoising with a STOI score improvement of 2.47 % and a PESQ score improvement of 0.18 relative to the LSTM system. For dereverberation, the average PESQ score improves from 2.34 to 2.44 and the average STOI score improves from 89.46 % to 91.01 %. Similar results are observed when using synthesized RIR signals, which demonstrates the effectiveness of the conformer-based system.

Next, we investigate the single stage for speech denoising and dereverberation simultaneously. As shown in Table 7.1, the conformer-based system is also better than the LSTM-based system. For example, when using recorded RIR signals, the average PESQ and STOI improve from 1.60 to 1.71 and from 78.25 % to 81.55 %. Both the LSTM and the two-stage conformer-based systems that use fixed weights boost the performance. For instance, the average PESQ scores of the LSTM-based system improves from 1.60 to 1.66, and for the proposed conformer-based system, it improves from 1.71 to 1.87. This observation is consistent with observations for the two-stage approach in [131]. This indicates the effectiveness of decomposing the original difficult task into multiple “easier” sub-tasks.

We further investigate the performance of the conformer-based two-stage system using the three fine-tuned training strategies presented earlier. We observe that all conformer-based two-stage approaches achieve similar PESQ scores, except for the CONF-N – CONF-R (scratch) approach, which is slightly worse. This demonstrates that proper weight initialization is also useful for noisy-reverberant speech enhancement. We also observe that fine-tuning the pre-trained models achieves better STOI scores than fixed pre-trained model weight approaches. The best performance is achieved by CONF-N – CONF-R (N-R-tuned) where both the denoising and the dereverberation conformer are fine-tuned using pre-trained models. The PESQ and STOI scores for each considered SNR condition are provided in Table 7.2.

System	recorded RIRs		synthesized RIRs	
	PESQ	STOI	PESQ	STOI
Noisy-reverberant speech	1.61	81.97	1.54	81.24
LSTM-N	2.47	85.07	2.44	84.58
CONF-N	2.65	87.54	2.60	87.40
Reverberant speech	1.84	83.09	1.83	86.09
LSTM-R	2.34	89.46	2.45	91.26
CONF-R	2.44	91.01	2.55	92.31
Noisy-reverberant speech	1.20	72.77	1.21	74.97
LSTM-N-R	1.60	78.25	1.64	80.41
CONF-N-R	1.71	81.55	1.72	82.60
Fixed Weights				
LSTM-N – LSTM-R (fixed-wgt)	1.66	78.52	1.71	80.62
CONF-N – CONF-R (fixed-wgt)	1.87	82.79	**1.91**	84.42
Fine-tuned weights				
CONF-N – CONF-R (scratch)	1.82	82.51	1.86	84.02
CONF-N – CONF-R (N-tuned)	1.87	82.99	1.90	84.56
CONF-N – CONF-R (N-R-tuned)	**1.88**	**83.07**	**1.91**	**84.95**

Table 7.1: Performance in terms of PESQ and STOI scores for the proposed and baseline systems.

The LibriSpeech corpus is also used to assess the ASR performance of the proposed conformer-based and baseline systems in terms of the word error rate (WER) and the WER reduction relative to unprocessed speech (R_{WER}). The results are shown in Table 7.3. Both the LSTM-based and the conformer-based systems improve the ASR performance when compared with the unprocessed speech. Both single-stage and two-stage conformer-based systems perform better than LSTM-based systems. We observe that CONF-N – CONF-R (N-R-tuned) and CONF-N – CONF-R (N-tuned) outperform CONF-N – CONF-R (fixed-wgt). This also supports the observation that the ASR results are related to the STOI scores. As mentioned in the ASR setup, we use the clean speech for acoustic model training. There is a significant mismatch between training and testing, which causes the overall WER to be very high. We believe that re-training the ASR with corrupted and enhanced

recorded RIRs								
System	PESQ				STOI			
	-5 dB	0 dB	5 dB	10 dB	-5 dB	0 dB	5 dB	10 dB
Noisy-reverberant speech	1.09	1.17	1.18	1.36	64.03	73.68	73.49	80.64
LSTM-N – LSTM-R (fixed-wgt)	1.41	1.66	1.68	1.93	68.21	79.51	81.02	86.30
CONF-N-R	1.41	1.68	1.74	2.04	71.85	82.55	83.98	88.70
CONF-N – CONF-R (fixed-wgt)	1.55	1.87	1.90	2.19	72.81	84.05	85.56	89.67
CONF-N – CONF-R (N-R-tuned)	1.57	1.88	1.92	2.19	73.53	84.29	85.67	89.68
synthesized RIRs								
System	PESQ				STOI			
	-5 dB	0 dB	5 dB	10 dB	-5 dB	0 dB	5 dB	10 dB
Noisy-reverberant speech	1.10	1.15	1.23	1.36	65.60	74.86	76.81	81.93
LSTM-N – LSTM-R (fixed-wgt)	1.38	1.66	1.76	2.01	68.99	81.05	83.71	87.75
CONF-N-R	1.38	1.69	1.82	1.98	72.30	83.16	85.56	88.48
CONF-N – CONF-R (fixed-wgt)	1.51	1.85	2.00	2.24	74.11	84.84	87.36	90.46
CONF-N – CONF-R (N-R-tuned)	1.51	1.86	2.00	2.24	74.71	85.15	87.44	90.43

Table 7.2: Performance in terms of PESQ and STOI scores for several SNRs.

speech will lead to further performance improvements.

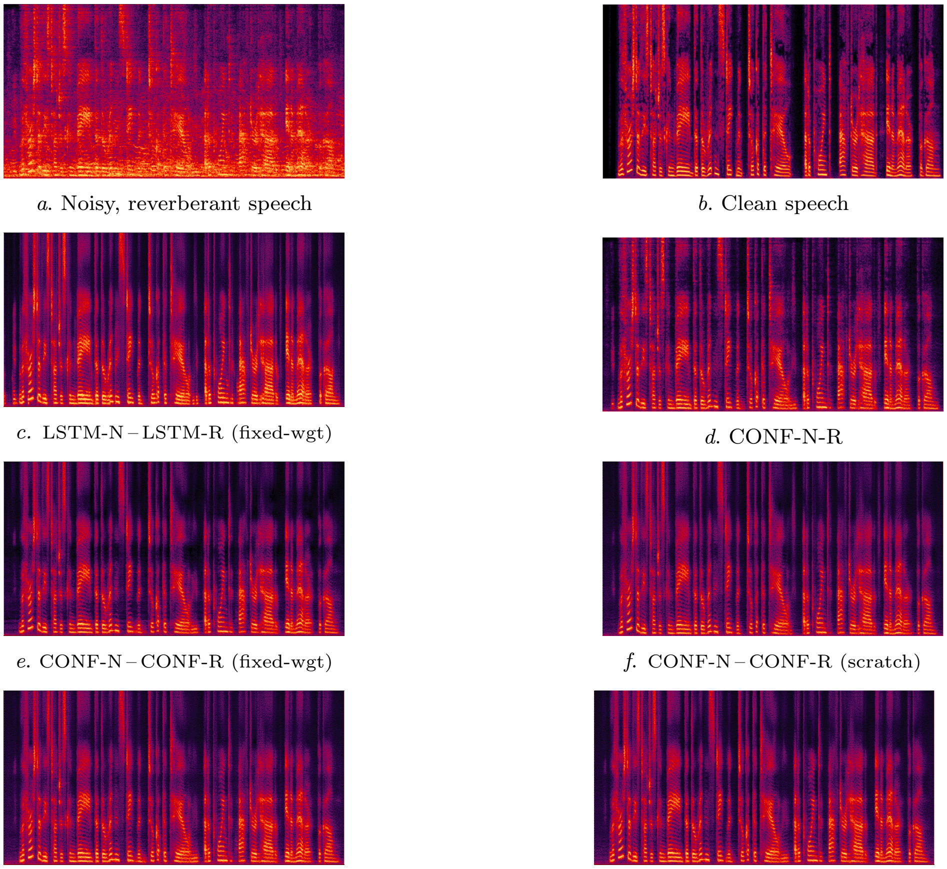
System	recorded		synthesized	
	WER [%]	R_{WER} [%]	WER [%]	R_{WER} [%]
Noisy-reverberant speech	63.01	–	55.25	–
LSTM-N-R	56.37	10.5	50.00	9.5
CONF-N-R	46.89	25.6	41.08	25.6
LSTM-N – LSTM-R (fixed-wgt)	54.10	14.1	50.11	9.3
CONF-N – CONF-R (fixed-wgt)	45.85	27.2	41.15	25.5
CONF-N – CONF-R (scratch)	46.03	26.9	41.61	24.7
CONF-N – CONF-R (N-tuned)	45.77	27.4	40.53	26.6
CONF-N – CONF-R (N-R-tuned)	**45.71**	**27.5**	**40.26**	**27.1**

Table 7.3: Speech recognition performance of the proposed and baseline speech enhancement systems

Finally, we visualize some examples of the enhanced speech. Consider the situation where clean speech is perturbed a recorded RIR and by Babble noise at an SNR of 5 dB. Fig. 7.3 shows spectrograms of the noisy speech signal, the clean speech target, the LSTM-based system and the proposed conformer-based enhanced speech system. Listening samples can be found in `https://linjucs.github.io/demo/nrdemo.html`.

7.5 Summary

In this chapter, we have introduced a novel speech enhancement method that uses two cascaded conformers that sequentially remove background noise and reverberation. Multiple training strategies were developed and their effectiveness was assessed. Experiments show that the pro-



a. Noisy, reverberant speech

b. Clean speech

c. LSTM-N – LSTM-R (fixed-wgt)

d. CONF-N-R

e. CONF-N – CONF-R (fixed-wgt)

f. CONF-N – CONF-R (scratch)

g. CONF-N – CONF-R (N-tuned)

h. CONF-N – CONF-R (N-R-tuned)

Figure 7.3: Spectrograms of a sample input mixed with Babble Noise at an SNR of 5 dB and recorded RIR, and of the signal output of the proposed conformer-based and baseline speech-enhancement systems.

posed conformer-based speech enhancement system outperforms single-stage methods in terms of commonly used speech enhancement metrics. It was shown that the pre-trained models can be fine-tuned to further boost performance, in particular the STOI score, which is an indicator for ASR performance. Future work will extend the proposed monaural algorithm to multi-channel scenarios. The performance of speech dereverberation is expected to be further improved by using spatial information.

Chapter 8

Conclusions and Future work

8.1 Conclusions

Room reverberation and background noise are two common distortions to speech signal in daily life. These distortions impact the effective communication among people and between human and machines. In this dissertation, we aim to develop monaural speech enhancement systems to improve speech intelligibility and quality of a speech signal that is degraded by these distortions. We have proposed several novel architectures for speech enhancement and the effectiveness of the proposed frameworks and techniques is verified by speech enhancement metrics and speech recognition scores. The contributions can be summarized as follows:

Chapter 2 presented a novel GAN based speech enhancement approach, named ForkGAN, and its variants. ForkGAN framework uses two decoders to extract the clean speech signal and the noise signal. Subsequent experiments demonstrate the effectiveness of the ForkGAN architecture. To further decouple the speech and the noise signals, two auxiliary training objectives are proposed based on the ForkGAN framework: margin-based loss, which explicitly maximizes the distance between clean speech and the noise signal, and a time-domain noise reduction loss which improves the prediction based upon the estimated noise signal. Both loss functions further boost the speech enhancement performance when compared with DNN-based and SEGAN methods. ForkGAN and its enhancements achieve consistent gains for many different noise conditions. We further extend ForkGAN to M-ForkGAN, which integrates mask-learning and feature-mapping. Experimental results with the TIMIT data set show that the proposed approach achieves a better performance for seen and unseen conditions with varying SNR when compared with the baseline systems. We also verified the effectiveness of the proposed mask-based loss. Finally, we adapted ForkGAN to S-ForkGAN, which directly operates on the acoustic features that use for speech recognition. The experiments show that the proposed S-ForkGAN method outperforms well-known GAN-based speech enhancement techniques, including GAN-DNN, GAN-LSTM and GAN-AE (SEGAN).

Chapter 3 demonstrated a novel multi-stage speech enhancement technique where each stage consists of a self-attention block followed by multiple stacks of temporal convolutional network blocks, and where a fusion block is inserted in systems with more than two stages. We show that the multi-stage structure serially refines the predictions. We show that the self-attention module produces dynamic representations effectively and that it is effective in mitigating non-stationary noise conditions. We show that the fusion block mitigates any speech loss in later stages. Extensive experiments have been conducted, which show that the proposed multi-stage speech enhancement system performs well, in particular with respect to the PESQ and STOI scores, and that it also performs well when used in speech recognition experiments, as it suppresses the word error rate.

Chapter 4 investigated the multi-scale temporal convolutional network (TCN) for speech denoising. We propose three multi-scale architectures to improve the speech enhancement performance, which consists of 1) TCN-dual, which has two different dilation factors in one dilated convolutional block; 2) TCN-flatten, which uses a fixed number of dilation factors in each dilated convolutional block and the output of each dilation factor are concatenated; 3) TCN-pyramid, which is similar to TCN-flatten but using a hierarchical feature fusion mechanism. In addition, we also explored several loss function for our proposed framework. The experiment results show that the proposed multi-scale TCN outperforms the villian TCN and combining the spectral domain loss and perceptual loss can further improve the performance. The proposed framework also achieved state-of-the-art performance on the public INTERSPEECH2020 DNS challenge dataset.

Chapter 5 presented and validated a novel speaker-aware speech enhancement method that uses a few seconds of reference clean speech. We first compared the proposed SASE with baseline systems using spectral mapping-MSE and mask-based signal approximation-MSE loss, respectively. The experimental results indicate that the proposed SASE system outperforms the baseline systems using both loss functions. The results also show that using mask-based signal approximation loss is better than spectral mapping-MSE loss. Adding self-attention achieves the best performance in terms of most metrics, especially for SI-SDR metric. We tested the proposed SASE approach with four different reference-speech durations. All achieved better performance in comparison with the CRN baseline, which demonstrates the effectiveness of the proposed method.

In Chapter 6, we proposed a novel speech enhancement framework without the need to use clean speech. The proposed N2N-SE method comprises a noise conversion module to generate noisy targets and a speech enhancement module that reconstructs expectations of the corrupted signal. A series of experiments have been conducted for four specific noise types. The proposed N2N-SE method achieves the best performance on perceptual quality and intelligibility metrics, and also show marginal ASR performance gains in most cases. The multi-style training ASR experiment results also show that MTR based acoustic model is also the effective for the enhanced speech.

Chapter 7 introduced a novel speech enhancement method that uses two cascaded conformers that sequentially remove background noise and reverberation. Multiple training strategies were developed and their effectiveness was assessed. Experiments show that the proposed conformer-based speech enhancement system outperforms single-stage methods in terms of commonly used speech enhancement metrics. It was shown that the pre-trained models can be fine-tuned to further boost performance, in particular the STOI score, which is an indicator for ASR performance.

8.2 Future work

In this dissertation, we have developed several monaural speech enhancement systems and the experimental results show that the proposed approaches can improve the speech perceptual quality and intelligibility, and also speech recognition performance. In order to apply the proposed enhancement systems to real-world applications, more studies need to be conducted in the following aspects.

1. **Multi-channel Speech Enhancement:** In this dissertation, we mainly focus on single-channel speech enhancement algorithms. However, microphone arrays are widely used in many modern speech processing systems, including smart phones, personal assistant, and other smart devices. With multiple microphones, spatial information can be exploited to complement spectral information for better de-noising and dereverberation. Thus, how to incorporate this information into the proposed systems could be an interesting research direction.

2. **Real-time Speech Enhancement:** Real-time speech enhancement has been an increasingly

important application in modern devices. All the models proposed in this dissertation are offline systems except for multi-scale TCN based methods, which does not consider causal setting and latency. To meet real-time applications, we need to adapt the proposed systems to causal systems and reduce the processing latency for inference, while keeping the performance to a high level. This could be an useful direction to explore.

3. **Complex-domain Feature based Speech Enhancement:** All speech enhancement approaches developed in this dissertation use real-valued DNNs and real-valued input features. However, recent studies [34, 102] show that using complex-valued DNNs or complex-domain input for complex spectral mapping achieved excellent speech enhancement performance. It would be interesting to explore more complex-domain based speech enhancement frameworks.

4. **Jointly training with ASR:** Speech enhancement is usually as a pre-processing module of the speech recognition systems. However, enhanced speech produced by neural network-based systems inevitably contains distortions, which could impact the performance for downstream task, e.g., speech recognition. Hence speech enhancement jointly training with ASR task could be an interesting research direction.

Bibliography

[1] Nasim Alamdari, Arian Azarang, and Nasser Kehtarnavaz. Self-supervised deep learning-based speech denoising. *arXiv:1904.12069*, April 2019.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[3] Deepak Baby and Sarah Verhulst. SERGAN: Speech enhancement using relativistic generative adversarial networks with gradient penalty. In *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Proc.*, pages 106–110, Brighton, United Kingdom, May 2019.

[4] Steven Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Trans. on Acoustics, Speech, and Signal Proc.*, 27(2):113–120, April 1979.

[5] Sanyuan Chen, Yu Wu, Zhuo Chen, Jian Wu, Jinyu Li, Takuya Yoshioka, Chengyi Wang, Shujie Liu, and Ming Zhou. Continuous speech separation with conformer. *arXiv:2008.05773*, August 2020.

[6] Zhou Chen, Yan Huang, Jinyu Li, and Yifan Gong. Improving mask learning based speech enhancement system with restoration layers and residual connection. In *Proc. Interspeech*, pages 3632–3636, Stockholm, Sweden, August 2017.

[7] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 8789–8797, Salt Lake City, UT, June 2018.

[8] Fu-Kai Chuang, Syu-Siang Wang, Jeih weih Hung, Yu Tsao, and Shih-Hau Fang. Speaker-aware deep denoising autoencoder with embedded speaker identity for speech enhancement. In *Proc. Interspeech*, pages 3173–3177, Graz, Austria, September 2019.

[9] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. VoxCeleb2: Deep speaker recognition. In *Proc. Interspeech*, pages 1086–1096, Hyderabad, India, September 2018.

[10] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). In *Proc. Int'l Conf. on Learning Representations*, pages 1–14, San Juan, Puerto Rico, May 2016.

[11] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. *arXiv:1901.02860*, January 2019.

[12] Alexandre Défossez, Gabriel Synnaeve, and Yossi Adi. Real time speech enhancement in the waveform domain. *arXiv:2006.12847*, September 2020.

[13] Chris Donahue, Bo Li, and Rohit Prabhavalkar. Exploring speech enhancement with generative adversarial networks for robust speech recognition. In *Proc. IEEE Int'l Conf. on Acoustics, Speech and Signal Proc.*, pages 5024–5028, Calgary, AB, April 2018. IEEE.

[14] Cunhang Fan, Jianhua Tao, Bin Liu, Jiangyan Yi, and Zhengqi Wen. Simultaneous denoising and dereverberation using deep embedding features. *arXiv:2004.02420*, April 2020.

[15] Cunhang Fan, Jianhua Tao, Bin Liu, Jiangyan Yi, Zhengqi Wen, and Xuefei Liu. Deep attention fusion feature for speech separation with end-to-end post-filter method. *arXiv:2003.07544*, March 2020.

[16] Yazan Abu Farha and Jürgen Gall. MS-TCN: Multi-stage temporal convolutional network for action segmentation. In *Proc. IEEE/CVF Conf. Comput. Vision and Pattern Recognition*, pages 3575–3584, Long Beach, CA, June 2019.

[17] Szu-Wei Fu, Ting-Yao Hu, Yu Tsao, and Xugang Lu. Complex spectrogram enhancement by convolutional neural network with multi-metrics learning. In *Proc. IEEE Int'l Workshop on Machine Learning for Signal Proc.*, pages 1–6, Tokyo, Japan, September 2017.

[18] Szu-Wei Fu, Chien-Feng Liao, Yu Tsao, and Shou-De Lin. MetricGAN: Generative adversarial networks based black-box metric scores optimization for speech enhancement. In *Proc. Int'l Conf. on Machine Learning*, pages 2031–2041, Long Beach, CA, June 2019.

[19] Yihui Fu, Jian Wu, Yanxin Hu, Mengtao Xing, and Lei Xie. DESNet: A multi-channel network for simultaneous speech dereverberation, enhancement and separation. *arXiv:2011.02131*, November 2020.

[20] John S. Garofolo. TIMIT acoustic phonetic continuous speech corpus. Technical report, Linguistic Data Consortium, 1993.

[21] Francois G. Germain, Qifeng Chen, and Vladlen Koltun. Speech denoising with deep feature losses. *arXiv:1806.10522*, June 2018.

[22] Ritwik Giri, Umut Isik, and Arvindh Krishnaswamy. Attention Wave-U-Net for speech enhancement. In *Proc. IEEE Workshop on Appl. of Signal Proc. to Audio and Acoustics*, pages 249–253, New Paltz, NY, October 2019.

[23] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, Cambridge, MA, 2017.

[24] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial sets. In *Proc. Conf. on Neural Inform. Proc. Sys.*, Montréal, QC, December 2014.

[25] Daniel Griffin and Jae Lim. Signal estimation from modified short-time Fourier transform. *Proc. IEEE/ACM Trans. on Audio, Speech and Language Proc.*, 32(2):236–243, April 1984.

[26] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.

[27] Kun Han, Yuxuan Wang, DeLiang Wang, William S Woods, Ivo Merks, and Tao Zhang. Learning spectral mapping for speech dereverberation and denoising. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(6):982–992, 2015.

[28] Xiang Hao, Xiangdong Su, Shixue Wen, Zhiyu Wang, Yiqian Pan, Feilong Bao, and Wei Chen. Masking and inpainting: A two-stage speech enhancement approach for low SNR and non-stationary noise. In *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Proc.*, pages 6959–6963, Barcelona, Spain, May 2020.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proc. IEEE Int'l Conf. Comput. Vision*, pages 1026–1034, Santiago, Chile, December 2015.

[30] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[31] Jen-Cheng Hou, Syu-Siang Wang, Ying-Hui Lai, Yu Tsao, Hsiu-Wen Chang, and Hsin-Min Wang. Audio-visual speech enhancement using multimodal deep convolutional neural networks. *IEEE Trans. Topics Comput. Intell.*, 2(2):117–128, April 2018.

[32] Tsun-An Hsieh, Cheng Yu, Szu-Wei Fu, Xugang Lu, and Yu Tsao. Improving perceptual quality by phone-fortified perceptual loss using Wasserstein distance for speech enhancement. In *Proc. Interspeech*, pages 196–200, Brno, Czech Republic, August/September 2021.

[33] Guoning Hu and DeLiang Wang. A tandem algorithm for pitch estimation and voiced speech segregation. *IEEE Trans. on Audio, Speech, and Language Proc.*, 18(8):2067–2079, 2010.

[34] Yanxin Hu, Yun Liu, Shubo Lv, Mengtao Xing, Shimin Zhang, Yihui Fu, Jian Wu, Bihong Zhang, and Lei Xie. DCCRN: Deep complex convolution recurrent network for phase-aware speech enhancement. Pre-Print 2008.00264, ArXiV, August 2020.

[35] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. Int'l Conf. on Machine Learning*, pages 448–456, Lille, France, July 2015.

[36] Umut Isik, Ritwik Giri, Neerad Phansalkar, Jean-Marc Valin, Karim Helwani, and Arvindh Krishnaswamy. Poconet: Better speech enhancement with frequency-positional embeddings, semi-supervised conversational data, and biased loss. *arXiv preprint arXiv:2008.04470*, 2020.

[37] Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs, February 2001.

[38] Perceptual objective listening quality prediction, March 2018.

[39] Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs, Feb. 2001.

[40] Xuan Ji, Meng Yu, Chunlei Zhang, Dan Su, Tao Yu, Xiaoyu Liu, and Dong Yu. Speaker-aware target speaker enhancement by jointly learning with speaker embedding extraction. In *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Proc.*, pages 7294–7298, Barcelona, Spain, May 2020.

[41] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 4401–4410, Long Beach, CA, June 2019.

[42] Saurabh Kataria, Jesús Villalba, and Najim Dehak. Perceptual loss based speech denoising with an ensemble of audio pattern recognition and self-supervised models. *arXiv:2010.11860*, October 2020.

[43] Saurabh Kataria, Jesús Villalba, and Najim Dehak. Perceptual loss based speech denoising with an ensemble of audio pattern recognition and self-supervised models. In *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Proc.*, pages 7118–7122, Toronto, ON, June 2021.

[44] Jaeyoung Kim, Mostafa El-Khamy, and Jungwon Lee. T-GSA: Transformer with Gaussian-weighted self-attention for speech enhancement. In *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Proc.*, pages 6649–6653, Barcelona, Spain, May 2020.

[45] Anurag Kimar and Dinei Florencio. Speech enhancement in multiple-noise conditions using deep neural networks, 2016.

[46] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[47] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *Proc. Int'l Conf. on Learning Representations*, pages 1–15, San Diego, CA, May 2015.

[48] Keisuke Kinoshita, Marc Delcroix, Tomohiro Nakatani, and Masato Miyoshi. Multi-step linear prediction based speech dereverberation in noisy reverberant environment. In *Proc. Interspeech*, pages 854–857, Antwerp, Belgium, August 2007.

[49] Vinith Kishore, Nitya Tiwari, and Periyasamy Paramasivam. Improved speech enhancement using TCN with multiple encoder-decoder layers. In *Proc. Interspeech*, Shanghai, China, October 2020.

[50] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L.Seltzer, and Sanjeev Khudanpur. A study on data augmentation of reverberant speech for robust speech recognition. In *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Proc.*, pages 5220–5224, New Orleans, LA, May 2017.

[51] Yuma Koizumi, Kohei Yatabe, Marc Delcroix, Yoshiki Maxuxama, and Daiki Takeuchi. Speech enhancement using self-adaptation and multi-head self-attention. In *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Proc.*, pages 181–185, Barcelona, Spain, May 2020.

[52] Yuichiro Koyama, Tyler Vuong, Stefan Uhlich, and Bhiksha Raj. Exploring the best loss function for DNN-based low-latency speech enhancement with temporal convolutional networks. *arXiv:2005.11611*, May 2020.

[53] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2Void-learning denoising from single noisy images. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 2129–2137, Long Beach, CA, June 2019.

[54] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R. Hershey. SDR – half-baked or well done? In *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Proc.*, pages 626–630, Brighton, United Kingdom, May 2019.

[55] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, pages 556–562, Denver, CO, 2001.

[56] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. Pre-Print 1803.04189, ArXiV, 2018.

[57] Andong Li, Chengshi Zheng, Cunhang Fan, Renhua Peng, and Xiaodong Li. A recursive network with dynamic attention for monaural speech enhancement. *arXiv:2003.12973*, March 2020.

[58] Andong Li, Chengshi Zheng, Renhua Peng, and Xiaodong Li. Two heads are better than one: A two-stage approach for monaural noise reduction in the complex domain. *arXiv:2011.01561*, November 2020.

[59] Chien-Feng Liao, Yu Tsao, Xugang Lu, and Hisashi Kawai. Incorporating symbolic sequential modeling for speech enhancement. In *Proc. Interspeech*, pages 2733–2737, Graz, Austria, September 2019.

[60] Ju Lin, Sufeng Niu, Adriaan J. van Wijngaarden, Jerome L. McClendon, Melissa C. Smith, and Kuang-Ching Wang. Improved speech enhancement using a time-domain GAN with mask learning. In *Proc. Interspeech*, pages 3286–3290, Shanghai, China, October 2020.

[61] Ju Lin, Sufeng Niu, Zice Wei, Xiang Lan, Adriaan J. van Wijngaarden, Melissa C. Smith, and Kuang-Ching Wang. Speech enhancement using forked generative adversarial networks with spectral subtraction. In *Proc. Interspeech*, pages 3163–3167, Graz, Austria, September 2019.

[62] Ju Lin, Adriaan J van Wijngaarden, Kuang-Ching Wang, and Melissa C Smith. Speech enhancement using multi-stage self-attentive temporal convolutional networks. *arXiv preprint arXiv:2102.12078*, 2021.

[63] Richard Lippmann, Edward Martin, and D Paul. Multi-style training for robust isolated-word speech recognition. In *ICASSP'87. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 12, pages 705–708. IEEE, 1987.

[64] Chang-Le Liu, Sze-Wei Fu, You-Jin Li, Jen-Wei Huang, Hsin-Min Wang, and Yu Tsao. Multichannel speech enhancement by raw waveform-mapping using fully convolutional networks. *IEEE/ACM Trans. Audio, Speech, Language Process.*, 28:1888–1900, 2020.

[65] Philipos C. Loizou. *Speech Enhancement.* CRC Press, Boca Raton, FL, 2nd edition, 2013.

[66] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori. Speech enhancement based on deep denoising autoencoder. In *Proc. Interspeech*, pages 436–440, Lyon, France, August 2013.

[67] Yen-Ju Lu, Chien-Feng Liao, Xugang Lu, Jeih weih Hung, and Yu Tsao. Incorporating broad phonetic information for speech enhancement. In *Proc. Interspeech*, pages 2417–2421, Shanghai, China, October 2020.

[68] Yiping Lu, Zhuohan Li, Di He, Zhiqing Sun, Bin Dong, Tao Qin, Liwei Wang, and Tie-Yan Liu. Understanding and improving transformer from a multi-particle dynamic system point of view. *arXiv:1906.02762*, June 2019.

[69] Yi Luo and Nima Mesgarani. Conv-TasNet: Surpassing ideal time-frequency magnitude masking for speech separation. *IEEE/ACM Trans. Audio, Speech, Language Process.*, 27(8):1256–1266, August 2019.

[70] Andrew L. Maas, Quoc V. Le, Tyler M. O'Neil, Oriol Vinyals, Patrick Nguyen, and Andrew Y. Ng. Recurrent neural networks for noise reduction in robust ASR. In *Proc. Interspeech*, pages 22–25, Portland, OR, September 2012.

[71] Craig Macartney and Tillman Weyde. Improved speech enhancement with the Wave-U-Net. *arXiv:1811.11307*, November 2018.

[72] Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda Shapiro, and Hannaneh Hajishirzi. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In *Proceedings of the european conference on computer vision (ECCV)*, pages 552–568, 2018.

[73] Daniel Michelsanti and Zheng-Hua Tan. Conditional generative adversarial networks for speech enhancement and noise-robust speaker verification. *arXiv preprint arXiv:1709.01703*, 2017.

[74] Arun Narayanan and DeLiang Wang. Ideal ratio mask estimation using deep neural networks for robust speech recognition. In *Proc. IEEE Int'l Conf. on Acoustics, Speech and Signal Proc.*, pages 7092–7096, Vancouver, BC, May 2013.

[75] Arun Narayanan and DeLiang Wang. Investigation of speech separation as a front-end for noise robust speech recognition. *IEEE/ACM Trans. Audio, Speech, Language Process.*, 22(4):826–835, April 2014.

[76] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Proc. European Conf. on Computer Vision*, pages 483–499, Amsterdam, The Netherlands, October 2016.

[77] Kuldip Paliwal, Kamil Wójcicki, and Benjamin Shannon. The importance of phase in speech enhancement. *Speech Commun.*, 53(4):465–494, April 2011.

[78] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. LibriSpeech: an ASR corpus based on public domain audio books. In *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Proc.*, pages 5206–5210, Brisbane, Australia, April 2015.

[79] Ashutosh Pandey and Deliang Wang. On adversarial training and loss functions for speech enhancement. In *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Proc.*, pages 5414–5418, Calgary, AB, April 2018.

[80] Ashutosh Pandey and DeLiang Wang. A new framework for CNN-based speech enhancement in the time domain. *IEEE/ACM Trans. Audio, Speech, Language Process.*, 27(7):1179–1188, July 2019.

[81] Ashutosh Pandey and DeLiang Wang. TCNN: Temporal convolutional neural network for real-time speech enhancement in the time domain. In *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Proc.*, pages 6875–6879, Brighton, United Kingdom, May 2019.

[82] Ashutosh Pandey and DeLiang Wang. Dense CNN with self-attention for time-domain speech enhancement. *arXiv:2009.01941*, September 2020.

[83] Santiago Pascual, Antonio Bonafonte, and Joan Serrà. SEGAN: Speech enhancement generative adversarial network. In *Proc. Interspeech*, pages 3642–3646, Stockholm, Sweden, August 2017.

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

[85] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. The Kaldi speech recognition toolkit. In *Proc. IEEE Workshop on Autom. Speech Recognition and Understanding*, Waikoloa, HI, December 2011.

[86] Leyuan Qu, Cornelius Weber, and Stefan Wermter. Multimodal target speech separation with voice and face references. In *Proc. Interspeech*, pages 1416–1420, Shanghai, China, October 2020.

[87] Chandan K. A. Reddy, Harishchandra Dubey, Vishak Gopal, Ross Cutler, Sebastian Braun, Hannes Gamper, Robert Aichner, and Sriram Srinivasan. ICASSP 2021 deep noise suppression challenge. *arXiv:2009.06122*, September 2020.

[88] Chandan KA Reddy, Ebrahim Beyrami, Harishchandra Dubey, Vishak Gopal, Roger Cheng, Ross Cutler, Sergiy Matusevych, Robert Aichner, Ashkan Aazami, Sebastian Braun, et al. The interspeech 2020 deep noise suppression challenge: Datasets, subjective speech quality and testing framework, 2020.

[89] Dayana Ribas, Jorge Llombart, Antonio Miguel, and Luis Vicente. Deep speech enhancement for reverberated and noisy signals using wide residual networks. *arXiv:1901.00660*, January 2019.

[90] Anthony W. Rix, John G. Beerends, Michael P. Hollier, and Andries P. Hekstra. Perceptual evaluation of speech quality (PESQ) – a new method for speech quality assessment of telephone networks and codecs. In *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Proc.*, pages 749–752, Salt Lake City, UT, May 2001.

[91] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, Oct. 1986.

[92] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Proc. Annual Conf. Int'l Speech Commun. Assoc.*, pages 338–342, Singapore, Singapore, September 2014.

[93] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, pages 2234–2242. Curran Associates, Inc., 2016.

[94] Pascal Scalart and Jozue Vieira Filho. Speech enhancement based on a priori signal to noise estimation. In *Proc. IEEE Int'l Conf. on Acoustics, Speech, and Signal Proc.*, pages 629–632, Atlanta, GA, May 1996. IEEE.

[95] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*, 2019.

[96] Michael L. Seltzer, Dong Yu, and Yongqiang Wang. An investigation of deep neural networks for noise robust speech recognition. In *Proc. IEEE Int'l Conf. on Acoustics, Speech and Signal Proc.*, pages 7398–7402, Vancouver, BC, May 2013. IEEE.

[97] Meet H. Soni, Neil Shah, and Hemant A. Patil. Time-frequency masking-based speech enhancement using generative adversarial network. In *Proc. IEEE Int'l Conf. on Acoustics, Speech and Signal Proc.*, pages 5039–5043, Calgary, AB, April 2018.

[98] Y. Cem Subakan and Paris Smaragdis. Generative adversarial source separation. In *Proc. IEEE Int'l Conf. on Acoustics, Speech and Signal Proc.*, pages 26–30, Calgary, AB, Apr. 2018. IEEE.

[99] Cees H. Taal, Richard C. Hendriks, Richard Heusdens, and Jesper Jensen. An algorithm for intelligibility prediction of time–frequency weighted noisy speech. *IEEE/ACM Trans. Audio, Speech, Language Process.*, 19(7):2125–2136, September 2011.

[100] Ke Tan and DeLiang Wang. A convolutional recurrent neural network for real-time speech enhancement. In *Proc. Interspeech*, pages 3229–3233, Hyderabad, India, September 2018.

[101] Ke Tan and DeLiang Wang. Complex spectral mapping with a convolutional recurrent network for monaural speech enhancement. In *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Proc.*, pages 6865–6869, Brighton, United Kingdom, May 2019.

[102] Ke Tan and DeLiang Wang. Learning complex spectral mapping with gated convolutional recurrent networks for monaural speech enhancement. *Proc. IEEE/ACM Trans. on Audio, Speech and Language Proc.*, 28:380–390, 2020.

[103] Ke Tan, Yong Xu, Shi-Xiong Zhang, Meng Yu, and Dong Yu. Audio-visual speech separation and dereverberation with a two-stage multimodal network. *IEEE Journal of Selected Topics in Signal Processing*, 14(3):542–553, March 2020.

[104] Xin Tang, Jun Du, Li Chai, Yannan Wang, Qing Wang, and Chin-Hui Lee. A LSTM-based joint progressive learning framework for simultaneous speech dereverberation and denoising. In *Proc. Asia-Pacific Signal and Inform. Proc. Assoc. Annual Summit and Conf.*, pages 274–278, Lanzhou, China, November 2019.

[105] Joachim Thiemann, Nobutaka Ito, and Emmanuel Vincent. The diverse environments multichannel acoustic noise database (DEMAND): A database of multichannel environmental noise recordings. In *Proc. Int'l Conf. on Acoustics*, pages 1–6, Montréal, Canada, June 2013.

[106] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-RMSprop: Divide the gradient by a running average of its recent magnitude. *Coursera: Neural networks for Machine Learning*, 4(2):26–31, 2012.

[107] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[108] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition*, pages 9446–9454, 2018.

[109] Cassia Valentini-Botinhao, Xin Wang, Shinji Takaki, and Junichi Yamagishi. Investigating RNN-based speech enhancement methods for noise-robust Text-to-Speech. In *Proc. ISCA Speech Synthesis Workshop*, pages 146–152, Sunnyvale, CA, September 2016.

[110] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A generative model for raw audio. In *Proc. ISCA Speech Synthesis Workshop*, page 125, Sunnyvale, CA, September 2016.

[111] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *J. Machine Learning Res.*, 9:2579–2605, November 2008.

[112] Andrew Varga and Herman J.M. Steeneken. Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems. *Speech Commun.*, 12(3):247–251, July 1993.

[113] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. Advances in Neural Information Proc. Sys.*, pages 5998–6008, Long Beach, CA, December 2017.

[114] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification. In *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Proc.*, pages 4879–4883, Calgary, AB, April 2018.

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

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

[117] Ke Wang, Junbo Zhang, Sining Sun, Yujun Wang, Fei Xiang, and Lei Xie. Investigating generative adversarial networks based speech dereverberation for robust speech recognition, 2018.

[118] Qing Wang, Jun Du, Li-Rong Dai, and Chin-Hui Lee. A multiobjective learning and ensembling approach to high-performance speech enhancement with compact neural network architectures. *IEEE/ACM Trans. on Audio, Speech and Language Proc.*, 26(7):1181–1193, 2018.

[119] Quan Wang, Hannah Muckenhirn, Kevin Wilson, Prashant Sridhar, Zelin Wu, John Hershey, Rif A. Saurous, Ron J. Weiss, Ye Jia, and Ignacio Lopez Moreno. Voicefilter: Targeted voice separation by speaker-conditioned spectrogram masking. In *Proc. Interspeech*, pages 2728–2732, Graz, Austria, September 2019.

[120] Yuxuan Wang, Arun Narayanan, and DeLiang Wang. On training targets for supervised speech separation. *IEEE/ACM Trans. Audio, Speech, Language Process.*, 22(12):1849–1858, December 2014.

[121] Nils L Westhausen and Bernd T Meyer. Dual-signal transformation lstm network for real-time noise suppression. *arXiv preprint arXiv:2005.07551*, 2020.

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

[123] Yangyang Xia, Sebastian Braun, Chandan KA Reddy, Harishchandra Dubey, Ross Cutler, and Ivan Tashev. Weighted speech distortion losses for neural-network-based real-time speech enhancement, 2020.

[124] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee. A regression approach to speech enhancement based on deep neural networks. *IEEE/ACM Trans. Audio, Speech, Language Process.*, 23(1):7–19, January 2015.

[125] Takuya Yoshioka, Armin Sehr, Marc Delcroix, Keisuke Kinoshita, Roland Maas, Tomohiro Nakatani, and Walter Kellermann. Making machines understand us in reverberant rooms: Robustness against reverberation for automatic speech recognition. *IEEE Signal Process. Mag.*, 29(6):114–126, 2012.

[126] Liwen Zhang, Ziqiang Shi, Jiqing Han, Anyan Shi, and Ding Ma. FurcaNeXT: End-to-end monaural speech separation with dynamic gated dilated temporal convolutional networks. In *Proc. Int'l Conf. on Multimedia Modeling*, pages 653–665, Prague, Czech Republic, June 2020.

[127] Qiquan Zhang, Aaron Nicolson, Mingjiang Wang, Kuldip K. Paliwal, and Chenxu Wang. Monaural speech enhancement using a multi-branch temporal convolutional network. *arXiv:1912.12023*, December 2019.

[128] Qiquan Zhang, Aaron Nicolson, Mingjiang Wang, Kuldip K. Paliwal, and Chenxu Wang. DeepMMSE: A deep learning approach to MMSE-based noise power spectral density estimation. *IEEE/ACM Trans. Audio, Speech, Language Process.*, 28:1404–1415, April 2020.

[129] Yan Zhao and DeLiang Wang. Noisy-reverberant speech enhancement using DenseUNet with time-frequency attention. In *Proc. Interspeech*, pages 3261–3265, Shanghai, China, October 2020.

[130] Yan Zhao, DeLiang Wang, Buye Xu, and Tao Zhang. Monaural speech dereverberation using temporal convolutional networks with self attention. *IEEE/ACM Trans. Audio, Speech, Language Process.*, 28:1598–1607, May 2020.

[131] Yan Zhao, Zhong-Qiu Wang, and DeLiang Wang. Two-stage deep learning for noisy-reverberant speech enhancement. *IEEE/ACM transactions on audio, speech, and language processing*, 27(1):53–62, 2018.