Speech Enhancement with Improved Deep Learning Methods

Mojtaba Hasannezhad

A Thesis
in
The Department
of
Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy (Electrical and Computer Engineering) at
Concordia University
Montréal, Québec, Canada

June 2021

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: **Mojtaba Hasannezhad**

Entitled: **Speech Enhancement with Improved Deep Learning Methods**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Electrical and Computer Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Prof. Anjali Awasthi

_____ External Examiner
Prof. Hon K. Kwan

_____ External to Program
Prof. Arsh Mohammadi

_____ Examiner
Prof. M.N.S. Swamy

_____ Examiner
Prof. Omair Ahmad

_____ Thesis Supervisor
Prof. Wei-Ping Zhu

Approved by _____
Prof. Wei-Ping Zhu, Graduate Program Director

July 29, 2021 _____
Prof. Mourad Debbabi, Dean
Gina Cody School of Engineering and Computer Science

Abstract

Speech Enhancement with
Improved Deep Learning Methods

Mojtaba Hasannezhad, Ph.D.

Concordia University, 2021

In real-world environments, speech signals are often corrupted by ambient noises during their acquisition, leading to degradation of quality and intelligibility of the speech for a listener. As one of the central topics in the speech processing area, speech enhancement aims to recover clean speech from such a noisy mixture. Many traditional speech enhancement methods designed based on statistical signal processing have been proposed and widely used in the past. However, the performance of these methods was limited and thus failed in sophisticated acoustic scenarios. Over the last decade, deep learning as a primary tool to develop data-driven information systems has led to revolutionary advances in speech enhancement. In this context, speech enhancement is treated as a supervised learning problem, which does not suffer from issues faced by traditional methods. This supervised learning problem has three main components: input features, learning machine, and training target. In this thesis, various deep learning architectures and methods are developed to deal with the current limitations of these three components.

First, we propose a serial hybrid neural network model integrating a new low-complexity fully-convolutional convolutional neural network (CNN) and a long short-term memory (LSTM) network to estimate a phase-sensitive mask for speech enhancement. Instead of using traditional acoustic features as the input of the model, a CNN is employed to automatically extract sophisticated speech features that can maximize the performance of a model. Then, an LSTM network is chosen as the learning machine to model strong temporal dynamics of speech. The model is

designed to take full advantage of the temporal dependencies and spectral correlations present in the input speech signal while keeping the model complexity low. Also, an attention technique is embedded to recalibrate the useful CNN-extracted features adaptively. Through extensive comparative experiments, we show that the proposed model significantly outperforms some known neural network-based speech enhancement methods in the presence of highly non-stationary noises, while it exhibits a relatively small number of model parameters compared to some commonly employed DNN-based methods.

Most of the available approaches for speech enhancement using deep neural networks face a number of limitations: they do not exploit the information contained in the phase spectrum, while their high computational complexity and memory requirements make them unsuited for real-time applications. Hence, a new phase-aware composite deep neural network is proposed to address these challenges. Specifically, magnitude processing with spectral mask and phase reconstruction using phase derivative are proposed as key subtasks of the new network to simultaneously enhance the magnitude and phase spectra. Besides, the neural network is meticulously designed to take advantage of strong temporal and spectral dependencies of speech, while its components perform independently and in parallel to speed up the computation. The advantages of the proposed PACDNN model over some well-known DNN-based SE methods are demonstrated through extensive comparative experiments.

Considering that some acoustic scenarios could be better handled using a number of low-complexity sub-DNNs, each specifically designed to perform a particular task, we propose another very low complexity and fully convolutional framework, performing speech enhancement in short-time modified discrete cosine transform (STMDCT) domain. This framework is made up of two main stages: classification and mapping. In the former stage, a CNN-based network is proposed to classify the input speech based on its utterance-level attributes, i.e., signal-to-noise ratio and gender. In the latter stage, four well-trained CNNs specialized for different specific and simple tasks transform the STMDCT of noisy input speech to the clean one. Since this framework is designed to perform in the STMDCT domain, there is no need to deal with the phase information, i.e., no

phase-related computation is required. Moreover, the training target length is only one-half of those in the previous chapters, leading to lower computational complexity and less demand for the mapping CNNs. Although there are multiple branches in the model, only one of the expert CNNs is active for each time, i.e., the computational burden is related only to a single branch at anytime. Also, the mapping CNNs are fully convolutional, and their computations are performed in parallel, thus reducing the computational time. Moreover, this proposed framework reduces the latency by %55 compared to the models in the previous chapters. Through extensive experimental studies, it is shown that the MBSE framework not only gives a superior speech enhancement performance but also has a lower complexity compared to some existing deep learning-based methods.

Acknowledgments

First and foremost, I would like to express my heartfelt gratitude and appreciation to my supervisor, Prof. Wei-Ping Zhu, for directing me to the field of speech enhancement, teaching me the fundamentals as well as cutting-edge techniques, supporting me when I was stymied in my research, and encouraging me when I faced difficulties.

In addition, I want to express my sincere appreciation to Prof. Benoit Champagne, McGill University, Canada, for providing me with a lot of help and support in my research and publication, which helped me increase my theoretical understanding in speech processing and enhance my technical writing skills.

My gratitude also goes to Concordia for providing me with the opportunity to study at such an excellent university, as well as to Microchip in Ottawa for sponsoring our NSERC CRD project.

I also like to extend my deep gratefulness to my group members and officemates, Ali Mohebbi, Hongjiang Yu, Guilherme Zilli, Kalpesh Ranipa, Zhiheng Ouyang. I am also grateful to Jafar Chaeb who has been always there for me, Emad Fallahzadeh, my great roommate, Sajjad Talebi, my big brother, whose very presence spreads love and joy, for their help and comforts throughout my life during the past four years.

I sincerely appreciate the professors in my examining committee for their guidance and comments on my comprehensive exam, proposal, seminar, and thesis.

Finally, I would like to express my deepest thanks and love to my parents and siblings. Their selfless love, encouragement, and support are always a source of strength for me to overcome all of life's disappointments and problems.

Contents

List of Figures

List of Tables

List of Abbreviations

AC-MFCC autocorrelation sequence MFCC

AMS Amplitude Modulation Spectrum

ASR Automatic Speech Recognition

BPTT Backpropagation Through Time

cIRM Complex Ideal Ratio Mask

CNN Convolutional Neural Network

CRN Convolutional Recurrent Neural Network

CS Complex Spectrogram

CUDA Compute Unified Device Architecture

DCT Discrete Cosine Transform

DFT Discrete Fourier Transform

DNN Deep Neural Network

DSCC Delta-Spectral Cepstral Coefficient

DT Decision Tree

EL Ensemble Learning

FC Fully-Connected Neural Network

FLOPs Floating-Points Operations

GAN Generative Adversarial Network

GD Group Delay

GF Gammatone Feature

GF-TPS Gammatone Frequency Target Power Spectrum

GFB Gabor filterbank

GFCC Gammatone Frequency Cepstral Coefficients

GFMC Gammatone Frequency Modulation Coefficient

GPUs Graphics Processing Units

IBM Ideal Binary Mask

IF Instantaneous Frequency

IFD Instantaneous Frequency Deviation

ILD Interaural Level Difference

IMDCT Inverse Modified Discrete Cosine Transform

IoT Internet of Things

IRM Ideal Ratio Mask

ITD Interaural Time Difference

KNN K-Nearest Neighbors

LDA Linear Discriminant Analysis

LOG-MAG log spectral magnitude

LOG-MEL Log Mel-Spectrum Feature

LSTM Long Short-Term Memory

MDCT Modified Discrete Cosine Transform

MFCC Mel-Frequency Cepstral Coefficients

MMSE Minimum Mean Square Error

MRCG Multi-Resolution Cochleagram

MSE Mean Square Error

MVDR Minimum Variance Distortionless Response

NB Naive Bayes

OLA Overlap-Add

ORM Optimal Ratio Mask

PAC-MFCC phase autocorrelation MFCC

PACDNN Phase-Aware Composite Deep Neural Network

PCA Principal Component Analysis

PD Phase Derivative

PITCH Time-frequency features based on pitch tracking

PLP Perceptual Linear Prediction

PNCC Power Normalized Cepstral Coefficients

PSM Phase Sensitive Mask

PSSA Phase-Sensitive Spectrum Approximation

RAS-MFCC relative autocorrelation sequence MFCC

RASTA-PLP relative spectral transform PLP

ReLU Rectified Linear Unit

RNN Recurrent Neural Network

SA Signal Approximation

SAE Squeeze-and-Excitation

SMM Spectral Magnitude Mask

SNR Signal-to-Noise Ratio

SSNR Segmental Signal-to-Noise Ratio

STDCT Short-Time Discrete Cosine Transform

STFT Short-Time Fourier Transform

STMDCT Short-Time Modified Discrete Cosine Transform

STOI Short-Time Objective Intelligibility

SVM support vector machine

t-SNE t-Distributed Stochastic Neighbor Embedding

tanh Hyperbolic Tangent Activation Function

TF Time-Frequency

TMS Target Magnitude Spectrogram

TPUs Tensor Processing Units

WAV raw waveform

ZCPA Zero-Crossings with Peak-Amplitudes

Chapter 1

Introduction

In this chapter, a brief introduction about speech enhancement and its real-world applications is first presented. Some traditional speech enhancement methods that are designed based on statistical signal processing and have been widely used in the past are then addressed. Next, a short review of basic deep learning methods in speech enhancement, including magnitude enhancement, phase enhancement and enhancement of speech under unseen conditions, is followed. Afterwards, the motivation and objectives of this research are expressed. Finally, the major contributions and chapter-by-chapter organization of this thesis are described.

1.1 Speech Enhancement and Its Applications

Speech plays the most significant role in human communication as the most common and convenient information carrier. In everyday listening situations, speech signals are often corrupted by ambient noises during their acquisition, leading to the degradation of quality and intelligibility of the speech for a listener. In a noisy environment, a normal human auditory system has a remarkable capability to track the sound of a specific speaker; however, it becomes very difficult for hearing-impaired listeners. The less the ratio of signal to noise is, the more challenging it is to understand speech for both normal and hearing-impaired listeners. Speech enhancement is one of the crucial topics in the speech processing area. It aims to suppress the unwanted ambient noise

contained in the acquired speech signal, which may include non-speech noise, interfering speech, and/or room reverberation, either to improve its quality or as a preprocessing procedure to make these applications robust to various noises. As a signal processing problem, speech enhancement aims to emulate the human auditory system to separate the particular target speech from a multiple source mixture [7]. During past decades, many researchers have studied speech enhancement methods to improve the user experience in speech communication.

Speech enhancement brings significant advantages in various applications such as automatic speech recognition (ASR), smart home devices, hearing prosthesis, voice communications, etc. [7]. For instance, many human-machine interfaces based on speech use ASR for interaction with intelligent electronic devices, such as dictation systems, voice-enabled search for mobile devices, and voice-controlled home entertainment systems. For large-scale real-world applications, speech interfaces facilitate human-machine interactions and significantly enhance the efficiency and functionality of home automation, which is emerging as one of the most popular applications of the internet of things (IoT). Indeed, there is currently an increasing number of smart home devices available on the market like Amazon Echo and Google Home that allow users to control various devices in their home or access external information sources via source command, making their lives more connected than ever to the internet. These cloud-based assistants equipped with artificial intelligence can respond in real-time to human needs via voice recognition. They can also help people with kinetic disabilities and improve their quality of life. Also, they are advantageous for the hearing prosthesis, and mobile communication [18]. Since the human-machine interfaces have to perform under difficult acoustic conditions, like in a living room which involves a wide variety of highly non-stationary noise sources, such as children's voices, television, or ambient music, robustness to noise has become an increasingly important issue, especially in the presence of room reverberation [19]. Further and even more importantly, speech enhancement increases hearing impaired users' ability to understand speech. Hearing aids perform very well in a normal situation, while the performance deteriorates in noisy environments. Hence, speech enhancement as a pre-processing stage before amplification in these devices helps users have a better listening

experience. Robustness to noise and reverberation remains a challenging problem that is actively driving research on speech enhancement. Consequently, with the fast development of the systems, like those mentioned above, there is an increasing demand for advanced speech enhancement algorithms.

Speech enhancement methods can be categorized into single-channel (or monaural), where a single microphone is used to capture the speech, and multi-channel, which takes advantage of spatial information obtained from multiple microphones. Monaural speech enhancement is more challenging as it relies on a smaller set of observations. When combined with spatial filtering (e.g., beamforming), it also forms the basis for multi-channel techniques. Apart from that, speech denoising, speech dereverberation, and speaker separation are considered three subtasks of speech enhancement. Speech denoising aims to suppress the background noise from an acquired noisy signal. When an acquired mixture contains two or more speech signals, speaker separation aims to separate these voices from each other. In a real indoor environment, speech could also be corrupted by reverberation, which is its echo, from surface reflections. Speech dereverberation deals with this type of distortion that corrupts speech signals along with time and frequency. In this work, our main interest lies in single-channel speech enhancement, although generalization to multi-channel processing is possible to recover the desired clean speech signal from the noisy observation. To this end, in this chapter, we briefly review the advances of speech enhancement from two perspectives, namely, traditional statistical signal processing-based method and the cutting edge deep learning approach methods.

1.2 Traditional Speech Enhancement Methods

Researchers have advanced numerous approaches to attenuate or remove noise from a corrupted speech signal in past decades. One of the most intuitive traditional speech enhancement methods is spectral subtraction [20]. In this method, the noise power spectrum is estimated in speech-absent

segments, which is then subtracted from the noisy speech spectrum. This method is easy to implement at low computational cost, and fast enough for real-time speech enhancement applications. However, the accuracy of estimating the noise power spectrum limits to a large extent the performance of this method. Extra distortions would appear in the processed clean speech with either under or over-estimation of noise. One example of these distortions is called notorious musical noise, which is due to the presence of residual peaks in the spectrum of the processed speech. The existence of musical noise in the processed speech could be more annoying than the original background noise. Hence, the difficulty of estimating an accurate noise impedes enhancement performance with the spectral subtraction method. To tackle this problem and improve speech enhancement performance, extensions to the spectral subtraction method were introduced in [21–24] which present a more flexible algorithm. The algorithms in [21, 22] aim to adjust the estimated noise spectrum to regulate the remaining residual and musical noise in the processed signal. In the former, the goal is to subtract an overestimate of the noise power spectrum from the noisy speech one and use a spectral floor factor to prevent the resultant signal from crossing a preset minimum level. The authors' goal in the latter is to lessen the amount of noise that is to be subtracted from the noisy speech spectrum and whiten it, leading to improved speech enhancement results. Benefiting from the fact that the effect of noise on speech spectrum differs depending on the frequency band, a multi-band spectral subtraction method was studied in [23, 24]. In these methods, spectral subtraction performs on individual non-overlapping frequency sub-bands of noisy speech. The authors showed that these methods lead to better speech enhancement results than the original spectral subtraction method.

An alternative to spectral subtraction is the Wiener filtering that is a well-known traditional speech enhancement method in the literature [7, 25, 26]. The idea behind this method is to multiply the noisy speech spectrum by a Wiener gain function that can be considered as a linear filter. This gain function is estimated by minimizing the mean square error (MSE) between spectra of the estimated and clean speech signals. Unlike the spectral subtraction method that introduces musical noise, Wiener filtering causes residual noise in the estimated speech signal. Many studies in the

relevant literature have been carried out to implement Wiener filtering in a more efficient way in order to tackle its shortcomings, such as [27–29]. In [27], a perceptually motivated approach was proposed, which is a modified version of the Wiener filtering method to enhance the speech corrupted by colored noise. This approach suppresses the perceptual effect of the residual noise by considering the frequency masking properties of the human auditory system. The proposed technique in [28] aims to estimate short-term linear predictive parameters of speech and noise signals from noisy input mixture and exploit them in waveform enhancement schemes. An adaptive time-domain Wiener filtering approach was proposed in [29] which depends on the local statistics of the speech signal, including local mean and variance. The advantage of this method over the original Wiener filtering was shown in the presence of adaptive white Gaussian noise and colored noise.

Other class of traditional speech enhancement methods is the Bayesian estimators. The idea is to drive an estimator by minimizing the statistical expectation of a minimum mean-square error (MMSE) cost function that penalizes errors in the clean speech estimate. MMSE is commonly used in the estimation theory due to the facility of its evaluation and the fact that it is mathematically tractable. Many extensions to the Bayesian estimator approach have been proposed in the related literature to model the error between the clean and estimated speech spectral magnitude, such as in [30, 31]. Ephraim and Malah [30] proposed an MMSE-based speech enhancement method that estimates the clean speech magnitude based on the noisy observation. They showed the superiority of this speech enhancement class over spectral subtraction and Wiener filtering. The authors in [31] proposed an extension to the Bayesian estimators where a power law and a weighting factor are involved in the cost function. These parameters were chosen based on characteristics of the human auditory system that leads to decreasing the estimator gain at high frequencies. These methods generally yield lower residual noise but still produce speech distortion.

Most of these traditional methods rely on some assumptions which do not fit real-world scenarios. For instance, most of these methods assume speech and noise being uncorrelated and noise

being stationary or more stationary than speech signal. These assumptions lead to inaccurate estimation of the underlying model statistics. In particular, these methods often fail to suppress highly non-stationary noises and unexpected adverse real-world scenarios. Hence, we move on with new deep learning-based methods not suffering from the limitations and shortcomings of the aforementioned traditional methods.

1.3 Deep Learning-Based Speech Enhancement Methods

Recently, deep neural network (DNN) -based methods have drawn a lot of attention thanks to the remarkable advances in parallel computing resources, including hardware and software. In terms of software, the advent of computing platforms like compute unified device architecture (CUDA) and deep learning libraries such as Tensorflow [32] and PyTorch [33] facilitated the implementation of DNN-based algorithms. In the matter of hardware, graphics processing units (GPUs) and tensor processing units (TPUs) provided high computational speed for DNN-based algorithms. DNN-based methods were first used in the computer vision field, and they were then extended to many other fields, such as handwriting classification [34], automatic machine translation [35], speech recognition [36], and language modeling [37].

DNN-based approaches in the speech processing area, including speech enhancement, speech synthesis, and speaker separation, do not rely on any prior assumption on the speech and noise, nor suffer from the issues faced by traditional methods based on statistical and mathematical models. It is believed that DNN-based methods outperform traditional ones on a large scale [7]. Moreover, DNN can offer low latency processing, which is crucial to many real-time applications, such as hearing aids [38].

Different DNN types, including fully-connected neural networks (FC), recurrent neural networks (RNN), and convolutional neural networks (CNN), were first vastly investigated in the computer vision field. The analyzed images in this field are usually 2-dimensional (2D) (or 3D to consider RGB channels). Unlike image, speech signal is 1D in the time domain and exhibits strong

correlations between consecutive samples. To benefit from DNN-based approaches in the speech processing field, short-time Fourier transform (STFT) of time-domain speech is usually calculated. Speech STFT then becomes a 2D representation (spectrogram) in the time-frequency (TF) domain where the horizontal and vertical axes represent time frames and frequency bins. Thus, most of the advanced DNN-based approaches in the computer vision field can be used in the speech processing area with the required adjustments [39].

The remarkable capability of DNN in modeling highly complex transformations has vastly advanced speech enhancement in adverse and variable acoustic scenarios. A DNN typically learns the highly complex relationship between a set of input signal features, either raw or processed data, and the desired training target, which could be the speech spectrum, a spectral mask, or any of their variations based on which the clean speech can be reconstructed.

Since speech enhancement in the STFT domain is desired, magnitude and phase enhancement have to be dealt with. In the following, an overview of state of the art in magnitude and phase enhancement is presented.

1.3.1 Magnitude Enhancement

Many studies have attempted to enhance the spectral magnitude of speech using DNN-based methods. Wang *et al.* in [40, 41], which was later extended to [42], proposed applying DNN to speech enhancement within a masking-based framework, as shown in Fig. 1. The authors employed an FC network for subband classification to estimate IBM. They initiated network parameters with restricted Boltzmann machine pretraining and used FC as a binary classifier. Note that RBM is a stochastic generative neural network. According to [43], the reason for using the RBM pretraining is that the FC network training starting with a randomly initialized network might fall in a poor local minimum, particularly in the case of a large number of hidden layers; hence, RBM pretraining as an unsupervised learning machine is applied [1, 20]. The input is fed to a 64-channel Gammatone filterbank to drive subband signals. The acoustic features are extracted for each TF unit, and they are then fed to 64 subband FC networks to learn more discriminative features. Finally, the

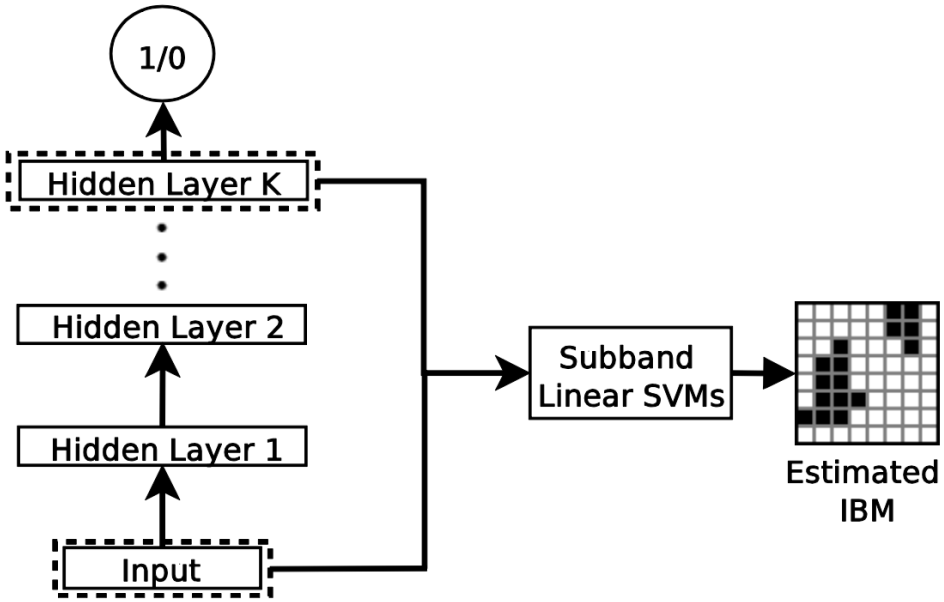


Figure 1: Schematic diagram of the FC-SVM system for IBM estimation.

input and learned features are concatenated, and the whole is fed to a simple linear support vector machine (SVM) to estimate the subband IBM efficiently. The authors reported strong enhancement results compared to some traditional speech enhancement methods [7].

The first mapping-based method in speech enhancement was introduced in [44] where the Mel-frequency power spectrum of noisy speech is mapped to that of clean speech using a deep auto-encoder comprising stacked multiple basic auto-encoders. A basic auto-encoder has an asymmetric architecture with one hidden layer, which performs as an unsupervised learning machine that maps an input signal to itself. They showed that adding depth of the deep auto-encoder consistently increases the performance given a large training data set.

The other powerful mapping method has been introduced in [1] where an FC network was used to map the log power spectrum of the noisy speech to the clean one. The block diagram of this work is shown in Fig. 2. To avoid getting stuck in the local minimum, a pretraining step using a restricted Boltzmann machine is adopted in this work as well. In the testing stage, the FC network estimates the spectrum of the clean speech from the noisy one. A dropout training technique was also adopted in this work to avoid over-fitting. Furthermore, to reduce discontinuity and obtain better enhancement, the acoustic context information, including full frequency band and context

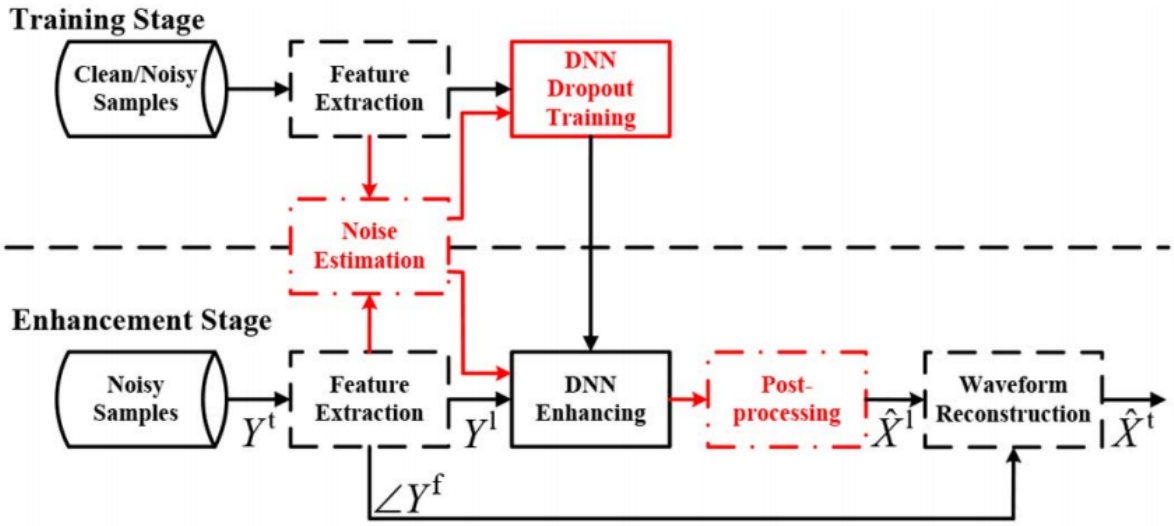


Figure 2: Block diagram of a mapping-based speech enhancement system using an FC network. In the testing stage, the network is trained with clean and noisy features of the input speech. In the testing stage, the network maps the noisy input features to the clean ones and then reconstructs the clean speech signal [1].

frame expanding, was utilized. This well-known framework showed significant improvements over traditional MMSE-based methods and could suppress non-stationary noises. Moreover, through the introduced DNN method, the musical noise does not appear in the output, which results in enhanced speech quality. Further, this work yields very good enhancement results for a speech signal mixed with unseen noises.

Some previous studies attempted to benefit from different training targets in their DNN-based speech enhancement framework. For example, the authors in [2] aimed to take advantage of both masking- and mapping-based methods. The argument is that neither of the training targets performs perfectly, while they can complement each other. Hence, an FC network is employed to jointly estimate IBM, IRM, and clean speech spectrogram, as shown in Fig. 3. This paper demonstrates that the joint mask- and spectrum-based training targets lead to a better speech enhancement performance than a single training target. Xu *et al.* [1], put one step further with introducing a multi-target and multi-input framework, illustrated in Fig. 4. Their auxiliary structure learns acoustic features, like MFCCs, and categorical information, like IBM, and integrates them into a

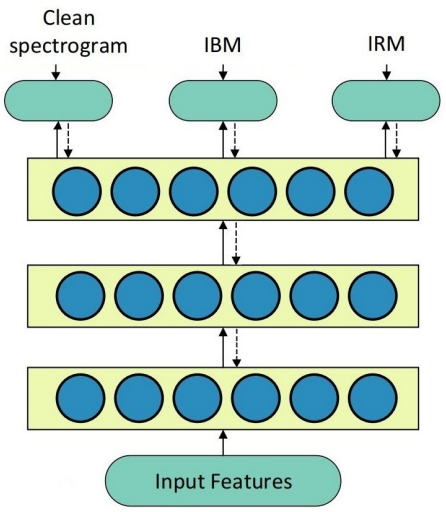


Figure 3: A multi-target framework [2].

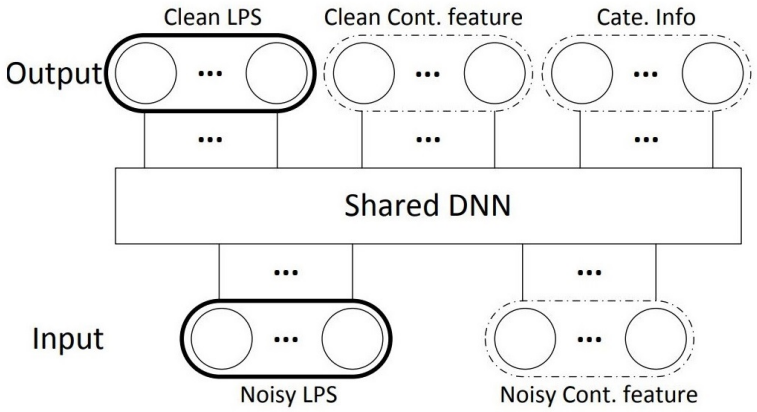


Figure 4: A multi-input and multi-target framework [3].

DNN-based architecture for joint optimization of all the parameters. The goal is to impose additional constraints to the network that are not available in a stand-alone mapping-based method. In this paper, it is demonstrated that joint log power spectrum and MFCC learning improves speech enhancement performance.

So far, we have reviewed some well-known mapping- and masking-based speech enhancement studies using the FC network. The more popular DNN-based speech enhancement methods became, the more advanced DNNs were studied in the field. In [4, 45, 46], RNN-LSTM network

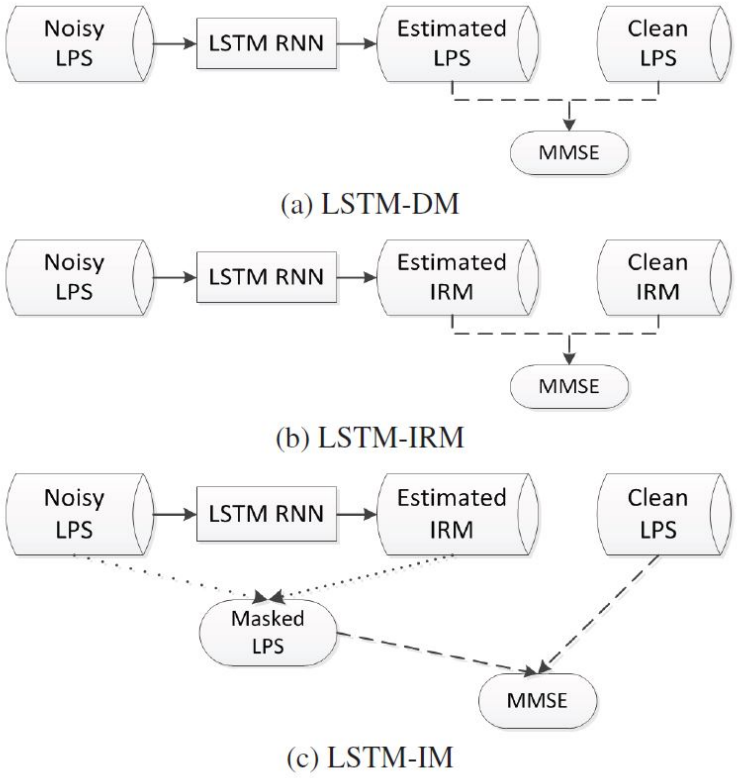


Figure 5: Block diagram of an LSTM-based speech enhancement framework with different training targets, (a) a mapping-based method with log power spectrum as the training target, (b) a masking-based method with IRM as the training target, a multiple-target approach combining IRM and log power spectrum [4].

was used for speech enhancement. The authors in [45] compared LSTM and FC networks to estimate a spectral mask and signal approximation, explained in Section 2.2.1. They compared an optimally designed three-layer FC with a two-layer LSTM and reported the superiority of LSTM for speech enhancement in their framework. The authors in [46] employed LSTM for speech enhancement in a noise-robust ASR application. They reported that LSTM as front-end processing remarkably improves speech recognition results. To be more precise, employing an LSTM to enhance speech before speech recognition leads to a 13.76% average word error rate improvement. In [4], an LSTM-based multi-target approach was proposed for speech enhancement, and the authors presented exciting conclusions. Fig. 5 shows the speech enhancement approaches studied in this work. In all three cases, the input is the log power spectrum of the noisy input

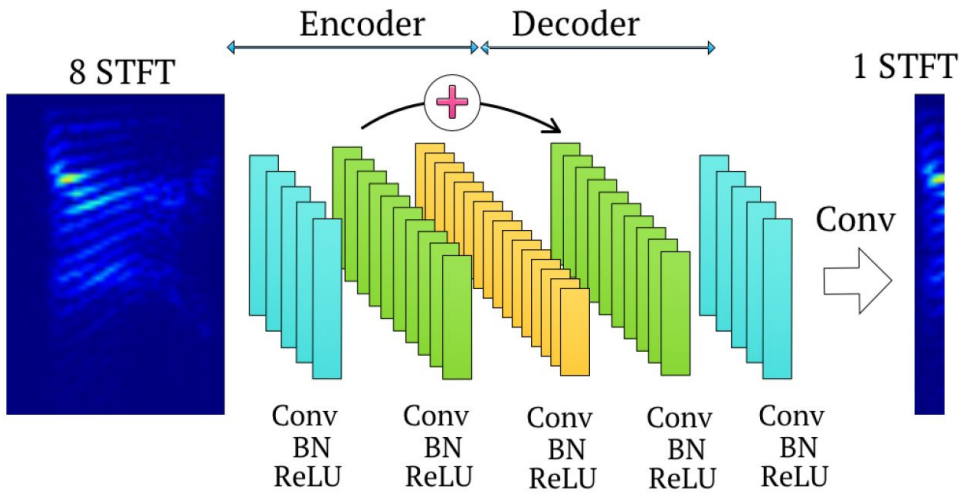


Figure 6: Block diagram of a mapping-based speech enhancement framework using redundant convolutional encoder-decoder [5].

speech while different training targets were used, including log power spectrum, IRM, and masked log power spectrum. Comparing the first and second frameworks, the authors observed that the mapping-based one yields better speech intelligibility at low SNR levels while the masking-based one outperforms at high SNRs levels. Thus, a new multiple-target joint learning framework was proposed to fully utilize the advantages of both masking- and mapping-based approaches that led to better speech enhancement results.

Besides, CNN has also been widely used in speech enhancement. Due to the weight sharing property of CNN, its number of parameters is less than FC and RNN, which is one of the reasons that CNN became popular for the researchers. Park *et al.* [5] employed CNN for the first time for speech enhancement in a mapping-based framework, shown in Fig. 6. They proposed a fully convolutional CNN (they called it Redundant Convolutional Encoder-Decoder) that does not have any FC layer; thus, it is very low complexity. Each layer in this DNN comprises a convolutional layer, a batch normalization layer, and a ReLU activation function layer. As shown in the figure, bypass connections are also added to facilitate training and improve performance. This encoder-decoder structure's input and training target are noisy and clean speech spectrogram, respectively. The encoder transforms a single channel input to high dimensional features, and the decoder then

transforms them to a clean speech spectrogram. This framework resulted in very good speech enhancement results while having low complexity. Many other CNN-based frameworks attempted to improve the speech enhancement performance in the frequency or time domain, such as [47–50].

1.3.2 Phase Enhancement

Most of the speech denoising methods focused on enhancing the speech magnitude solely and used the noisy phase to restore the estimated speech, thereby underestimating the impact of phase enhancement on the overall performance. Besides, the lack of clear structures in the phase spectrogram renders its estimation difficult, especially by DNN [44]. Nonetheless, the advantages of exploiting phase information for speech enhancement have been demonstrated in [51], where the authors showed that processing the phase spectrum along with magnitude can further improve perceptual speech quality and boost both objective and subjective enhancement results. Besides, the noisy speech phase follows the phase of background noise more than that of the target speech at negative SNR levels. In other words, phase enhancement is prominent, especially for low input SNR levels. Hence, phase enhancement has recently drawn researchers' attention [8, 51].

Gunawan et al. [52] introduced a method that iteratively estimates the time domain source signals in a mixture using multiple input spectrogram inversions given the corresponding estimated STFT magnitude responses. With the magnitude response as a constraint, the missing phase information is iteratively recovered through spectrogram inversion estimates. At each iteration, source estimates are updated utilizing the average error between the mixture and the sum of estimated sources.

In another study [53], an MMSE phase estimation method is presented to estimate the phase information for the signal reconstruction of sources from a single channel mixture observation, assuming given knowledge of signal spectrum amplitude. The pair of phases with the lowest group delay is chosen among several phase candidates, which are obtained from the minimization problem. The estimated phase and magnitude response are combined to reconstruct the sources.

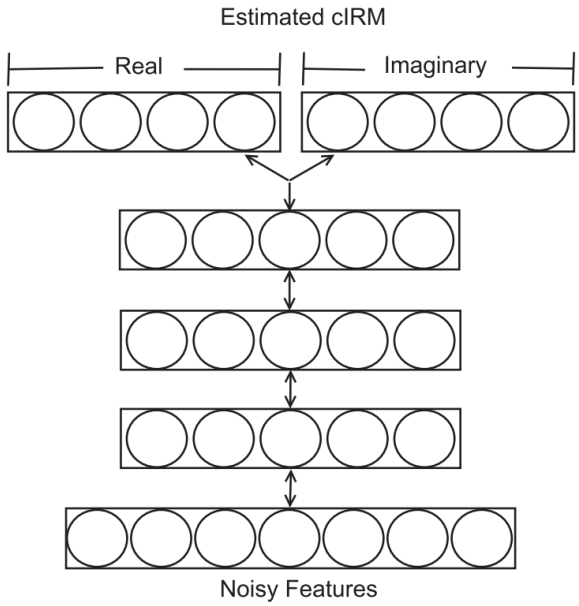


Figure 7: Schematic diagram of an FC-based architecture used to estimate the real and imaginary components of a ratio mask for simultaneous magnitude and phase enhancement [6].

Krawczyk et al. [54] introduced a method to use the noisy information and estimate the fundamental frequency to reconstruct the spectral phase between harmonic components across frequency and time, while unvoiced frames are left unchanged.

All the methods mentioned above focused on phase enhancement, and only a few works have considered simultaneous magnitude and phase enhancement. For example, Williamson et al. [6] introduced a DNN to jointly estimate the real and imaginary components of a complex ideal ratio mask defined in the complex domain, which can be considered a multi-target method, as shown in Fig. 7. The network output, i.e., the real and imaginary components of the mask, is to be multiplied by the noisy spectrogram of speech to enhance both the magnitude and phase of the noisy speech. More details about phase enhancement will be presented in Chapter 4.

1.3.3 Generalization to Unseen Conditions

For supervised speech enhancement, generalization to untrained conditions is crucial. In other words, the mismatch between training and the test conditions has to be addressed. Although DNNs could be successful in noise-independent speech enhancement, they have limitations in modeling

a large number of speakers. In this sense, generalization concerns three main aspects, namely generalization to unseen noise types and different SNR levels, as well as generalization to many speakers [7, 55].

The first generalization is for unseen SNR levels. Regarding [56], although experiments demonstrated that the supervised speech enhancement is not sensitive to precise SNR levels, various SNR levels can be embedded in the training set to tackle the generalization in terms of SNR. The reason for this SNR level independence is that the local SNRs in the TF unit level usually vary over a wide range which leads to the necessary range of SNR levels that should be provided for the generalization task.

The second aspect of generalization involves unseen noise types. For mapping-based algorithms like [1], a noise-aware training is proposed in [57] which means the input feature vector includes an explicit noise estimate. It is shown that a DNN with noise-aware training can be better generalized to accommodate the noise types which had not been included in the training dataset. Moreover, for the masking-based methods, the authors in [58] proposed that DNN can be trained to estimate IRM at the frame level. Moreover, IRM is simultaneously estimated over several consecutive frames, and different estimations for the same frame are averaged to produce a smoother and more accurate mask [7]. Wang et al. in [42] demonstrated that the issue of generalization to unseen noise types could be significantly tackled through large-scale training with a wide variety of noises.

Finally, the speech enhancement task must not be affected by various speakers. According to [59], an FC network cannot model numerous speakers if the speaker generalization is aimed to be addressed by training with various speakers. Chen et al. in [55] proposed to use LSTM and large-scale training with many speakers to overcome this problem.

1.4 Motivations and Objectives of the Research

1.4.1 Motivations

Nowadays, the market of applications that require speech and audio processing is rapidly growing. For instance, the market share of only smart speakers was valued at 11.9 billion U.S. dollars in 2019, while projections suggest that this annual figure could grow to over 35.5 billion U.S. dollars by 2025 [60]. This sheds light on the necessity of research about high performance while low-cost algorithms in speech processing. Although speech enhancement has been vastly studied over the past decades, there is still a big room for improvement and several uncertain and unresolved issues. In the following, we summarize the motivation behind this study from different perspectives.

Traditional vs. DNN-based methods: As discussed in Section 1.2, traditional speech enhancement methods that are unsupervised and rely on statistical models cannot achieve satisfactory results in real-world environments. That is because these models are based on unrealistic assumptions such as noise and speech being uncorrelated or stationary.

Nowadays, deep learning as a primary tool to develop data-driven information systems has led to revolutionary advances in numerous areas, including speech enhancement. In this context, speech enhancement is treated as a supervised learning problem, which does not rely on any prior assumption about the statistical properties of speech and noise, nor suffers from the above issues faced by traditional methods. This enables DNN-based speech enhancement methods to handle highly non-stationary noises in real-world scenarios. Furthermore, the non-linearities of DNN empower it in modeling highly complicated transformations between noisy and clean speech. As such, a major motivation of this work is to investigate advanced supervised methods for speech enhancement. As mentioned before, the desired supervised learning problem has three main components, features, training target, and learning machine. Thus, we investigate all these components to end up with an appropriate DNN-based framework for speech enhancement.

Magnitude solely vs. Magnitude and Phase Enhancement: As emphasized in Section 1.3.2,

most of the research about speech enhancement in the literature focused on magnitude enhancement and ignored processing the phase due to lack of a clear structure in its spectrum and difficulty of its estimation. Given DNN's powerful learning ability, further improvement of the speech enhancement performance can be achieved by processing the phase alongside magnitude. Recently, some studies attempted to embed phase processing as a part of their speech enhancement algorithm by including phase information in a spectral mask, a complex spectrogram, or a derivation of the phase itself. Although these studies have yielded good speech enhancement results, there are still limitations and shortcomings with these methods. For instance, a complex spectrogram that includes both magnitude and phase has been considered as the training target in some studies, while it is shown that it fails to deal with unseen conditions. Hence, the advantage of simultaneously processing phase and magnitude that leads to a further enhancement from one side and the shortcomings of the existing methods from another side motivate us to conduct research phase processing using DNN-based algorithms.

Complexity vs. Performance: To design a speech enhancement algorithm that is suitable for real-word applications, we have to consider not only the objective measurements, including quality and intelligibility of the processed speech, but also the cost to achieve that performance. There are a few concern about the model complexity that has to be taken into account while designing an speech enhancement algorithm. Depending on the application, speech enhancement can be accomplished online or on the edge (offline). The former case allows having almost no limitation about the model complexity since very powerful machines can be used to take care of very fast computations on the cloud; thus, the performance will be outstanding. However, the latter case, which is most common, dictates the limitations that we have to take care of while designing the algorithm. In order to have a DNN-based system that performs on the edge, low-complexity design is as important as the performance because it affects both cost and size of the product. It has to be noted that the DNN performing speech enhancement is supposed to be part of a bigger model, i.e., the complexity of a system is the summation of those of different components. Hence, our desired DNN has to be as low-complexity as possible. It is worth pointing out that complexity includes

two aspects: the number of trainable parameters of the model and the computational complexity. As such, not only the model has to contain a small number of parameters, but also it has to have low computations.

Besides, the higher the complexity, the more the computational time will be. In other words, a high complexity model causes high computational latency that is not tolerable in many applications such as hearing aids.

There are many DNN-based speech enhancement algorithms available in the literature; but, they work well only offline because they are either high complexity models or their computations are too high that cannot be accomplished on the edge. Furthermore, some of the available methods suffer from large latency which can be as long as several seconds, which is not acceptable in many real-word applications. Consequently, one of the major motivations of this work is to develop DNN-based speech enhancement algorithms that are of low-complexity and low-latency while achieving satisfactory performance.

1.4.2 Objectives

When treating speech enhancement as a supervised learning problem, there are three main components, i.e., input features, training target, and learning machine, that have to be appropriately designed so that the maximum performance is achieved while keeping the model complexity low. As such, the main objectives of this work are summarized as follows:

- The input of DNN in a supervised speech enhancement algorithm has to be discriminative so that the learning machine can make a proper transformation of it to the desired training target. Many acoustic features have been introduced in the literature as the input of DNN, and some studies even considered a combination of them as the input. However, the best features that suit a DNN-based method are the Gammatone-domain ones extraction of which takes a long time and requires a huge deal of computations. Furthermore, these features do not fit well the DNN-based speech enhancement framework. To this end, we suggest DNN itself performs the feature extraction so that the most suitable and discriminative features are extracted.

Hence, we employ a fully convolutional and low-complexity CNN with some modifications to maximize its performance as the feature extractor. Besides, since speech contains strong temporal dependencies, we adopt an LSTM to transform the features extracted by CNN to the training target while benefiting from temporal contextual information of the speech. Moreover, as the training target, the phase information is embedded in a spectral mask called phase sensitive mask so that the speech phase is partially processed along with magnitude. The whole model complexity is relatively lower than most speech enhancement methods in the literature while it takes care of not only transformation but also feature extraction.

- Since the main focus of this work is to enhance phase along with magnitude using a very low complexity model, we propose a composite model where the components perform in parallel to accelerate the process, and it requires very low computational resources. A modified CNN with attention mechanisms and LSTM with grouping strategy are employed to lower the complexity while enhancing the performance and speeding up the process. Besides, a multi-target strategy is adopted where a spectral mask as a subtask for magnitude enhancement and a phase derivative as another subtask for phase enhancement are considered. Different phase derivatives are investigated specifically for phase enhancement in this framework. The advantages of this model over some well-known DNN-based speech enhancement methods are demonstrated through extensive comparative experiments. It is worth mentioning that this work presents one of the simplest models in the literature with the smallest amount of computations and memory footprint.

- Both methods mentioned above are masking-based methods performing in the STFT domain that comprises complex values and requires processing phase and magnitude separately. However, performing speech enhancement in modified discrete cosine transform (MDCT) domain leads to dealing with only real values that contain the whole speech information. Moreover, the MDCT domain saves calculations since the network deals with only real magnitude values, not both magnitude and phase. The employed DNN in this

work is fully convolutional, allowing parallel computation that makes processing very fast. In addition, we decreased the latency by almost 45% in this model compared with the two previous models. Further and even more importantly, we add a speech classifier as a pre-processor before the main DNN. The idea is based on the fact that speech enhancement is mainly affected by SNR level and gender. Hence, we use a simple speech classifier to classify the input speech into male with high SNR, female with high SNR, male with low SNR, and female with low SNR. Then, a very low complexity DNN takes care of the enhancement.

1.5 Organization

The thesis organization is as follows. Detailed background about deep learning hyperparameters, the prevalent datasets, and evaluation metrics will be presented in Chapter 2. In Chapters 3 and 4, two masking-based methods that are based respectively on serial and parallel models, both performing in STFT domain, will be presented. In Chapter 5, a mapping-based method in the MDCT domain is presented, and finally, Chapter 6 concludes the thesis. The structure of this thesis is detailed below.

Chapter 2 provides the background information in which the noisy speech model is first explained, and the equations are presented in both time and STFT domain. Then, Section 2.3 provides detailed explanations about DNN hyperparameters that have to be decided during the model design. Finally, the clean and noise datasets and the evaluation metrics are presented in Section 2.4.

Speech enhancement using a serial hybrid DNN is proposed in Chapter 3. The high-level block diagram of this model is presented in Section 3.2. Next, the details about different training targets for this model are explained in Section 3.2.3. Afterward, the modified CNN and RNN are presented in Sections 3.2.1 and 3.2.2, respectively. Finally, the experimental results, analysis, and a wide range of comparisons are provided in Section 3.3.

A low-complexity and phase-aware parallel DNN for Speech Enhancement is proposed in

Chapter 4. The high-level block diagram of this model is provided in Section 4.2. The composite model structure is discussed in Section 4.2.1. The training target of this model is comprised of a spectral mask and a phase derivative that is expressed in Section 4.2.2. Next, the clean speech is reconstructed using the estimated mask and phase derivative in Section 4.2.3. The detailed model architecture and experimental results are finally presented in Section 4.2.4 and 4.3, respectively.

A mapping-based model in the MDCT domain for speech enhancement is presented in Chapter 5. The transformation from time to the MDCT domain and then the reconstruction of time-domain speech based on MDCT values are explained in Sections 5.2.1. The speech classifier along with different classification approaches is explained in 5.2.2. Section 5.2.3 explains the fully-convolutional expert DNNs architecture. The system description is then provided in Section 5.2.4. Finally, the experimental results and comparisons are provided in Section 5.3.

In Chapter 6, the concluding remarks that highlight these thesis contributions are presented. Moreover, the three models presented in the previous chapters are compared in many respects. Finally, the plausible future work beyond this thesis research will be presented.

Chapter 2

Background

The background for DNN-based speech enhancement is presented in this chapter. First, the noisy speech is mathematically modeled. Then, a DNN-based speech enhancement frame work is introduced, including input speech features, output training targets and popular neural network models as learning machines. Next, hyper parameters of DNN that we frequently refer to throughout this thesis are expressed. Finally, several prevalent databases that are commonly used for testing and training, and evaluation metrics that we used in this thesis are introduced.

2.1 Noisy Speech Model

The target speech is usually corrupted by background noise which can be broadly divided into two main categories of additive noise from other sound sources and convolutive noise from surface reflections. As the former is the most common factor which degrades speech quality, the noisy speech is modeled in the time-domain as follows,

$$y(t) = x(t) + n(t) \tag{1}$$

where $y(t)$, $x(t)$, and $n(t)$ denote the noisy observation, clean speech, and noise at time t, respectively. There is a common assumption that speech and noise are statistically independent of each

other [8][61].

STFT is usually employed to model the noisy speech in the transform domain. It describes the sinusoidal frequency and phase content of local sections of a signal as it changes over time, where the harmonic structure of the speech can be clearly distinguished [62]. After sampling the input speech signal in discrete-time, by segmentation (framing), windowing, and applying FFT, we have the following complex-valued model for the noisy speech in the TF domain.

$$Y(k,l) = X(k,l) + N(k,l) \tag{2}$$

where $Y(k,l)$, $X(k,l)$ and $N(k,l)$ are STFT of the noisy speech, clean speech, and the noise signal, at frequency bin l and time frame k [61].

There are two expressions for $Y(k,l)$. The first one is polar coordinates, i.e., magnitude and phase, which is used to enhance STFT of a noisy speech. It is defined as follows,

$$Y_{k,l} = |Y_{k,l}|\, e^{i\angle Y_{k,l}} \tag{3}$$

where $|Y_{k,l}|$ and $\angle Y_{k,l}$ represent the magnitude and phase response of STFT at time k and frequency l. Each TF unit in the STFT representation is a complex number with real and imaginary components by which the magnitude and phase responses are computed below:

$$|Y_{k,l}| = \sqrt{\Re\left(Y_{k,l}\right)^2 + \Im\left(Y_{k,l}\right)^2} \tag{4}$$

$$\angle Y_{k,l} = tan^{-1}\frac{\Re\left(Y_{k,l}\right)}{\Im\left(Y_{k,l}\right)} \tag{5}$$

The other expression is the definition of $Y(k,l)$ with Cartesian coordinates using the expansion of the complex exponential as given below,

$$Y_{k,l} = |Y_{k,l}|\, cos\left(\angle Y_{k,l}\right) + i\,|Y_{k,l}|\, sin\left(\angle Y_{k,l}\right) \tag{6}$$

$$\Re\left(Y_{k,l}\right) = |Y_{k,l}|\, cos\left(\angle Y_{k,l}\right) \tag{7}$$

$$\Im\left(Y_{k,l}\right) = |Y_{k,l}|\, sin\left(\angle Y_{k,l}\right) \tag{8}$$

Clearly, these real and imaginary spectrograms are the same except for a shift of $\pi/2$ radians.

2.2 DNN-based Speech Enhancement Framework

A high-level block diagram of a DNN-based speech enhancement framework is shown in Fig. 8. There are two main stages: training and testing. In the training stage, the features of noisy input speech are extracted and fed to the DNN. Also, the training targets are calculated based on clean and noisy speech and set as the output of the DNN. Then, the network will be trained using the input features and training targets, i.e., the DNN learns the relationship between its input and output. In the training stage, the features of noisy input speech are calculated and fed to the well-trained DNN that is supposed to estimate the corresponding training target. Finally, the clean speech will be reconstructed using the estimated training target. As shown, there are three main components that have to be carefully designed: input features, training targets, and the DNN. The more discriminative the input features are, the less demand on the DNN to successfully perform the desired testing/estimation. Moreover, an appropriately chosen training target can boost the learning and generalization capabilities of the model in unseen conditions [7]. Furthermore, each DNN type possesses specific attributes. Thus, DNN has to be accurately designed considering its input and output for the specific task of speech enhancement. Hence, we can divide the development of a DNN-based speech enhancement algorithm into two subtasks: choosing an appropriate set of input features and training targets and designing a suitable DNN. These two subtasks are detailed below.

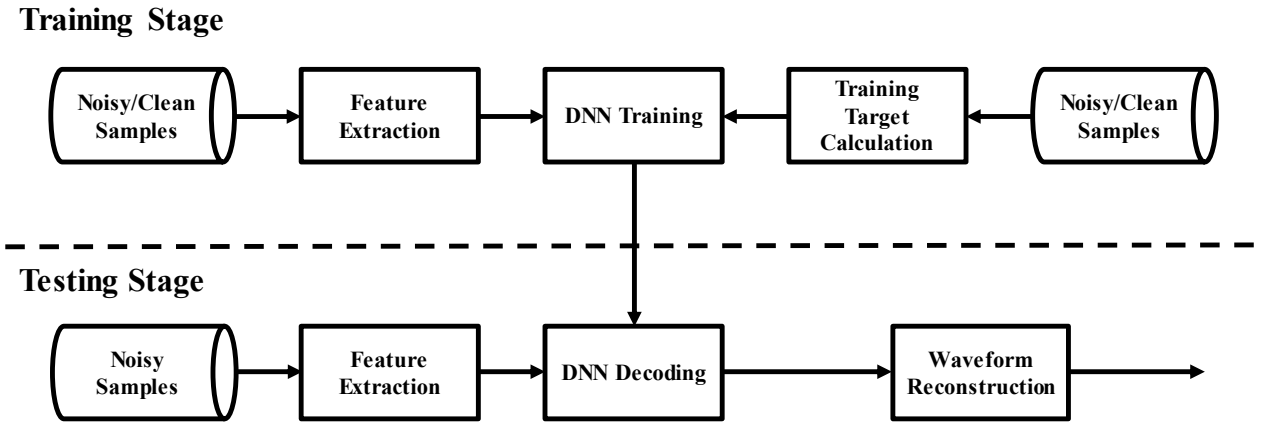


Figure 8: High-level block diagram of a DNN-based speech enhancement framework

2.2.1 Input Features and Output Training Targets

Input Features

Selecting proper discriminative features can greatly impact the DNN-based speech enhancement performance. Many acoustic features have been introduced in the literature for the task of speech enhancement. In [17], an extensive survey is conducted to evaluate and compare a list of acoustic-phonetic features for speech enhancement at low signal-to-noise ratio (SNR) inputs. The comparison was carried out using a FC network and the enhancement performance is evaluated using standard objective speech intelligibility metrics. The features investigated in this study can be categorized as follows.

- Mel-domain features:

  mel-frequency cepstral coefficient (MFCC),

  log mel-spectrum feature (LOG-MEL),

  delta-spectral cepstral coefficient (DSCC).

- Gammatone-domain features:

  gammatone feature (GF),

  gammatone frequency modulation coefficient (GFMC),

gammatone frequency cepstral coefficient (GFCC).

- Linear prediction features:

  perceptual linear prediction (PLP),

  relative spectral transform PLP (RASTA-PLP).

- Zero-crossing feature:

  zero-crossings with peak-amplitudes (ZCPA)

- Medium-time filtering features:

  power normalized cepstral coefficients (PNCC),

  suppression of slowly-varying components and the falling edge of the power envelope (SSF).

- Autocorrelation features:

  relative autocorrelation sequence MFCC (RAS-MFCC),

  phase autocorrelation MFCC (PAC-MFCC),

  autocorrelation sequence MFCC (AC-MFCC).

- Pitch-based feature:

  Time-frequency features based on pitch tracking (PITCH).

- Modulation domain features:

  Gabor filterbank (GFB),

  amplitude modulation spectrogram features (AMS).

- Time-domain feature:

  raw waveform (WAV).

- Spectral magnitude feature:

  log spectral magnitude (LOG-MAG).

- Multi-resolution feature:

  Multi-Resolution Cochleagram (MRCG).

Table 1: Comparison of different feature complexity [17].

Feature	Dimension	Frame size (ms)	Extraction time (ms/frame)
AC-MFCC	31	20	2.625
AMS	15	32	0.160
GFB	311	25	1.592
GF	64	20	12.768
GFCC	31	20	13.192
GFMC	31	20	15.234
LOG-MAG	161	20	0.048
LOG-MEL	40	20	0.027
MFCC	31	20	0.030
MRCG	256	420	13.475
PAC-MFCC	31	20	0.086
PITCH	384	10	76.337
PLP	13	20	0.282
PNCC	13	25.6	11.993
RAS-MFCC	31	20	2.332
RASTA-PLP	13	20	0.324
SSF-I	31	50	1.487
SSF-II	31	50	1.480
WAV	320	20	0.000

In [17], an interesting comparison is presented to show the computational complexity of the above-mentioned features. They compared different features in terms of their dimension per time frame, the frame size (milliseconds), and the extraction time (millisecond per frame), as shown in Table 1. It is worth mentioning that a Dell OptiPlex 780 PC with a quad-core processor at 2.66 GHz and 8 GB RAM is used to perform this comparison. The authors in [39] divided these features into highly engineered modulated features and low engineered plain features based on their computations. Hence, considering Table 1, features like MRCG and LOG-MAG fall into the former and latter groups, respectively, regarding complexity of their computations, where Gammatone-domain features take a long extraction time compared to spectrum-based ones.

Furthermore, Table 2 shows a comparison of the above-mentioned features in terms of the percentage of short-time objective intelligibility (STOI) improvement averaged on a set of testing

Table 2: Comparison of different features in terms of STOI improvement (%) averaged on a set of test noises. *Sim. Room Impulse Responses* and *Sim. RIRs* denote simulated and recorded reverberation, respectively [17].

Feature	Matched noise			Unmatched noise		
	Anechoic	Sim. RIRs	Rec. RIRs	Anechoic	Sim. RIRs	Rec. RIRs
MRCG	**7.12**	**14.25**	**12.15**	**7.00**	7.28	8.99
GF	6.19	13.10	11.37	6.71	7.87	8.24
GFCC	5.33	12.56	10.99	6.32	6.92	7.01
LOG-MEL	5.14	12.07	10.28	6.00	6.98	7.52
LOG-MAG	4.86	12.13	9.69	5.75	6.64	7.19
GFB	4.99	12.47	11.51	6.22	7.01	7.86
PNCC	1.74	8.88	10.76	2.18	**8.68**	**10.52**
MFCC	4.49	11.03	9.69	5.36	5.96	6.26
RAS-MFCC	2.61	10.47	9.56	3.08	6.74	7.37
AC-MFCC	2.89	9.63	8.89	3.31	5.61	5.91
PLP	3.71	10.36	9.10	4.39	5.03	5.81
SSF-II	3.41	8.57	8.68	4.18	5.45	6.00
SSF-I	3.31	8.35	8.53	4.09	5.17	5.77
RASTA-PLP	1.79	7.27	8.56	1.97	6.62	7.92
PITCH	2.35	4.62	4.79	3.36	3.36	4.61
GFMC	-0.68	7.05	5.00	-0.54	4.44	4.16
WAV	0.94	2.32	2.68	0.02	0.99	1.63
AMS	0.31	0.30	-1.38	0.19	-2.99	-3.40
PAC-MFCC	0.00	-0.33	-0.82	0.18	-0.92	-0.67

noises. It is worth noting that the authors compared these features under three different conditions: no reverberation, simulated reverberation (indicated by *Sim. RIRs*), and recorded reverberation (indicated by *Sim. RIRs*). The highest average score achieved by MRCG features are due to the fact that they are built using high-quality GF features with additional contextual information that improves the speech enhancement performance.

To take the advantage of some of the aforementioned features together, [63] proposed a complementary set of features through feature selection using group Lasso. This complementary set, which have been used in many studies, comprises AMS, RASTA-PLP, MFCC, and PITCH, and it is demonstrated that this feature set can boost performance of speech enhancement.

Although these comparisons could give a good idea about the performance of a DNN-based speech enhancement method using different speech features, we have to note that these comparisons are made based on a FC network as the DNN and a spectral mask as the training target. Hence, the result would differ if using other DNN types and training targets. Consequently, the optimal choice of input features depends on not only the quality of the features themselves, but also other components of the system.

Training Targets

Training target, or the objective function, is estimated by DNN given the input features. There are two main types of training targets in the literature, namely masking-based and mapping-based. The former describes time-frequency relationships of clean speech and background interference, and the latter corresponds to a spectral representation of clean speech. In the literature, several types of mapping and masking based training targets are introduced [1, 6, 45, 64–69]. We present two examples of each types in this section.

The very first masking-based training target was the ideal binary mask (IBM) through which speech intelligibility for both normal-hearing and hearing-impaired people is dramatically improved [64]. The IBM is defined on spectrogram or cochleagram of a noisy speech where it assigns 1 to a TF unit if the SNR within this unit exceeds a certain local criterion and 0 otherwise. Despite the very good performance of IBM, it labels each TF unit sharply. So, ideal ratio mask (IRM) was introduced, as the soft version of IBM, by dividing clean over noisy speech power spectrogram, with the assumption that speech and noise are uncorrelated [65]. It is shown that IRM leads to better speech ineligibility than IBM. The estimated mask by DNN is then multiplied by the input noisy speech spectrogram to reconstruct the clean speech magnitude one. The clean speech is then reconstructed using the estimated magnitude and the phase of noisy input speech.

Furthermore, some mapping-based training targets have been introduced in the literature. Gammatone frequency target power spectrum (GF-TPS) is a mapping-based training target which is defined on a cochleagram based on a gammatone filterbank [66]. Target magnitude spectrogram

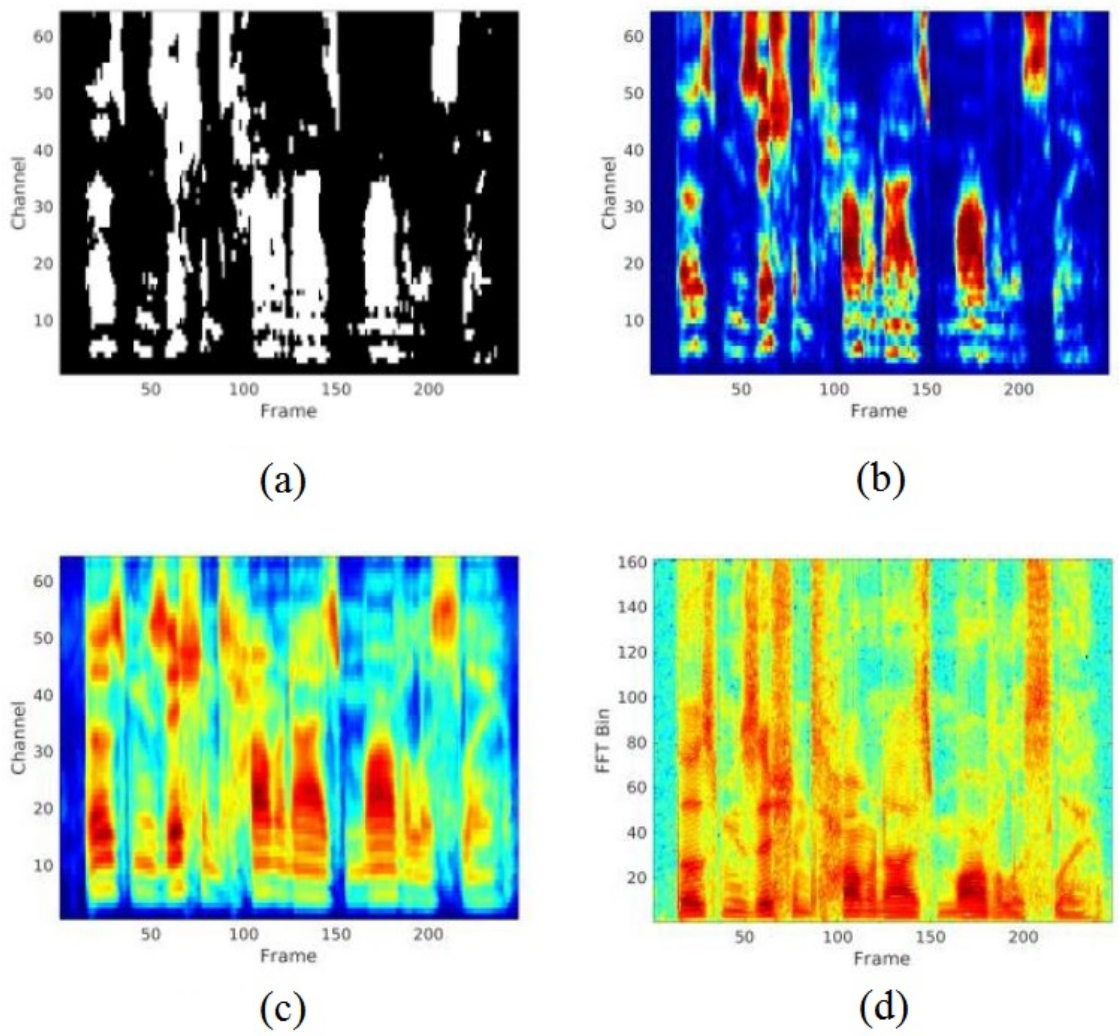


Figure 9: An illusion of different training targets for an utterance mixed with a factory noise at SNR level of -5 dB, (a) IBM, (b) IRM, (c) GF-TPS, (d) TMS [7].

(TMS) is another mapping-based training target that is defined on a spectrogram where the machine aims to estimate the magnitude spectrogram of the clean speech from the noisy input speech [57]. The estimated training target is then directly transformed to the clean speech.

In [45], a combination of masking- and mapping- based training targets is introduced as signal approximation (SA) seeking to maximize segmental SNR (SSNR). The goal of SA is to train a ratio mask estimator that minimizes the difference between the spectral magnitude of clean speech and that of the estimated one. SA training target achieves better a good enhancement performance. An illusion of the above-mentioned training targets for an utterance mixed with a factory noise at

SNR level of -5 dB is shown in Fig. 9.

To comprise mapping and masking based training targets, it is generally concluded that, on the one hand, for higher input SNR and the purpose of dereverberation, masking-based training targets perform better. And, on the other hand, for lower input SNR and denoising purposes, mapping-based training targets give higher performance in terms of speech intelligibility [4]. Furthermore, according to performance comparisons in [7], IRM mask and spectral magnitude mask (SMM) map emerge as preferred training targets. The argument made in these papers is based on experiments that are conducted using a simple FC network and specific input features, while the conclusion may differ when changing the system components, like network architecture, the complexity of the model, the input features, etc. Hence, the superiority of a training target depends on all the model components.

2.2.2 Common DNNs as the Learning Machine

Typically, a DNN comprises multiple layers, each containing components, like neurons, weights, biases, and activation functions, the values of which change according to build an appropriate transformation function between DNN input and output. Changing the arrangement of these components and the computation style leads to different DNN architectures with various attributes. The most common DNNs in the literature are fully-connected neural network (FC), recurrent neural network (RNN), convolutional neural network (CNN), and generative adversarial network (GAN). Each of these DNNs exhibits specific attributes and have their advantages and limitations. In the following, we briefly explain each DNN type along with some well-known related studies where these DNNs are adopted as the learning machine.

Fully-Connected Neural Network

Fully-connected neural network, which we call FC for simplicity, is the most popular class of DNNs that has been commonly used in many applications including speech enhancement. FC network consists of an input and output layer as well as several hidden layers between them. An

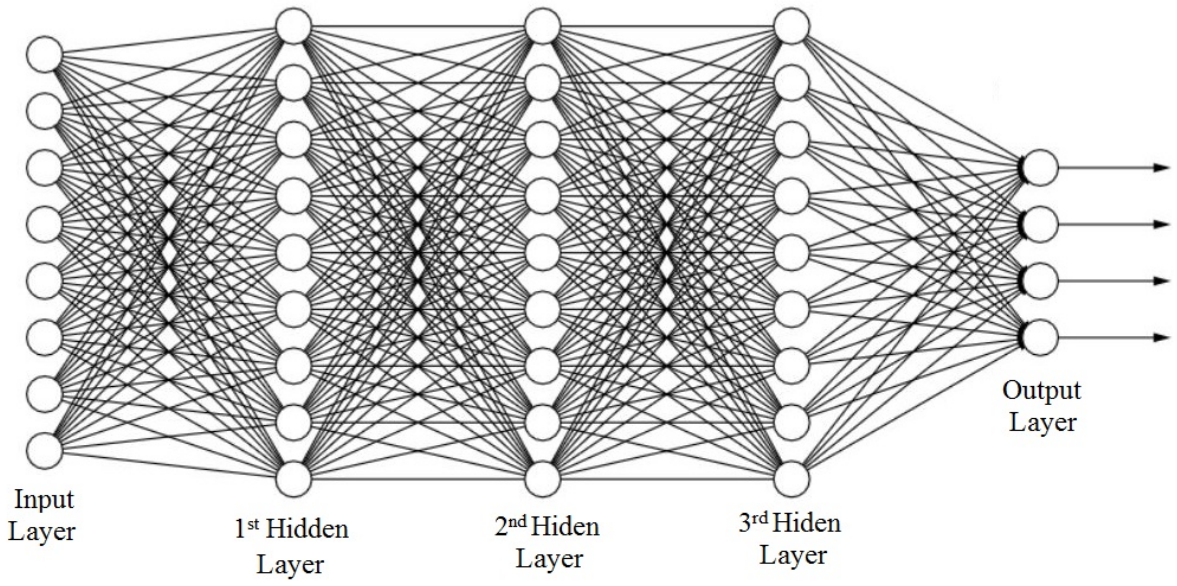


Figure 10: An FC network with three hidden layers.

example of a three-layer FC network is shown in Fig. 10. Each current layers' input is the output of its previous layer. All the neurons between two consecutive layers are connected to each other, and that is why it is called fully-connected network. Since DNN must be highly non-linear to be able to model any transformation function, an activation function is adopted in every neuron, which provides DNN with the required non-linearity (different activation functions will be detailed in the next chapter).

The parameters, i.e., weights and biases, of such a network are adjusted using the classic back-propagation method which aims to minimize the prediction error between the actual and estimated training target values through a gradient descent algorithm [7]. It is worth mentioning that very deep FC networks have suffered from the vanishing problem in their training process. Vanishing problem happens when the input and output layers are far from each other, i.e. there are many hidden layers between them, which makes the absolute value of gradient become smaller progressively during the backpropagation. This causes the layers close to the input layer not to be effectively modified, i.e. the training is not accomplished properly. To solve this problem, restricted Boltzman machine is first introduced to pre-train a DNN before the subsequent supervised training [70]. Later, the ReLU activation function was replaced with the traditional sigmoid activation function

showing that the moderately FC network can be effectively trained without any pre-training [71]. Also, in order to facilitate training, He et al. [72] introduced skip connection technique for very deep FC networks where the input of some layers close to input are directly connected to those close to the output layer. This way, the gradients find a shortcut to get to the layers close to the input layer so that they are being trained.

Although FC network is a very popular DNN as a powerful learning machine for the regression and classification purposes, it has its own limitations. One of the most important drawbacks of a FC network is its high complexity. The issues with a high-complexity model will be discussed later in detail; but in short, a high complexity model cannot be implemented in many small applications such as smart living assistants. Even if the manufacturer accepts the cost of embedding huge DNNs in the device, these DNNs have high computational complexity that cannot be accomplished on the edge, i.e., the computations must be done on the cloud, thus the device cannot perform offline. An simple example is the speech-to-text application on mobile phones that does not work offline. Furthermore, such a DNN suffers from high computational time that is not tolerable in many applications such as real-time speech communication and hearing aids, where in the latter case, even a very short latency can be noticeable to a wearer.

Recurrent Neural Network

The other class of DNNs is called recurrent neural network (RNN) that has been vastly studied in the speech enhancement field because of its capability in modeling temporal information of the speech. The word *recurrent* refers to feedback connections formed among RNN neurons which are associated with a time-delay operation. This operation gives rise to a memory structure in the RNN, which allows it to model temporal dynamics of the input data. In effect, due to RNN temporal unfolding, it can be considered as a DNN with several layers depending on the length of input data varying from one to infinity [73, 74].

RNN can easily replace FC with no significant modification since they both have similar usage in speech enhancement. FC network treats input samples independently and is unable to relate

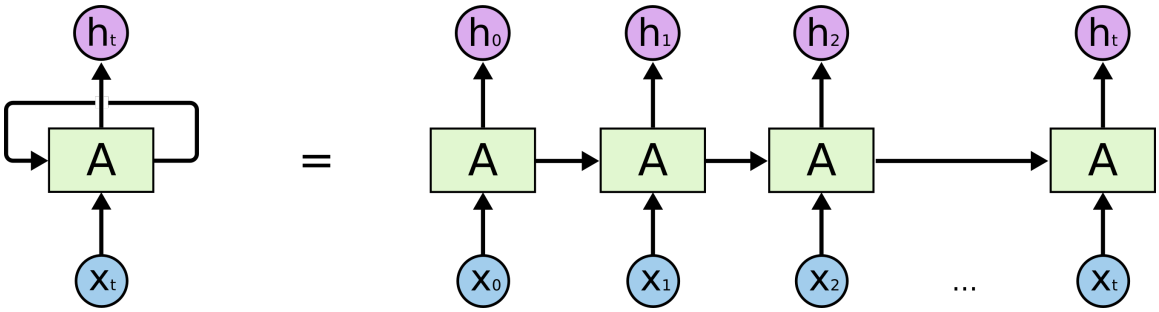


Figure 11: Unrolling an RNN, X_t and h_t denote input and hidden state at time step t.

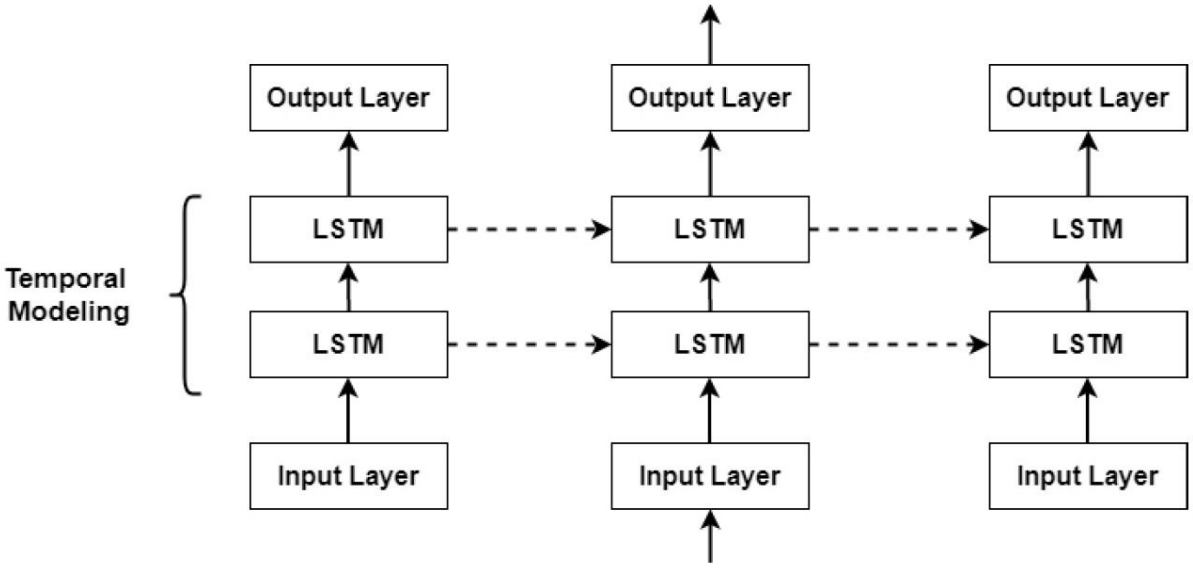


Figure 12: An example of an LSTM network. The information are passing through time, as well as passing from the input to output layer [8].

contiguous frames to each other. As a common practice hence, several input time frames are provided at once to the FC network so that it somehow captures temporal dependency between speech current and previous time frames. However, RNN using its memory attribute is enabled to exploit the temporal information; thus, it requires only one speech time frame as input to model the contextual relation between consecutive time frames. In other words, RNN is able to treat input speech time frames as a sequence, though it is fed by only one single-frame input, and models the changes over time which makes it a natural choice for modeling the temporal dynamics of speech [75].

The learning method for RNN is backpropagation through time (BPTT) where the recurrent

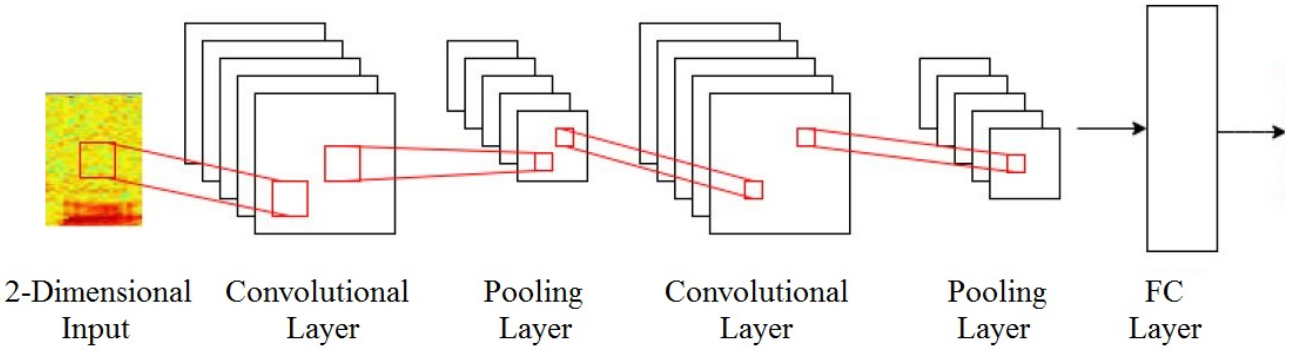


Figure 13: An FC network with three hidden layers [8].

model is treated as an FC with an unlimited number of layers, as shown in Fig. 11. BPTT conceptually works by unrolling all input time steps. Although training RNN with BPTT looks straightforward, there are well-known problems of exploding and vanishing gradients introduced in [76]. The former refers to the exponential growth of the gradient norm due to the explosion of the long-term dependencies, while the latter refers to the exponentially fast decrease of those dependencies [77]. Thus, LSTM was introduced to solve these problems that stop RNN from learning long-term dependencies. To build an LSTM network, as shown in Fig. 12, the neurons in RNN are replaced with LSTM units containing a memory cell with gates, facilitating the information flow over time, avoiding very long term dependencies. In the LSTM block, there are one memory cell and three gates where the forget gate controls how much previous information should be erased from the cell, and the input gate controls how much information should be added to the cell. The details of an LSTM unit and a DNN built up of them will be discussed later. To emphasize the temporal modeling in an LSTM network, the dash lines in Fig. 12 show that the information passes along the time dimensions, in addition to the information pathway from the input to output layer.

Convolutional Neural Network

Convolutional neural network (CNN) is the other DNN class that is well suited for pattern recognition. The CNN structure is highly customizable, allowing researchers to develop an extensive range of designs. An example of a traditional CNN is shown in Fig. 13, a so-called two-layer CNN.

CNN is made up of a cascade connection of pairs of a convolutional layer and a sub-sampling layer. Convolutional layers, comprising several feature maps, learn to extract local features with many filters, or called kernels interchangeably, which are applied in a sliding window across the entire input through weight sharing. It means that the neurons within the same module are constrained to have the same connection weights despite their different receptive fields. Multiple convolutions are performed on the input using different kernels resulting in disparate feature maps. Convolutional layers are usually followed by a sub-sampling layer, such as max-pooling, average pooling, or a combination of them, which performs a down-sampling operation along the spatial dimensions leading to reduced resolution and sensitivity to local variations. Conventionally the pair of a convolutional and sub-sampling layer is considered as one layer. The role of these pairs is to exploit high-level features of the input. Finally, an FC network receives the information extracted by its previous layers to perform the final regression or classification [7, 8, 78]. More details about CNN and its new architectures will be presented in the next chapter.

Generative Adversarial Network

Generative adversarial networks (GANs) have been recently introduced as generative models that can be supervised or unsupervised. GAN has drown researchers' attention due to its capability in synthesizing convincing images when it is trained with a dataset of natural images [79]. GAN aims to map samples from some prior distribution to the ones in another distribution of actual image or audio training data. There are two main components in a GAN structure, a generative model G and a discriminative model D, which are simultaneously trained. The main task of G is to generate samples that are as similar as possible to the training data through imitation of real data distribution, while D, which is usually a binary classifier, discriminates between the generated samples by G and the target ones from the training data. The goal of D is to distinguish the real data from dataset and the fake ones generated by G. This framework is similar to a two-player adversarial game where G, in the training stage, tries to lean an accurate mapping to generate samples similar to those in the real dataset so as to fool D, while D is also getting better at distinguishing the real data from

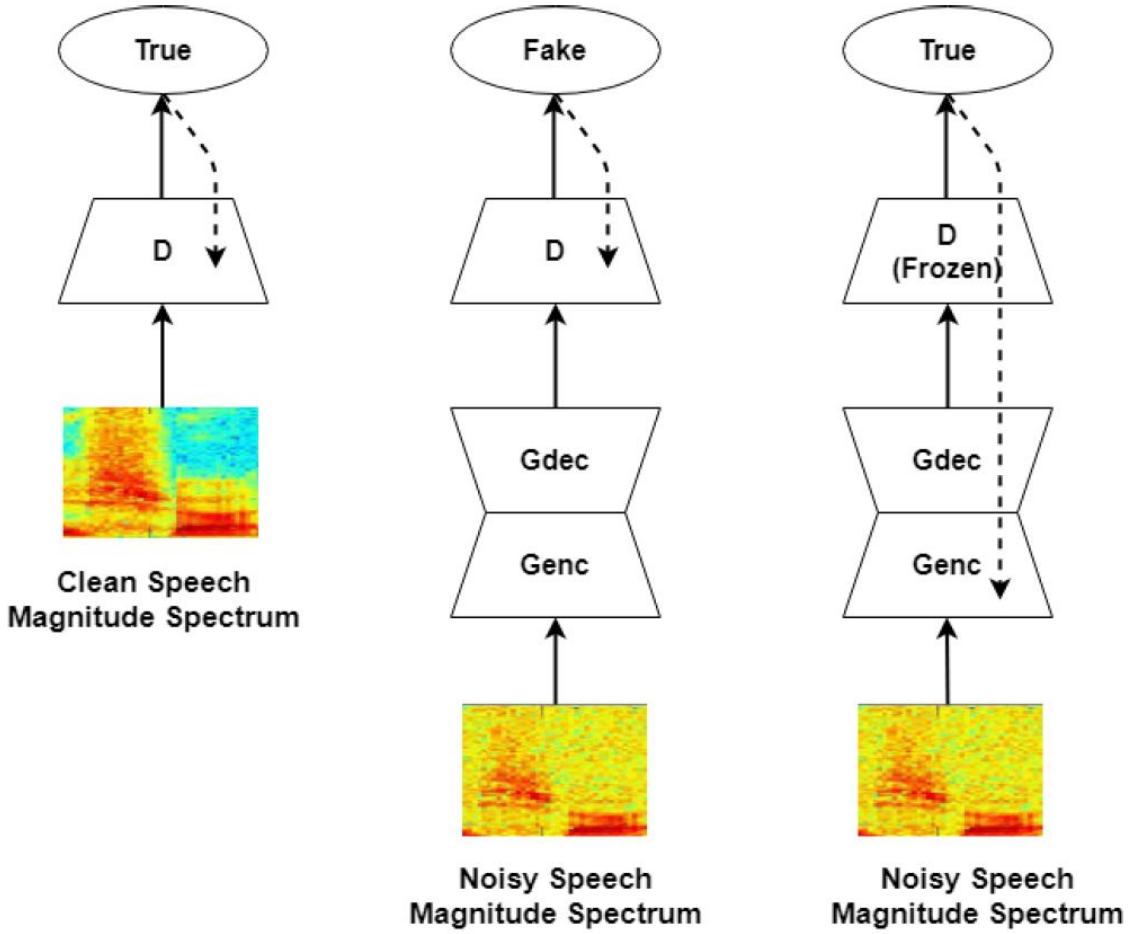


Figure 14: GAN architecture in the training stage. Genc and Gdec refer to generative encoder and decoder, respectively. Dashed lines represent gradient backpropagation. (a) *D* backpropgates a batch of real samples, (b) *D* backpropgates a batch of fake samples generated by *G* and classify them as fake, (c) *D* is frozen and *G* backpropgates to make *D* misclassify [8, 9].

the fake ones. The training will continue until both *G* and *D* increase their accuracy as much as possible and the generated samples by *G* and the real samples in the dataset are indistinguishable. The final goal is to shape the generator's loss function under the discriminator's guidance. An example of enhancing the noisy speech magnitude is shown in Fig. 14 where the structure of *G* is similar to an auto-encoder that aims to generate the clean speech magnitude given the noisy one [79]. Although both *G* and *D* are used in the training stage to improve their accuracy so that the output of *G* is corrected towards the realistic distribution, only the trained *G* will be used in the testing stage to generate the desired samples. In the speech enhancement field, GAN is mostly

adopted as a supervised learning problem in a deterministic manner, which is employed to map the noisy speech spectrum to the clean one [7, 8].

2.3 Hyper Parameters of DNN

While designing a DNN-based method, some hyperparameters have to be carefully chosen to accomplish the training correctly. Choosing correct values for these parameters guarantees better results, while inappropriate values would make the DNN not to be trained. In the following, these parameters are shortly expressed.

2.3.1 Neurons and Layers

In a DNN architecture, the number of layers and neurons per layer are the primary parameters that have to be carefully chosen. Typically, the first layer is crucial as it links the input data with the rest of the network, which has to be carefully designed to capture enough and appropriate information from the input and transfer them to the next layer. In case that some layer does not comprise enough neurons, it will be a bottleneck for the rest of the network and might cause performance degradation. In effect, DNN tries to transform input information from a low-dimensional into a high-dimensional space where more discriminative information is exposed to the network to form an accurate mapping between input and output. It is worth mentioning that increasing the number of layers and neurons would result in better results in theory; however, such a deep and wide network encounters vanishing and over-fitting problems in practice. According to [73], a network that is wide, a large number of neurons per layer, and shallow, a small number of layers, are easier to overfit, while a deep network, a large number of layers, and narrow, a small number of neurons per layer, are easier to underfit.

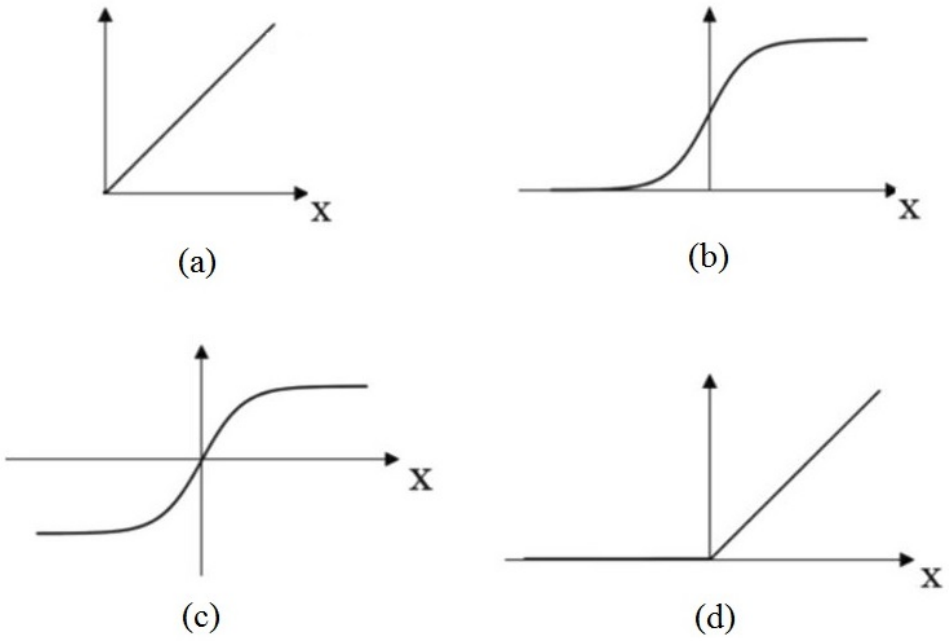


Figure 15: Activation functions, (a) linear, (b) sigmoid, (c) hyperbolic tangent, (d) ReLU [10].

2.3.2 Activation Function

DNNs are considered universal function approximates, i.e., they can compute and learn any function at high levels of non-linearity. Almost any process can be represented as a functional computation in DNNs. Hence, there is a need for non-linear activation functions to add non-linearity to DNN to make it more powerful to learn complicated transformations between input and output. These non-linear functions have a degree more than one and a curvature form. In addition, another significant feature of an activation function is to be differentiable so as to perform a backpropagation optimization strategy that uses gradient descend or any other optimization technique. In the following, we explain some essential and commonly used activation functions.

Linear activation function: This activation function performs as a constant multiplication to its input, i.e., $f(x) = cx$, which means that no non-linearity will be added to the network with all linear activation functions. This function in shown in Fig. 15 (a). In other words, using linear activation function for the whole network leads to an output that is a multiplication of the input data. In a DNN framework, when the values of the training target have no limitation, linear

activation function is used to allow DNN output values have wide and arbitrary fluctuations.

Sigmoid activation function: One of the most common activation functions is sigmoid that performs like a constrain on its output, i.e., it limits its output to a value between zero and one, as shown in Fig. 15 (b). The equation for this activation function is given below.

$$sigmoid(x) = \frac{1}{1 + \exp^{-x}} \tag{9}$$

The gradient of this activation function is a smooth bell-shaped function that maximizes at zero. Although this activation function is widely used, especially for classification tasks, there are some disadvantages to it. Firstly, the vanishing gradient problem might happen since its gradient is almost zero for large and small inputs. Secondly, this activation function is computationally expensive due to its exponential form. Lastly, its output is not symmetric around zero that would be problematic in some cases.

Hyperbolic tangent activation function (tanh): This activation function that is a shifted version of sigmoid is defined as follows,

$$tanh(x) = \frac{2}{1 + \exp^{-2x}} = 2 * sigmoid(2x) - 1 \tag{10}$$

Unlike sigmoid, hyperbolic tangent activation function is symmetric around zero, making it easier to model data with values varying from a negative to the positive range, as shown in Fig. 15 (c). Although this activation function is suitable for many applications, it might cause a vanishing gradient problem if used for the hidden layers. Besides, this activation function suffers from high computations like sigmoid.

Rectified linear unit activation function (ReLU): This activation function is the most widely used one for different classes of DNN. Its equation is very simple as $ReLU(x) = max(0, x)$. ReLU, shown in Fig. 15 (d), is very simple and efficient since its gradient is either zero or one that makes it a very low computation activation function and resistant to the vanishing gradient problem. It was recently proved that ReLU has six times improvement in convergence than *tanh*.

It is worth pointing out that a linear activation function is an ideal option for a DNN to be optimized well with common gradient-based methods, while DNN requires non-linearity to be able to model complex relationships in the training data. ReLU activation function is optimal in the view of both criteria mentioned above since it is nearly linear that makes the model easy to optimize while providing enough non-linearities to the model. ReLU is often used as the activation function of the hidden layers, while for the output layer of a DNN, other activation functions are commonly used [80].

2.3.3 Batch Size

Backpropagation, as the most common optimization method in the deep learning field, uses the gradient of prediction error to update weights and biases of a DNN. The empirical gradient is estimated from a subset of training samples that is called a batch. To shed light on the importance of the batch size, it is to be mentioned that it directly affects the resulting model and convergence speed.

The gradient descent method can be categorized into three forms given the batch size: (a) *batch gradient descent* where the whole dataset is considered as a big batch, (b) *stochastic gradient descent* where each sample is considered as batch size, and (c) mini-batch gradient descent where the whole dataset is divided into several batches. Each of these methods brings pros and cons as studied in [73]. In short, the case *a* leads to the best convergence; but, requires passing the whole dataset at once through the network that is not efficient in many large-scale problems. Case *b* can easily jump out of poor local optima, and it is very fast; but, it causes a very noisy gradient because the parameters are updated based on every single sample and may fluctuate around the minimum. Case *c* offers the best trade-off with an unbiased gradient and small variance, which can converge fast enough. As such, the entire dataset is divided into several batches, and the network is then updated once per batch. For example, the network parameters are being updated twice if there are two batches.

2.3.4 Epoch

The training algorithm has to work through the entire dataset many times to minimize the loss function. The number of times that this process repeats is called epoch that is traditionally a large number so that the loss function is minimized. Each epoch contains one or several batches. Monitoring the curve of the loss function against the epoch gives an idea about under-fitting, over-fitting, or suitably fitting to the given dataset [8].

2.3.5 Learning Rate

Learning rate is one of the most critical configurable hyper parameters that highly impact the training process. The step size with which the trainable parameters of a model are updated per batch is called learning rate, i.e., it controls how quickly the model is adapted to the problem. It accepts values between 0 and 1, where a very small learning rate causes a prolonged training process or might have stuck training to a local minimum, while its large value might make the training process unstable or causes quickly learning a sub-optimal set of trainable parameters. The choice of the best learning rate is still based on trial and error, and there is no theoretical way to obtain an optimal learning rate. It is worth pointing out that a combination of the learning rate and batch size directly impacts the learning behavior [73]. It is common to adopt a dynamic learning rate during the training process. Hence, training usually starts with a large learning rate and small batch size. After several epochs, the learning rate gets smaller while the batch size becomes bigger.

2.4 Evaluation Tools

To model speech enhancement as a supervised learning problem, large datasets are required for training and testing. Besides, evaluation of these methods requires speech quality measurement metrics representing the end-user opinion of perceived speech quality. This section will introduce several common datasets and metrics for speech enhancement.

2.4.1 Training Datasets

DNN must be trained with large datasets to obtain a good speech enhancement performance. Two datasets are required in both training and testing stages: clean speech and noise. Hence, the noisy speech can be simulated by mixing clean utterances and noises at the desired SNR level. Several clean and noise datasets are introduced below.

Clean Speech

An appropriate dataset for supervised speech enhancement should include utterances with good audio and text characteristics. The former refers to the recording quality of utterances and their total duration, as well as the number of speakers and their attributes. We mean the accent, dialect, tempo, and distribution of age group and gender by attribute. The latter includes language and richness of contexts in terms of inclusiveness of many words and their repetition [8, 39]. Some common clean speech datasets are given below.

TIMIT [81]: This dataset provides recordings for acoustic-phonetic studies that are used to develop and evaluate speech recognition algorithms. TIMIT dataset consists of utterances from 438 male and 192 female speakers featuring eight major dialect divisions of American English. Ten short phonetically rich sentences are recorded from each speaker, leading to 6300 utterances. The dataset is recorded at a 16 kHz sampling rate.

IEEE Corpus [82]: This corpus comprises a collection of utterances originally utilized for standardized communication system testing, like Voice over IP, cellular phones, or other telephones. Since speech enhancement requires repeatable and standardized sequences, the IEEE corpus has been commonly utilized in this field. This corpus consists of 720 utterances spoken by a single male, recorded at a sampling frequency of 25 kHz. These utterances are phonetically balanced in a way that the appearance frequency of a specific phoneme is matched with that in English.

MUSAN [83]: MUSAN is a complementary dataset that includes speech utterances in 12 languages, music in several genres, and many noises. This dataset is mainly designed for evaluating

voice activity detection systems and discrimination of speech and noise. The duration of the speech portion of MUSAN is about 60 hours, including 40 hour recordings of US government hearings, committees, and debates, and the rest 20 hours is speech in 11 other languages. It is worth pointing out that the 40-hour portion is totally in English.

TED-LIUM [84]: Laboratoire Informatique de l'Universite du Maine (LIUM) has developed this dataset for automatic speech recognition systems using the English content of Technology, Entertainment, and Design (TED) talks. The first version of this dataset released in [84] comprised 118 hours of speech, 82 and 36 hours by males and females, respectively. This dataset is made up of 774 talks; each is around 9 minutes on average. Then, the second and third versions were released in [85] and [86], respectively, to enrich the dataset in terms of language. The new dataset includes 2.56 million words. Since the talks are recorded in large arenas and conference halls packed with the audience, there is noticeable reverberation and noise in the recordings.

LibriSpeech [87]: This dataset is originally designed for automatic speech recognition systems. LibriSpeech includes recordings of free public domain audiobooks called LibriVox [88] readings, all completely in English. The sampling rate of recordings of this dataset is 16 kHz, and its entire duration is about 1000 hours. The readings are spoken by 5500 speakers, almost half males and females. Each recording is limited to 25 minutes in this dataset to avoid imbalances in per-speaker audio duration. This dataset is subsequently extended to multi-language in [89].

Noise

Noise audio files are often employed to generate noisy speech for training and testing a supervised speech enhancement algorithm. Noise can be recorded in an actual noisy or simulated environment. The more noises are included in the training dataset, the better the model will be generalized to the unseen acoustic environments, as mentioned in Section 1.3.3. Several common noise datasets in the supervised speech enhancement field are explained below.

Aurora-2 [90]: This dataset is mainly designed to feature the most plausible noisy environments for telecommunication terminals. The duration of each noise file is 10 seconds, recorded

at a sampling rate of 8 kHz. The noises in this dataset are recorded in an exhibition hall, subway, restaurant, babble, car, street, train, and airport.

NOISEX-92 [91]: One of the most popular and commonly used noise datasets in DNN-based speech enhancement is NOISEX-92. This dataset contains white noise and various highly non-stationary noises. The duration of each of these audio files in this dataset is around four minutes. The sampling rate of the recordings is 16 kHz. One of the reasons for the popularity of this dataset is its inclusiveness, i.e., including many noise types such as factory noise, pink noise, various military noises like fighter jets (Buccaneer and F16), HF radio channel noise, babble noise, destroyer noises like engine room and operations room, tank noises like Leopard and M109, car noises like Volvo, and machine gun.

CHiME-5 [92]: There is a biannual competition for speech recognition and separation (separating speech from background noise) where the organizer releases a corresponding dataset each time. There is a common practice that researchers utilize this dataset in academic research as well. The background noise in this dataset is recorded in 20 different real dinner parties in the home. These parties occur in different locations, i.e., kitchen, living room, and dining room. There are four people at these parties, two guests and two hosts. All the participants know each other and act naturally. The duration of these noises is half an hour.

2.4.2 Objective Evaluation Metrics

There are two approaches to evaluate the processed speech quality: subjective listening and objective metrics. The former refers to evaluating the processed speech by real listeners. This method provides more trustful evaluation results; however, it requires experienced or trained listeners and extensive resources. This method is thus expensive and time-consuming. The latter refers to the proposed engines that evaluate the processed speech considering linguistics, psychoacoustics, and semantics knowledge. These metrics are usually highly correlated with subjective results. Some common objective metrics are explained below.

PESQ

This metric, proposed in [93], aims to characterize the quality of speech perceived by users. The process to calculate the PESQ score is as follows. The processed and original speech passes through a normalizer to have an equal listening level. Afterward, they are aligned in time to avoid time delays. Finally, an auditory transform processes them to calculate the loudness spectra of both signals. The difference between loudness spectra is averaged over time and frequency to produce the prediction of subjective perceptual mean opinion score. PESQ faithfully reflects the subjective score; thus, it is a reliable metric to measure speech quality.

STOI

The second important and common measure to evaluate the objective assessment of speech intelligibility is short-time objective intelligibility (STOI) [94]. This metric measures speech intelligibility by averaging the correlation of short-time temporal envelopes between the original and processed speech. The range is between 0 and 1, the higher, the better intelligibility. STOI is shown to be highly correlated with subjective scores.

SSNR

The segmental SNR (or SSNR) in dB is one of the most widely used scores for speech enhancement evaluation, which measures speech distortion physically [95]. SSNR can be evaluated in both the time and frequency domain, while the time domain one is usually adopted for speech enhancement. After time alignment of the original and processed signal, SSNR is calculated using the following equation.

$$\text{SNR}_{seg} = \frac{10}{M} \sum_{m=0}^{M-1} \log_{10} \frac{\sum_{n=Nm}^{Nm+N-1} \|x(n)\|^2}{\sum_{n=Nm}^{Nm+N-1} \|\hat{x}(n) - x(n)\|^2} \tag{11}$$

where *M* and *N* denote the number of frames and frame size, respectively.

Chapter 3

A Serial Hybrid Neural Network for Speech Enhancement

3.1 Introduction

Nowadays, deep learning as a primary tool to develop data-driven information systems has led to revolutionary advances in speech enhancement. In this context, speech enhancement is treated as a supervised learning problem where DNN learns the highly complex relationship between a set of input signal features and a desired training target. In this chapter, we develop a DNN-based speech enhancement framework to remove the background noise in STFT domain.

As mentioned in Section 2.2.1, there are two main types of training targets in the literature, namely mapping-based and masking-based. Well-known mapping-based methods using FC and CNN networks for speech enhancement were presented in [96] and [5], respectively. Other examples of mapping-based methods for supervised speech enhancement can be found in [44, 97]. However, these approaches entail the use of a large training dataset to achieve an accurate mapping [12].

Instead of mapping the input noisy speech to the enhanced version, DNN can be exploited to predict a spectral mask to be applied to the noisy speech spectrogram. The masking-based methods

are inspired by the masking effect of the human auditory mechanism. The goal is to estimate a spectral mask using DNN and then apply it to the noisy speech; thus, the speech-dominant regions are retained and the noise-dominant ones are suppressed. Wang *et al.* [66] carried out a study to evaluate the enhancement performance of DNN models using different training targets. They employed an FC network to either directly estimate the STFT of the spectral magnitude of the clean speech or a spectral mask, including IRM and SMM. They concluded that estimating a spectral mask is more efficient than directly estimating the clean speech magnitude spectrum as far as quality and intelligibility scores are concerned.

Most of the above-mentioned training targets focused on enhancing the speech magnitude solely and used the noisy phase to restore the estimated speech, thereby underestimating the impact of phase enhancement on the overall performance. As discussed in Section 1.3.2, some researchers showed the advantages of exploiting phase information in speech enhancement. Erdoghan *et al.* [68] introduced a phase-sensitive spectrum approximation (PSSA) as an extension of SMM and showed that the incorporation of phase information leads to a better signal-to-distortion ratio of the estimated clean speech in comparison to SMM. Williamson *et al.* [6] introduced a DNN-based technique to predict a complex IRM (cIRM) to enhance the noisy speech phase and magnitude simultaneously. In [6] and [98], respectively, an FC network and a composite model were employed to estimate cIRM, leading to improved quality and intelligibility of the enhanced speech. Accurate estimation of the mask is very important to the enhancing performance. As such, a delicately designed DNN is required in addition to designing an efficient mask.

As mentioned in Section 2.2.2, DNN-based methods for speech enhancement often employ an FC network that comprises a large number of parameters. More importantly, these methods neglect temporal information even though speech exhibits strong temporal dependencies. Unlike an FC network that processes input samples independently, RNN with self-connections treats input samples as a sequence and models the information flow over time. Making RNN a natural choice to model the temporal dynamics of speech using information extracted from previous frames; thus, RNN can be employed as a learning machine for speech enhancement [7, 55]. Besides, LSTM

networks were introduced for sequence learning because RNN suffers from the vanishing and exploding gradient problem. Jitong *et al.* [55] employed an LSTM network to estimate IRM and showed that compared to FC networks, LSTM significantly improves speech enhancement performance while boosting speaker generalization capability. In [99], a modified version of LSTM called bidirectional LSTM (BLSTM) was proposed, which combines information from past and future states to calculate the output sequence, thus taking full advantage of the input signal's contextual information.

The choice of features for the network input is very important in the learning process since inappropriate inputs may result in deviation of the output from its reference value. Instead of using conventional features, a CNN can be employed to extract the most appropriate input speech features for the enhancement task. As an efficient method of feature extraction, a traditional CNN made up of cascade connections of convolutional and pooling layers was employed for speech recognition in [100] and for acoustic scene classification in [101]. However, due to the small receptive field of CNN filters, the general contextual information of speech is suppressed. To address this problem, a new CNN structure with 1D convolution in the frequency domain and 2D dilated convolution in the STFT domain was proposed in [12] to enlarge the receptive field while keeping the number of parameters and memory footprint small.

In this chapter, we propose a novel serial hybrid neural network that integrates a CNN and LSTM for speech enhancement based on phase sensitive mask (PSM) estimation. The novel contributions of this framework are summarized as follows.

1. A new low-complexity fully-convolutional CNN that facilitates learning, accelerates convergence, and reduces the number of model parameters is proposed to extract the most appropriate features of the input speech.

2. An attention technique is adopted to adaptively emphasize the valuable features extracted by CNN and suppress less important ones.

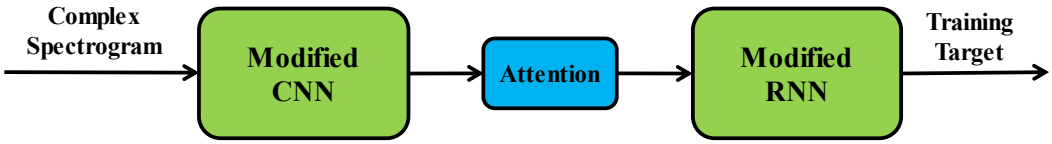


Figure 16: A high-level block diagram of the serial hybrid model.

3. An LSTM is employed to take advantage of temporal dependencies of speech and accomplish the regression between the CNN-extracted features and the mask values.

4. Different RNN variations are evaluated and analyzed to optimize the network structure in terms of performance, computation time, memory footprint, and number of model parameters.

5. A grouping strategy is adopted to reduce the number of RNN parameters. Moreover, different forms of grouping strategy are compared in terms of the objective quality of the enhanced speech and the number of trainable parameters.

6. The proposed model is evaluated using different datasets and compared to some related DNN-based methods. Different training targets are also investigated to exploit the phase information alongside magnitude enhancement so as to achieve the best performance.

The rest of this chapter is organized as follows: the high-level block diagram of this model is presented in Section 3.2. The details about different training targets for this model are explained in Section 3.2.3. The modified CNN and RNN are presented in Sections 3.2.1 and 3.2.2, respectively. Finally, the experimental results including a wide range of comparisons and relevant discussions are provided in Section 3.3.

3.2 Proposed Serial Hybrid Model

Fig. 16 shows the high-level block diagram of the proposed serial hybrid model. In this model, feature extraction is executed by a modified CNN structure that is fully-convolutional with frequency dilated convolutions. An attention block emphasizes the valuable features, and a modified RNN variation then maps these features to the training target by exploiting the speech's temporal information. The key components of the model are discussed in the following.

3.2.1 Modified CNN Structure

Dilated Convolution

CNN was initially designed for image classification. A conventional CNN is made up of pairs of convolutional and pooling layers followed by an FC network. The former aims to extract features, while the latter accomplishes classification. As mentioned in Section 2.2.2, a pair of convolutional and pooling layers is usually called a convolutional layer. This kind of CNN structure can be easily employed for speech enhancement in the STFT domain since the speech spectrogram looks like an image.

The conventional CNN kernels were initially designed to capture local correlations of an image for the image processing purpose because the image usually exhibits local correlations while speech spectrogram mainly possesses non-local correlations along the frequency axis [102]. On the one hand, non-local correlations in speech spectrogram, like the correlation of harmonics, can be exploited to improve the clean speech spectrogram's prediction. However, since the frequency dimension of speech spectrogram as the input of CNN is at the rate of a few hundred, limitation of the receptive field (the local area from the previous layer) of convolution layers results in destroying global correlations of speech spectrogram [12]. On the other hand, the pooling layer reduces resolution and sensitivity to local variations; this is so obstructive for the CNN architecture if used for spacial dimension reduction [7]. As pointed out in [103], max-pooling keeps merely very rough information and discards the rest. Besides, average pooling neglects the importance of

local structure by attenuating individual grid contribution in a local region.

A common practice is to enlarge the CNN kernel size along the frequency axis to meet large receptive field requirements, like in [104] where filters with the size of 25 are utilized, which, however, leads to high complexity and consequently low speed. Another approach to enlarge the receptive field of the convolution layer is to use stride convolution, which can reduce run time by reducing the size of intermediate representations and introduce some translation invariance [105]. However, striding makes the TF cell prediction overly smooth and less accurate which means reducing the spatial resolution.

To overcome the limitations mentioned above, dilated convolution was introduced and already successfully applied for imaging segmentation [106], and speech synthesis [11]. Consider a regular causal convolution for 1D data, as shown in Fig. 17. In this figure, four causal convolutional layers are stacked, and the kernel size is two. This leads to the receptive field size of only five, which is very small and does not benefit from contextual information. It is worth pointing out that this convolution guarantees no use of future time steps while calculating the current time step. Furthermore, the run time of CNNs is faster than RNNs since there is no feedback connection in the network. With the stacked causal convolutional structure, there is a need for many layers to enlarge the receptive field, or a large CNN kernel size, as mentioned above.

However, dilated convolutional layers address this issue without increasing the model size or computational cost. Such layers expand the area that the kernel covers by skipping a certain number of values. The way it works is similar to using a large kernel by dilating it with zeros while it is pretty more efficient. This is like a pooling layer; but, the input and output sizes are similar. A CNN with stacked dilated convolutional layers is shown in Fig. 18. In this figure, the kernel size is still two, but dilations of 1, 2, 4, and 8 for each layer lead to a receptive field of 16. Hence, the receptive field can be exponentially expanded by stacking dilation convolution layers while the input resolution and coverage are kept intact.

Based on the conventional convolution of a 1D signal F and a kernel k, we define the dilated

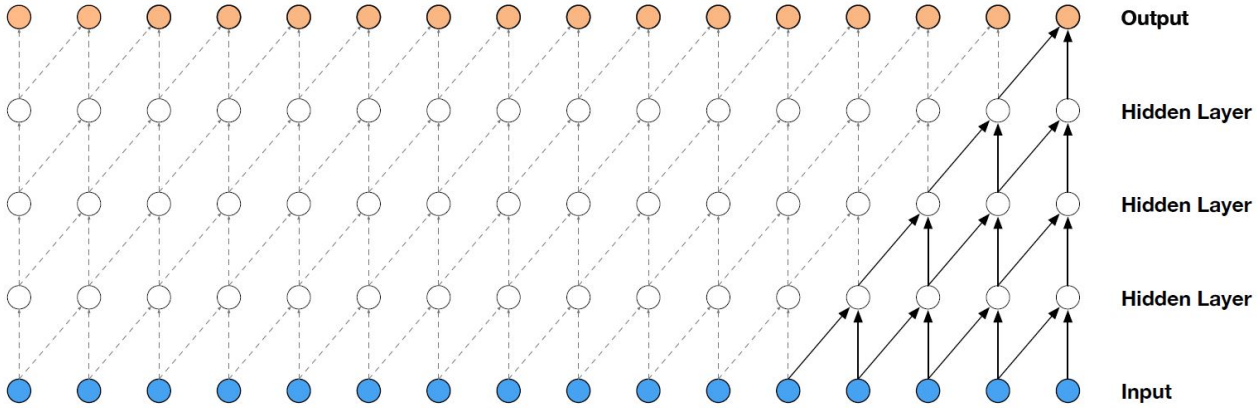


Figure 17: Visualization of stacking of causal convolutional layers [11].

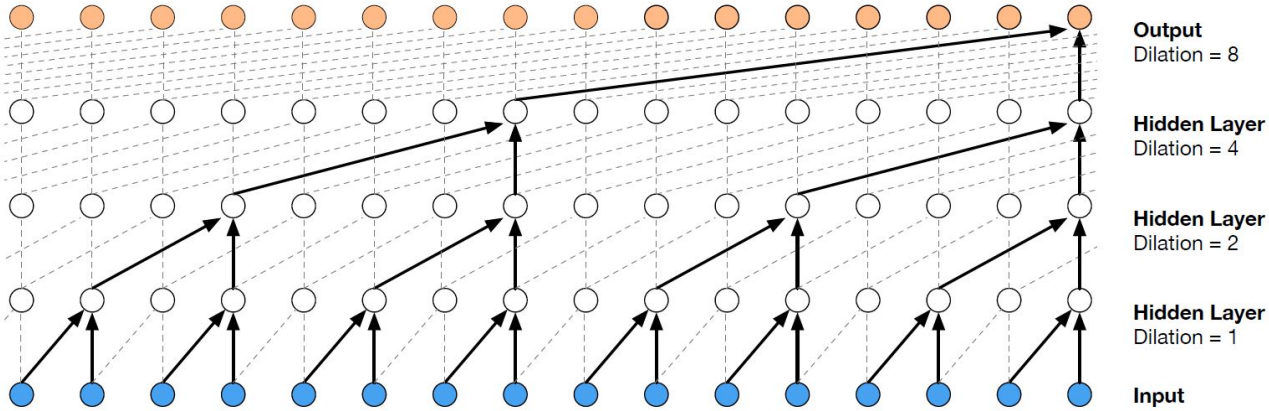


Figure 18: Visualization of stacking of dilated causal convolutional layers [11].

convolution with dilation factor l as,

$$(k *_l F)(t) = \sum_{\tau=-\infty}^{\infty} k(\tau)F(t - l\tau) \tag{12}$$

where t and $*_l$ denote the discrete time and dilated convolution, respectively. Obviously, this definition reduces to a regular convolution when l=1. Also, it can be easily extended to 2D convolution. Figure 19 shows a dilated 2D frequency convolution with an increasing dilation factor along the frequency axis. It is worth pointing out that the CNN in the hybrid model is designed to capture the spectral information; thus, there is no dilation alongside the time axis.

Inspired by [11], we employ a fully-convolutional CNN with frequency dilated convolution to

exploit the most appropriate speech features. This CNN obtains a large receptive field along the frequency axis while keeping CNN filter size relatively small. Furthermore, to facilitate the model training, residual learning and skip connection techniques [72] are adopted. It is worth mentioning that there is no pooling layer in this CNN structure.

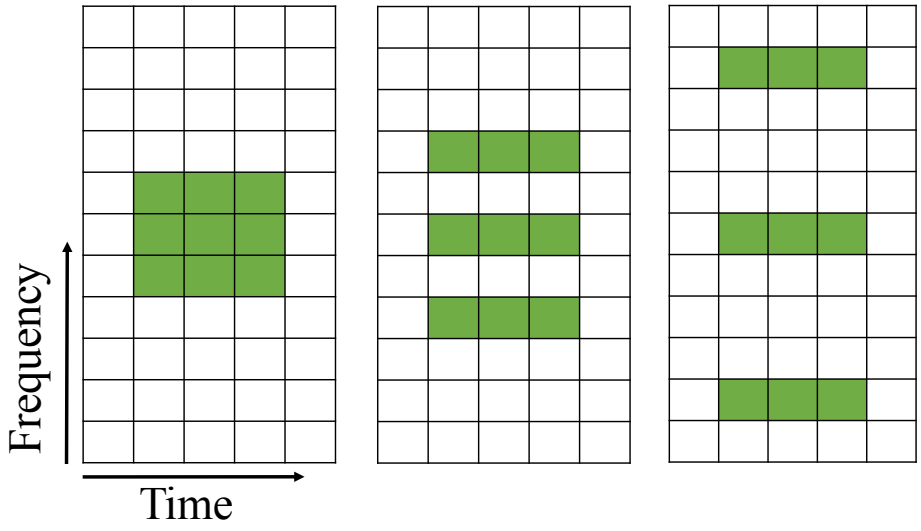
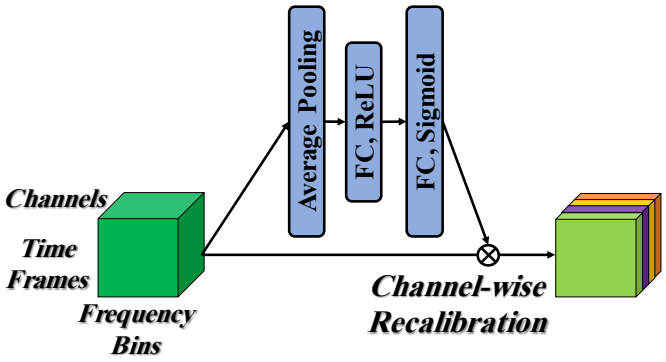


Figure 19: Frequency-dilated convolution. With filter size 3×3, the dilation rate from left to right is 1, 2, and 4, respectively. No dilation along the time axis [12].

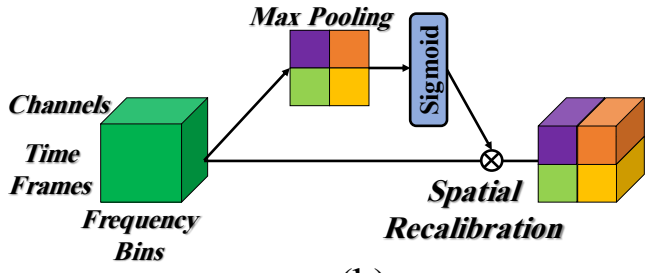
Attention Techniques

CNN contains many feature maps that may have different levels of significance. Accordingly, emphasizing informative feature maps improves the model performance. By recalibrating feature maps, an attention mechanism adaptively emphasizes the informative ones while suppressing others.

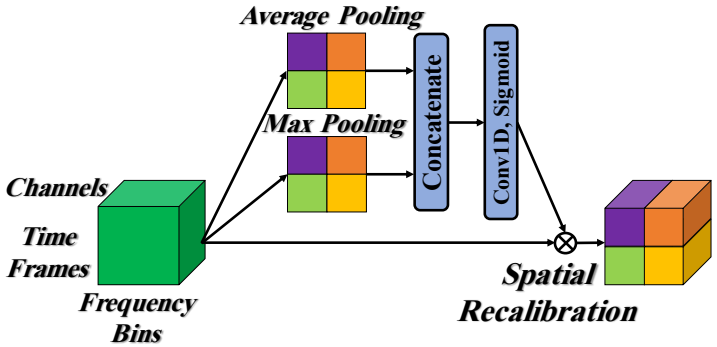
A successful attention mechanism termed squeeze-and-excitation (SAE) was introduced in [107] focusing on channel relationships. In this approach, illustrated in Fig. 20 (a), an average-pooling operation spatially aggregates the global information of each feature map to a channel descriptor in the squeeze stage. Then, an FC network captures channel-wise dependencies by adjusting the descriptor in the excitation stage. Finally, the original feature maps are recalibrated by the excitation values, and the results are delivered to the subsequent layer.



(a)



(b)



(c)

Figure 20: SAE attention techniques, (a) channel-wise, (b) spatial, (c) spatial using both max and average pooling.

Inspired by SAE but aiming to take advantage of pixel-wise spatial information, Roy *et al.* [108] introduced spatial SAE, illustrated in Fig. 20 (b), wherein the squeeze operation is performed along channels while the excitation is spatial.

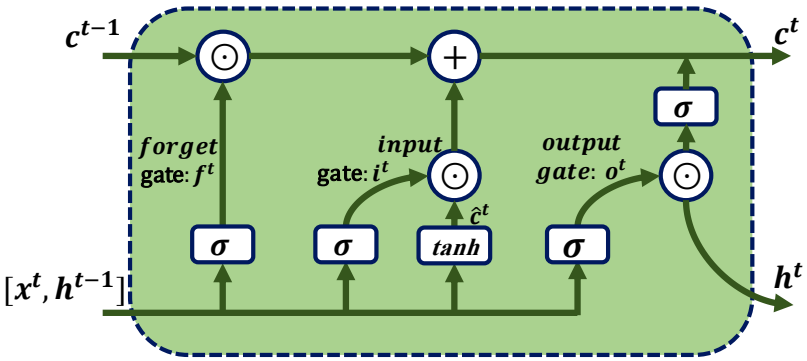
Inspired by [108], we employ a spatial SAE between CNN and LSTM to emphasize significant features generated by CNN. In particular, we use both average and max pooling across different channels to squeeze the input information, as shown in Fig. 20 (c). A convolutional layer then combines the results, and finally, the output is element-wise multiplied with the input feature maps. In this work, we investigate the use of these attention techniques in the proposed model.

3.2.2 Modified RNN Variations

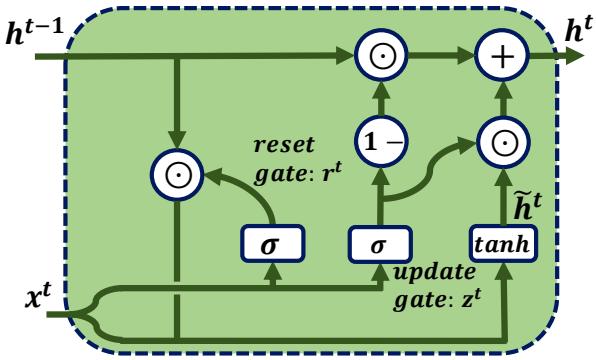
RNN Variations

Most studies on speech enhancement try to take advantage of strong temporal dependencies of speech and provide useful temporal contextual information to a neural network by utilizing a window of frames as input due to the impact of contiguous frames on the current frame [19, 55]. However, not all contiguous frames have the same impact on the current frame. Also, the information beyond this window is not exploited. In this context, an RNN treats input samples as a sequence and can model the changes over time, making it the best choice to model the temporal dynamics of speech. Furthermore, it is demonstrated in [109] that LSTM, as the most widely used type of RNN, is beneficial for low-latency enhancement and it, even without using future frames, outperforms a fully connected model with future frames. More importantly, LSTM is an effective approach for speaker-independent speech enhancement compared to an FC network that fails to model various speakers [55].

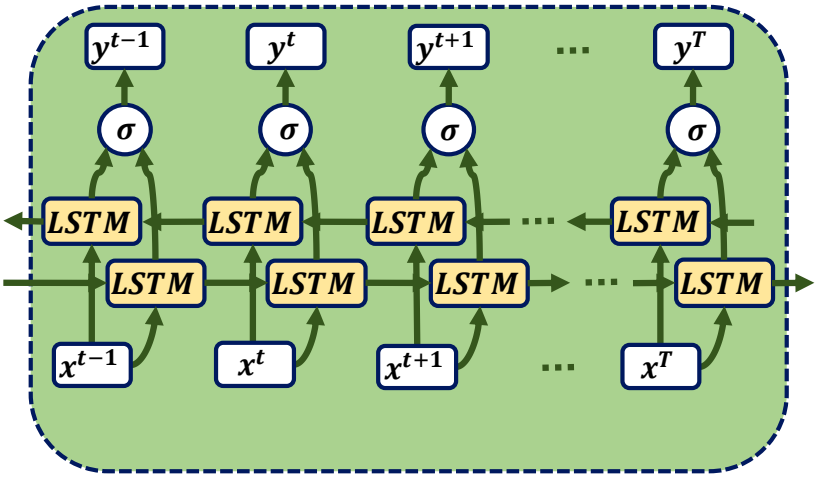
LSTM prevents a general RNN from vanishing and exploding the gradient, a problem caused by very long-term dependencies. It contains a memory cell with three gates, i.e., input gate, forget gate, and output gate, to facilitate information flow over time. The input gate controls how much information should be added to the cell; the forget gate decides how much previous information should be erased from the cell; and the output gate computes the next hidden state. Figure 21 (a)



(a)



(b)



(c)

Figure 21: RNN variations. Block diagrams of (a) LSTM, (b) GRU, (c) BLSTM.

illustrates an LSTM unit. Assume $x^t \in \mathbb{R}^{M\times 1}$ is an external input at time t, and $h^{t-1} \in \mathbb{R}^{N\times 1}$ is a recurrent hidden state at time $t-1$. Then, the three gates can be defined as i^t, f^t, and $o^t \in \mathbb{R}^{N\times 1}$, respectively, which are expressed as:

$$i^t = \sigma(W_i x^t + U_i h^{t-1} + b_i) \tag{13}$$

$$f^t = \sigma(W_f x^t + U_f h^{t-1} + b_f) \tag{14}$$

$$o^t = \sigma(W_o x^t + U_o h^{t-1} + b_o) \tag{15}$$

where σ is a sigmoid activation function; $W \in \mathbb{R}^{N\times M}$ and $U \in \mathbb{R}^{N\times N}$ are are the weight matrices and $b \in \mathbb{R}^{N\times 1}$ represents the bias vector. The current value of memory cell state, c^t, is calculated based on an intermediate candidate and the previous value of the internal memory cell state, represented by $\hat{c}^t$ and c^{t-1}, respectively, as expressed below.

$$\hat{c}^t = tanh(W_c x^t + U_c h^{t-1} + b_c) \tag{16}$$

$$c^t = f^t \odot c^{t-1} + i^t \odot \hat{c}^t \tag{17}$$

$$h^t = o^t \odot tanh(c^t) \tag{18}$$

where $\odot$ denotes element-wise multiplication, and h^t is the current hidden state. Considering the dimension of cell state and input vector as N and M, the total number of parameters for an LSTM network is $4 \times (N^2 + NM + N)$.

Combining the forget and input gates in LSTM into a single one, GRU is introduced with two gates r^t and z^t, named reset and update gates, respectively. GRU as a variation of LSTM is faster and computationally more efficient than LSTM, while in some cases, it yields even better performance on less training data [110]. GRU structure is depicted in Fig. 21 (b). At each step,

GRU is implemented by the following set of equations,

$$z^t = \sigma(W_z x^t + U_z h^{t-1} + b_z) \tag{19}$$

$$r^t = \sigma(W_r x^t + U_r h^{t-1} + b_r) \tag{20}$$

$$\hat{h^t} = \sigma(W_h x^t + U_h(r^t \odot h^{t-1}) + b_h) \tag{21}$$

$$h^t = (1 - z^t) \odot h^{t-1} + z^t \odot \hat{h}^t \tag{22}$$

Equations (21) and (22) are similar to (16) and (17) where one gate and its associated parameters are omitted. The total number of parameters for a GRU-based network is then $3 \times (N^2 + NM + N)$ [111]. GRU has no memory unit and exposes full hidden content without any control. Thus, it is computationally more efficient, while its performance is sometimes on par with LSTM [110].

Other variations of RNNs are bidirectional networks, such as BLSTM and BGRU, introduced to take full advantage of input information. A bidirectional cell is made up of two LSTM layers connected to the same output where the output sequence is calculated using both forward and backward hidden sequences, i.e., past and future states, simultaneously. Figure 21 (c) illustrates a BLSTM structure.

As will be seen from our experimental results and comparison in Section 3.3.4, LSTM is the best trade-off among the RNN variations for the proposed model for speech enhancement in terms of quality of the results, computational time, memory footprint, and the number of model parameters. As such, an LSTM network is employed to perform the final regression in our enhancement system.

Complexity Reduction using Grouped Recurrent Networks

RNNs have been widely used for sequence learning and achieved state-of-art results in many applications. However, RNNs suffer from high complexity caused by parameter redundancies in weight matrices that transfer hidden states between different steps and those transforming feature

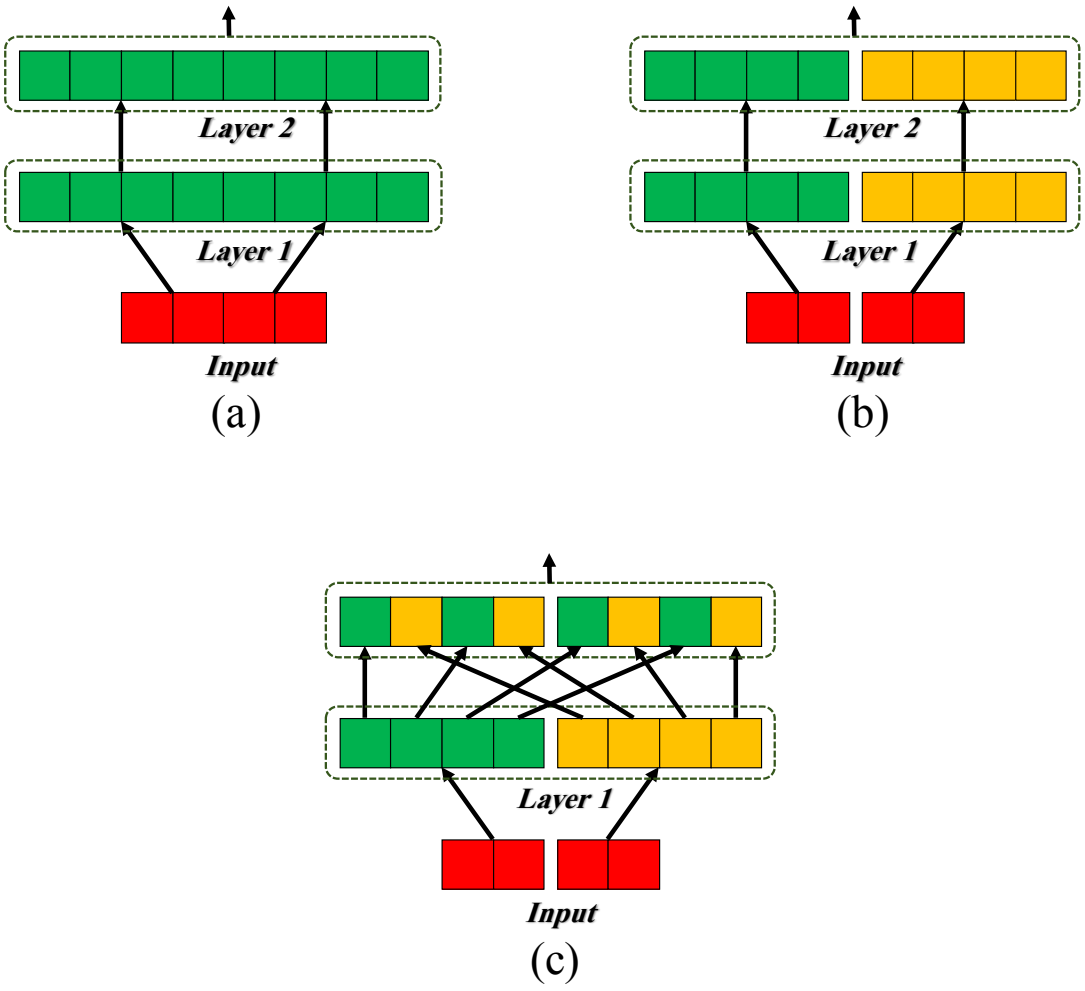


Figure 22: A two-layer RNN network with (a) no group strategy, (b) group strategy, (c) group strategy and representation rearrangement.

representations from a low to a high level. To alleviate this issue, group recurrent layers were introduced in [112] that reduce the complexity of RNN while maintaining the same level of performance. This technique is successfully employed in the RNN part of a high complexity gated convolutional recurrent networks [113].

Ignoring the bias vector b_i, the number of required parameters to implement equation (13) is $N^2 + N \times M$. If the input x and recurrent layer h are split into K disjoint groups performing independently, the number of parameters becomes,

$$K \times \left((\frac{N}{K})^2 + \frac{N}{K} \times \frac{M}{K} \right) = \frac{N^2 + N \times M}{K} \tag{23}$$

As such, the number of RNN parameters drops by K. Fig. 22 (a) and (b) depicted a standard and grouped RNN, respectively. In a grouped RNN, intra-group dependencies are efficiently learned. However, inter-group dependencies, i.e., the dependencies across different groups, are lost since individual groups perform independently. Since inter-group correlations are cut off in this architecture, the representation power drops. To tackle this issue, a parameter-free representation rearrangement technique between consecutive group layers was introduced [112], as illustrated in Fig. 22 (c). It is to grant the subsequent layers access to all groups' outputs to capture the inter-group dependencies. This regrouped RNN reduces our model's complexity while keeping the performance nearly intact.

3.2.3 Training Targets

The noisy speech model in both time and STFT domain was explained in Section 2.1. As mentioned, $Y(k,l)$, $X(k,l)$, $N(k,l)$ represent the STFTs of noisy speech signal, $y(t)$, clean speech signal, $x(t)$, and noise signal, $n(t)$, over consecutive frames, respectively, and k and l denote the time frame and frequency discrete indices, respectively.

In the sequel, we shall often represent complex STFT coefficients in terms of their magnitude and phase, as $X(k,l) = |X(k,l)|\angle X(k,l)$, or in terms of the real and imaginary parts, as $X(k,l) =$

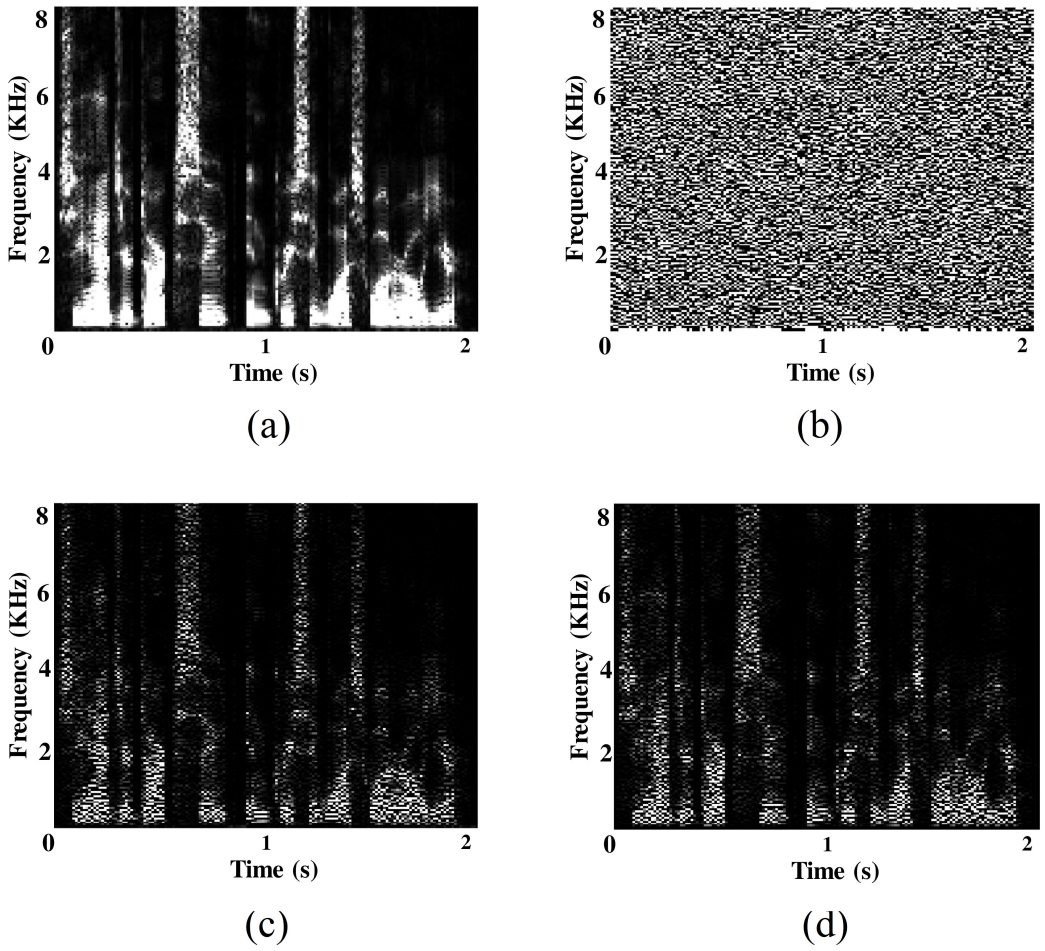


Figure 23: Spectrogram plot of clean speech (a) magnitude, (b) phase, (c) real component, and (d) imaginary component.

$\Re(X(k,l)) + i\Im(X(k,l))$. Fig. 23 (a) and (b) show the spectrogram plots of the magnitude and phase of a representative clean speech utterance, while Fig. 23 (c) and (d) depict the real and imaginary spectrogram components of the same utterance, respectively. As shown, the magnitude spectrum of the clean speech exhibits a clear structure that is amenable to supervised learning and thus has been considered as the training target in many studies, such as [44, 97], where the DNN-estimated magnitude spectrogram is combined with the noisy phase to resynthesize the clean speech. Besides, some studies such as [65, 66], consider IRM as the training target, as below,

$$IRM(k,l) = \sqrt{\frac{X^2(k,l)}{X^2(k,l) + N^2(k,l)}} \tag{24}$$

The estimated IRM will be multiplied by the noisy speech spectrogram, and then the estimated clean speech will be resynthesized using the noisy phase. Meanwhile, phase processing is prominent for speech enhancement, particularly at low SNR levels, as the phase of background noise is dominant at these SNR levels. Hence, the PSM as an extension to IRM was introduced in [68] to exploit the phase information in the enhancement procedure, which is defined as,

$$PSM(k,l) \;=\; \Re\left(\frac{X(k,l)}{Y(k,l)}\right) \;=\; \Re\left(\frac{|X(k,l)|}{|Y(k,l)|}e^{i(\angle X(k,l)-\angle Y(k,l))}\right) \;=\; \frac{|X(k,l)|}{|Y(k,l)|}cos(\zeta) \quad (25)$$

where ζ is the difference of noisy and clean speech phases within each TF cell.

As shown in Fig. 23 (b), the phase spectrogram looks quite random since the wrapped phase values fall in $(-\pi, \pi]$. Thus, direct estimation of the phase spectrogram is intractable for DNN [6]. Hence, some studies like [12] considered the complex spectrogram's real and imaginary components as the training target and had the neural network directly estimate them. Since both components appear similar, except for a shift of $\pi/2$ radians, and possess a clear structure akin to the magnitude spectrogram, as shown in Fig. 23 (c) and (d), they are amenable to supervised learning. Furthermore, the similarity and correlation between the two components make it possible to estimate both components by a single neural network. From another perspective, the parameter sharing mechanism to simultaneously predict both components boosts learning and generalization capability. In particular, the authors of [12] and [114] improved the estimation of these real and imaginary components as two highly correlated subtasks through parameter sharing.

In [6], cIRM was suggested as the training target of the neural network. From $X(k,l) = M(k,l) \circ Y(k,l)$, the complex ratio mask $M(k,l)$ can be computed as,

$$M = \frac{Y_r X_r + Y_i X_i}{{Y_i}^2 + {Y_r}^2} + i\frac{Y_r X_i - Y_i X_r}{{Y_i}^2 + {Y_r}^2} \quad (26)$$

where r and i denote the real and imaginary components, and $\circ$ represents element-wise multiplication. Here, the argument (k,l) is discarded for brevity. The authors of [6] considered both real and imaginary components of M as two subtasks to be estimated by a single DNN to enhance

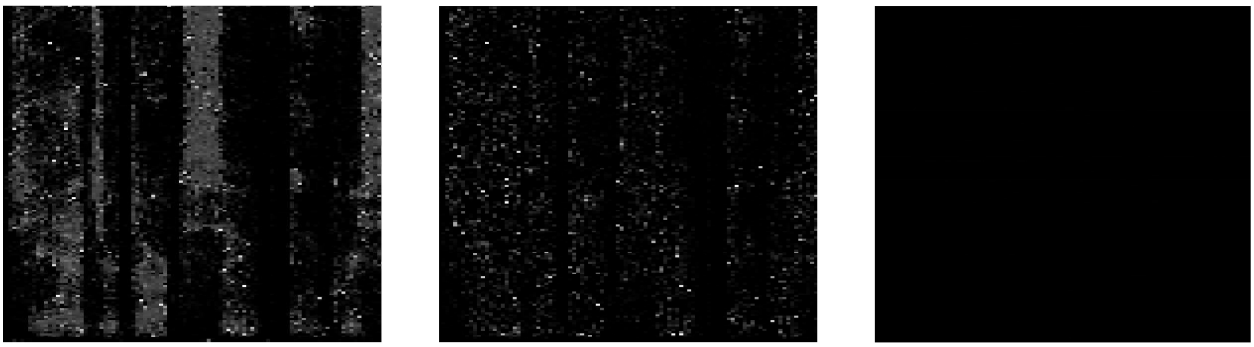


Figure 24: Spectrogram plot of (a) cIRM real part, (b) cIRM imaginary part, (c) Estimated cIRM imaginary part.

magnitude and phase simultaneously.

Different statements about whether mapping or masking performs better for speech enhancement can be found in the literature. In [66] , it is claimed that estimating cIRM outperforms direct estimation of real and imaginary components of a complex clean speech spectrogram for speech enhancement, while in [114] the advantage of direct estimation of a complex spectrogram over cIRM is stressed. These controversial comments likely stem from different DNNs and training datasets employed in these methods. An ideal cIRM can faithfully recover the complex spectrogram of clean speech and the clear structure of its real and imaginary components, as shown in Fig. 24 (a) and (b), making it amenable to supervised learning. However, the neural network is surprisingly unable to estimate the imaginary component of cIRM. In [115], the authors supposed that it is because of the lack of a learnable pattern in the imaginary component of the cIRM. Fig. 24 (c) shows the imaginary part of the cIRM estimated by a well-trained DNN. As shown, there is no information in the imaginary part. This complies with the argument made in [102] about the disability of DNN to estimate the imaginary component of cIRM, and supports the results shown in [116]. Meanwhile, the marginal advantage of cIRM over PSSA reported in [6] could result from using a different number of model parameters to estimate the training target. Based on the above observations, we are motivated to employ PSM as the training target in our proposed hybrid model.

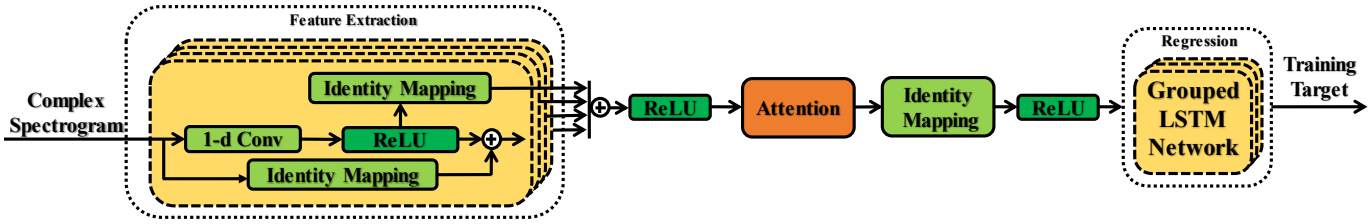


Figure 25: Detailed serial hybrid model.

3.2.4 Detailed Serial Hybrid Neural Network

As shown in Fig. 25, the first stage of the proposed hybrid model is to exploit a CNN with dilated 1D frequency convolution to extract an enriched set of input speech features. CNN's input is the real and imaginary parts of the noisy speech spectrogram. In this CNN, four 1D convolutional layers are stacked with an increasing dilation rate of two, i.e., $1, 2, 4,$ and 8. All kernel sizes are 1×7, and the number of channels for four layers is $16, 32, 16,$ and 8 in order for the CNN structure to be symmetric. ReLU is employed as the activation function. Residual learning is also applied to ease training by bypassing each layer's input to its output by an identity mapping layer, with 1×1 kernels, which fixes the number of channels. The output of the 1D convolution layer and the bypassed input signal are summed as input to the next layer. Each layer's skipped outputs are then forwarded using 32-channel identity mapping layers, and their summation is later fed to the attention block. It is worth mentioning that, instead of summation, the outputs could be stacked; but, we found that the summation here yields better results. Average and max pooling operations are simultaneously applied to the input feature maps of the attention block, and the results are concatenated and then combined using a convolutional layer with kernel size of 1×7 and sigmoid activation function. The input feature maps are reweighted by their multiplication with the output of this convolutional layer. The attention block's output is later fed to the last two-channel identity mapping layer. Consequently, both channels' outputs are reshaped and concatenated to be delivered to the LSTM network.

The LSTM network has three hidden layers, each comprising 256 units. Grouping strategy and representation rearrangement are applied between layers 2 and 3. A dynamic RNN is used

to perform fully-dynamic unrolling of inputs which speeds up the process in the sense that the input can have variable time steps. Here, the time step is the number of previous frames the cell used to compute the hidden state. Recurrent dropout is applied at a rate of 0.3 to mitigate the probable over-fitting problem. It is worth pointing out that because the number of parameters is limited compared to the high volume of the training dataset, the model learns merely basic data information. It means that the network does not suffer from over-fitting and is well generalized to unseen conditions.

Finally, a single affine dense layer transforms the LSTM network output to the PSM. On the one hand, choosing a mask as the training target addresses the global variance problem. As reported in [96], direct estimation of spectrogram causes an over-smoothing problem in the estimated signal compared to the reference signal, leading to a muffling effect, while a masking-based approach does not encounter this problem. On the other hand, mask estimation narrows the dynamic range of information that the network has to estimate. Besides, the advantage of PSM over other training targets in the hybrid model is demonstrated in Section 3.3.6.

3.3 Experimental Results

3.3.1 Experimental Setup

The proposed serial hybrid model is evaluated with TIMIT dataset. A 60-utterance subset is randomly selected from the dataset and kept aside for the testing stage, i.e., it is not used in training. Highly non-stationary noises from NOISEX-92 corpus, namely restaurant, babble, street, and factory, are selected to evaluate the model. Each noise file is divided into two parts, one for training and the other for testing, to ensure that the noise is unseen during the testing stage. We mix random chunks of the first part of the noises mentioned above with the clean utterances at different SNR levels. To this end, an energy ratio is defined as follows,

$$ER = \sqrt{\frac{X^2(t)}{N^2(t) \times 10^{\left(\frac{SNR}{10}\right)}}} \tag{27}$$

where $X(t)$ and $N(t)$ denote the energy of clean and noise signals, respectively. Then, the noisy signal is derived as below,

$$y(t) = x(t) + n(t) \times ER \tag{28}$$

The clean utterances are mixed with the noises at SNR levels of $-5, 0, 5,$ and 10 dB results in more than 100 k mixtures (6300 utterances $\times 4$ SNR levels $\times 4$ noises) for the training stage. In the testing stage, the unseen utterances are mixed with random cuts of the unseen noise part at unmatched SNR levels of $-6, 0, 6,$ and 12 dB, which gives 960 utterances (60 utterances $\times 4$ SNR levels $\times 4$ noises), half males and half females.

Furthermore, the proposed model is trained with 300 utterances from IEEE corpus mixed with the first half of 20 noises from NOISEX-92 (including airport, babble, buccaneer1, car, destroyerengine, destroyerops, exhibition, f16, factory, hfchannel, leopard, m109, machinegun, pink, restaurant, street, subway, train, volvo, and white) at SNR levels of $-5, 0, 5,$ and 10 dB, i.e., 24 k mixtures (300 utterances $\times 4$ SNR levels $\times 20$ noises.) Then, 50 unseen utterances mixed with random cuts of unseen part of different noises at unmatched SNR levels of $-6, 0, 6,$ and 12 dB, i.e. 4 k mixtures (50 utterances $\times 4$ SNR levels $\times 20$ noises) for the testing stage. Moreover, the model is evaluated with totally unseen noises named *Coffee Shop* and *Busy City Street* from [117] to show the generalization capability of the model to unseen noises at unmatched SNR levels.

The sampling rate is 16 kHz for all utterances segmented using the Hanning window with a frame length of 20 ms and 50% overlap between adjacent frames, i.e., 10 ms frame shift. A 320-point discrete Fourier transform (DFT) is computed where each frame consists of 160 samples.

The cost function is defined as the mean square error (MSE) to measure the difference between the mask-filtered and ground-truth spectrograms, as follows,

$$MSE = \frac{1}{LK} \sum_{l} \sum_{k} \left[\hat{M}(k,l) Y(k,l) - X(k,l) \right]^2 \tag{29}$$

where L and K are, respectively, the total number of time frames and that of frequency bins in each batch. An alternative is to measure the error between the estimated and ground-truth mask

values, as follows,

$$MSE = \frac{1}{LK} \sum_{l} \sum_{k} \left[M(k,l) - \hat{M}(k,l) \right]^2 \tag{30}$$

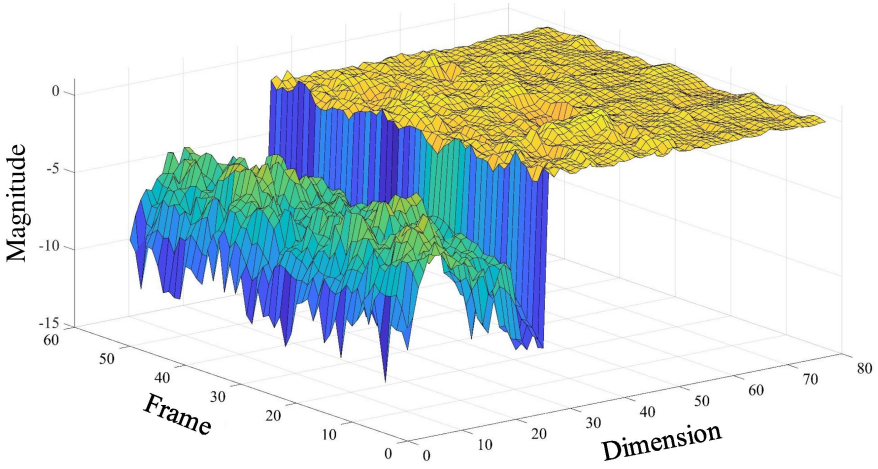
We have found that better results can be obtained if the cost function is defined between the estimated and ground-truth mask values. Adam optimizer [118] as an extension to stochastic gradient descent is used to update model parameter values during the training stage iteratively. The learning rate is initially 0.001, and then decays at a rate of 0.9 after each 1000 training steps. By each training step, we mean updating the model parameters based on the samples in one batch. In our experiments, each batch contains time frames of one noisy utterance. It is worth mentioning that these utterances have a minimum size of 100 and whatever maximum size. In case that the number of frames of an utterance is more than 500, we break it into more batches.

It is worth mentioning that all the experiments are performed using a single NVIDIA GeForce RTX 2080 GPU with 8 GB memory and a 2.2 GHz AMD Ryzen Threadripper 2920X 12-Core Processor. The average processing time of a 1-second utterance using the proposed model is around 8 milliseconds.

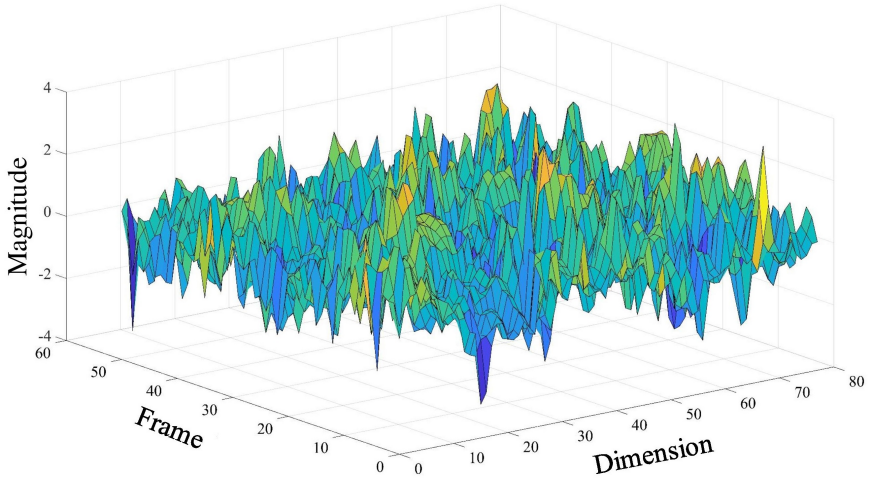
3.3.2 Feature Extraction and Analysis

Numerous acoustic-phonetic feature types have been introduced in the literature, and each could outperform others depending on the application. To investigate the impact of different inputs on the hybrid model performance, we evaluate the network with high-quality Gammatone-domain MRCG features [119], spectrum-based log Mel-filterbank energy features [69], and CNN-extracted features.

It is a common practice to concatenate original features (static) with their delta (first-order time derivative) and acceleration (second-order time derivative), called dynamic features, as they carry the temporal information of the static features [120]. As such, the dimension of log Mel-filterbank and MRCG features is 78 (26 static + 2×26 dynamic) and 768 (256 static + 2×256 dynamic), respectively. However, static and dynamic features appear in different ranges. Fig. 26 (a) shows



(a)



(b)

Figure 26: Input features visualization, (a) Log Mel-filterbank energy features concatenated with their delta and acceleration, (b) Normalized to zero mean and unit variance log Mel-filterbank energy features concatenated with their delta and acceleration.

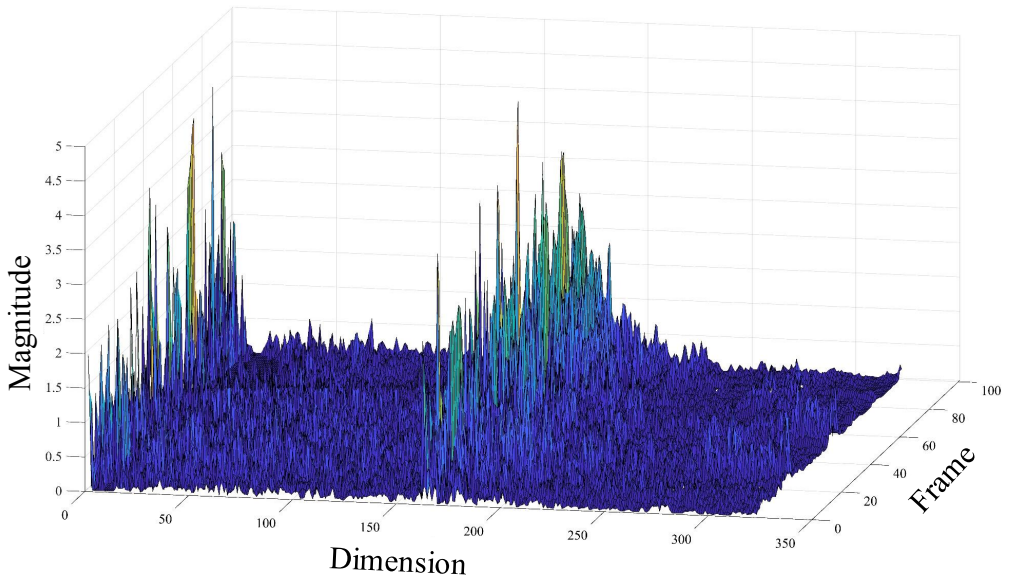


Figure 27: Features extracted by CNN.

log Mel-filterbank energy features concatenated with their delta and acceleration for several frames of a speech signal where there is a considerable gap between the difference of static and dynamic features in terms of mean and variance. To unify the values and also provide unbiased involvement of different elements of feature vectors, normalization to a standard range across all the features is required [99]. Input features are commonly normalized to zero mean and unit variance, as shown in Fig. 26 (b).

Besides, since the mapping is to be done by a neural network, we can let the network also decide what sort and combination of features are better to be exploited to improve the performance. To this end, we employed a CNN with dilated 1D and 2D frequency convolution with a kernel size of 1×7 and 5×7, respectively, to observe which one gives better performance. To get a perspective about how the CNN-extracted features look like, the features for several consecutive frames are shown in Fig. 27.

Fig. 28 shows the average PESQ score improvement resulting from using different features and illustrates a comparison of computational time, memory footprint, and the number of the model's whole parameters using different feature extraction methods. It is to be mentioned that time and memory are normalized to 1 for simplicity, and the number of model parameters is in million. Note

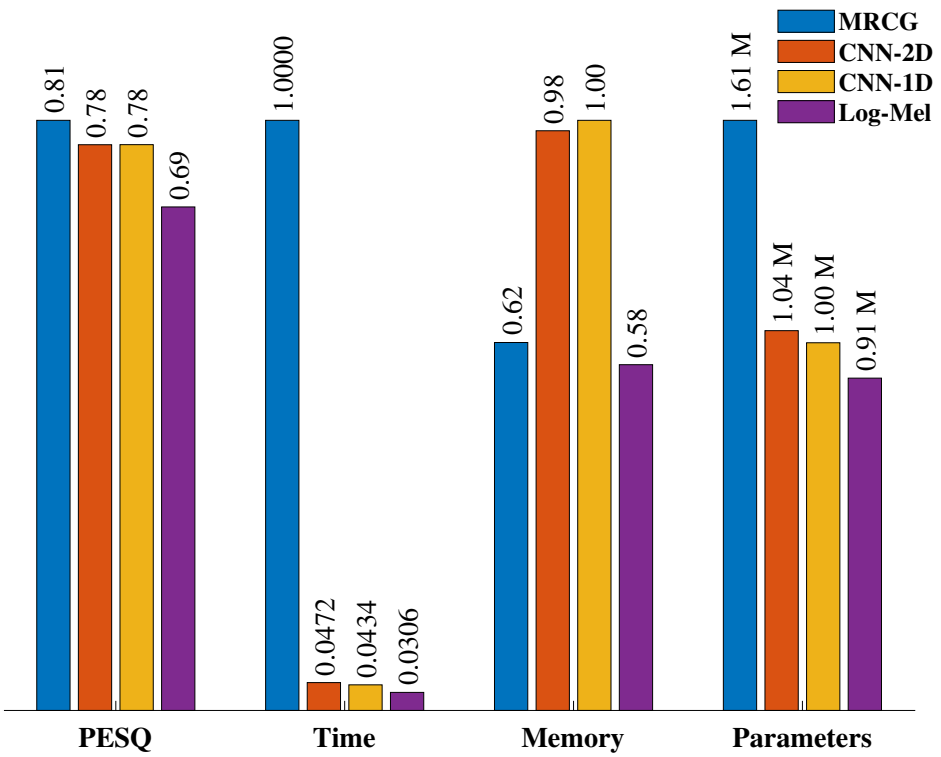


Figure 28: Feature comparison in terms of the average PESQ score improvement, computational time, memory, and number of parameters (in Million).

that the comparisons are performed using TIMIT dataset and four noises, namely, babble, factory, restaurant, and street, as mentioned in Section 3.3.1.

As shown in the figure, on the one hand, log Mel-filterbank energy features concatenated with dynamic features do not lead to satisfactory enhancement results in comparison with other experimented feature extraction methods. In contrast, in terms of computational time, memory footprint, and the number of parameters, they lead to the lowest. The reason is that these features do not bear the necessary and adequate information required for the network to establish an accurate mapping. On the other hand, MRCG features give very good results, indicating that this high-dimensional feature set carries a significant amount of information. Obviously, this feature set's high dimensionality leads to a high number of model parameters. Also, these features benefit from both local and contextual information as they are computed from four cochleagrams at different spectro-temporal resolutions with enriched information. However, Gammatone-domain feature extraction usually takes a long time. As shown in the figure, extracting MRCG features takes the longest

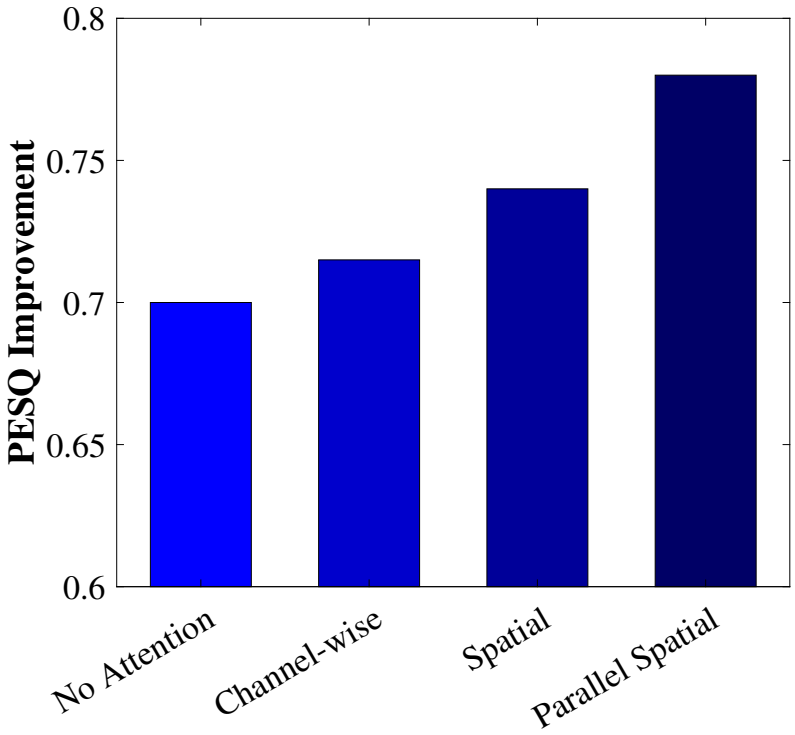


Figure 29: Comparison of different attention techniques in the hybrid model.

time.

Besides, the performance using CNN with 1D convolution is similar to that using CNN with 2D convolution, while it is better than using log-Mel features as CNN extracts the most appropriate features in our model. Exacting features using CNN takes less time than Gammatone-domain features like MRCG, requiring more memory for its computations. As seen in the figure, CNN with 1D convolution entails less time and parameters. As such, we can conclude that the best trade-off regarding performance, computational time, memory, and number of parameters is to extract features using a CNN with 1D convolution.

3.3.3 Benefit of Attention

As explained in Section 3.2.4, the CNN output contains 32 feature maps that will be sent to the attention block. The attention mechanism is to model the interdependencies among feature maps to boost their representative capability.

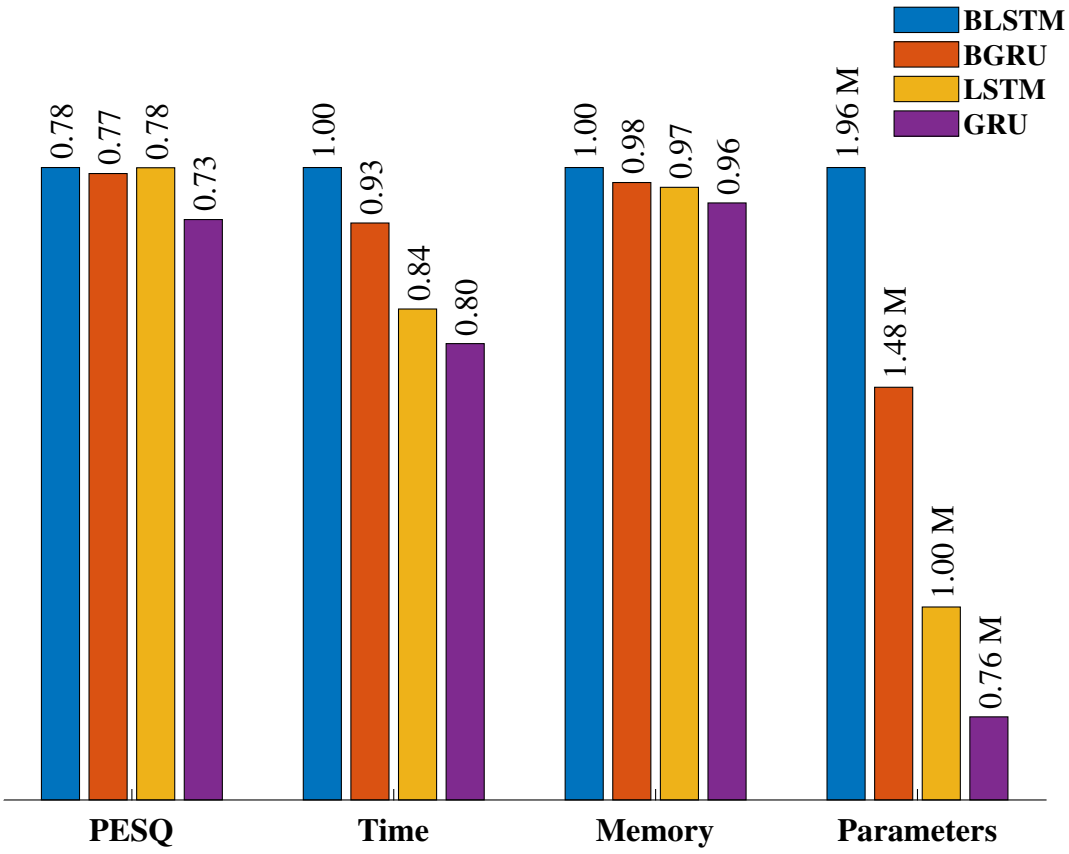


Figure 30: Comparison of different units in terms of the average PESQ score improvement, computational time, memory, and number of parameters (in Million).

As described in Section 3.2.1, three different attention mechanisms, namely, channel-wise, spatial, and parallel spatial, are investigated in the hybrid model. The comparison in terms of the average PESQ improvement is shown in Fig. 29. Clearly, using the attention technique improves the performance in general, and moreover, the parallel spatial attention outperforms the other two techniques. This is because the importance of the feature maps' pixels is emphasized through both average and max pooling operations. As such, we adopt the parallel spatial attention technique in the hybrid model.

3.3.4 Comparison of RNN Types

In this section, we aim to investigate the model performance using LSTM, BLSTM, GRU, and BGRU. All the networks are trained and tested with the same configuration, each comprising 3

Table 3: Comparison results of different Grouped RNN configurations.

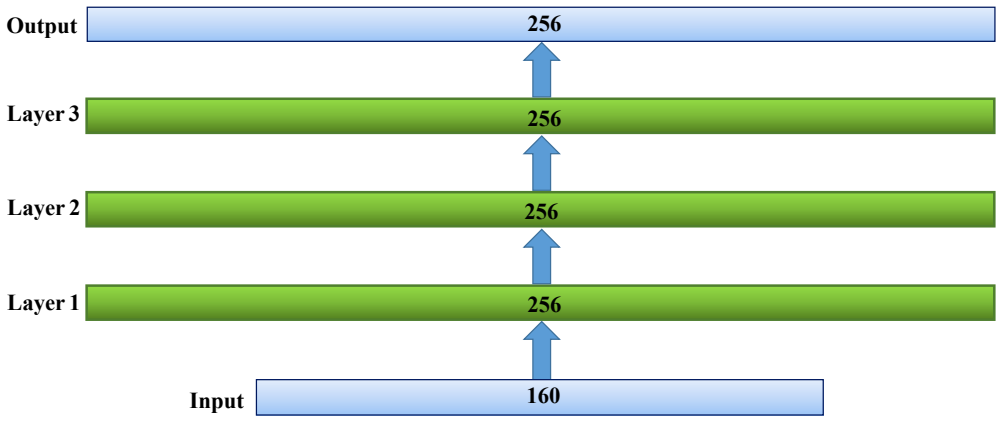
Model	a	b	c	d	e
Avg. PESQ improvement	0.64	0.61	0.54	0.63	0.56
No. Parameters (Million)	1.53	0.79	0.42	1.00	0.74

hidden layers of 256 units. The training and testing datasets are as mentioned in Section 3.3.2. Figure 30 shows the average of PESQ improvement for different noises and SNR levels, as well as computational time, memory footprint, and the number of parameters. As shown, GRU does not yield satisfactory results PESQ-wise, while in terms of other measurements, it achieves the lowest. BLSTM, BGRU, and LSTM yield almost the same results in terms of PESQ score, while the number of model parameters using BLSTM and BGRU is roughly twice and 1.5 times than LSTM. Consequently, BLSTM and BGRU take longer computational time and entail more memory than LSTM. Hence, we can conclude that LSTM is the most appropriate RNN variation for mask estimation in the proposed model.

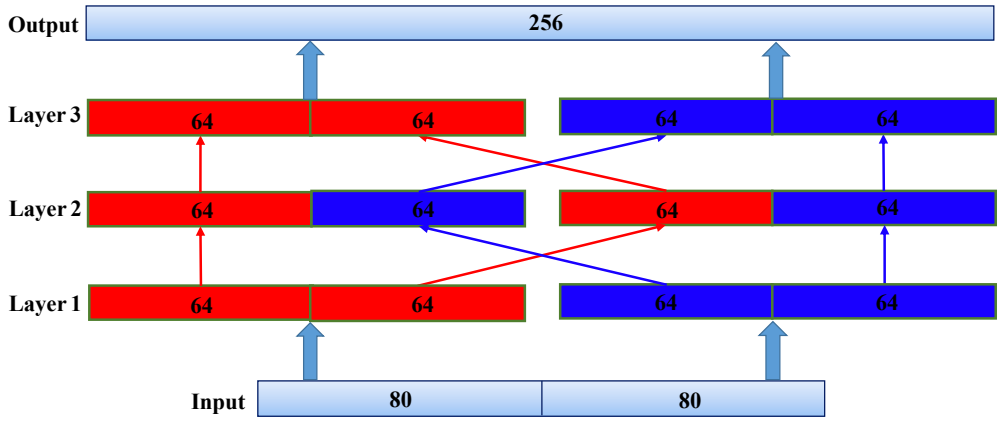
3.3.5 Comparison of Evaluation of Different Grouping Strategies

In this section, we evaluate five LSTM network configurations as shown in Fig. 31 in terms of the PESQ score of the results and the number of model parameters to find the best trade-off for our model. Fig. 31 (a) shows a standard three-layer LSTM structure with 256 units per layer where no grouping strategy is adopted. Fig. 31 (b) and (c) illustrate the same network using a grouping strategy where both input and hidden layers are split into 2 or 4 groups, respectively, and representation rearrangement is applied to the hidden layers. Fig. 31 (d) and (e) show similar architectures, but the grouping strategy and representation rearrangement are only adopted between layers 2 and 3, respectively.

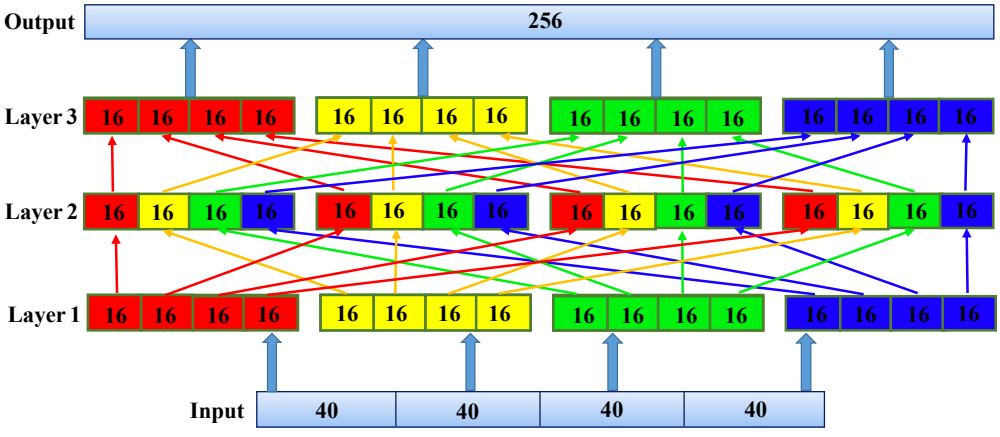
The comparison results, in terms of the average quality and the number of parameters, are shown in Table 3. The training and testing datasets are as mentioned in Section 3.3.2. As illustrated, a standard LSTM network (a) yields an average PESQ score of 0.64 with 1.53 M parameters, while using the grouping strategy only between layers 2 and 3 (d) not only does yield roughly the


Output
256
Layer 3
256
Layer 2
256
Layer 1
256
Input
160

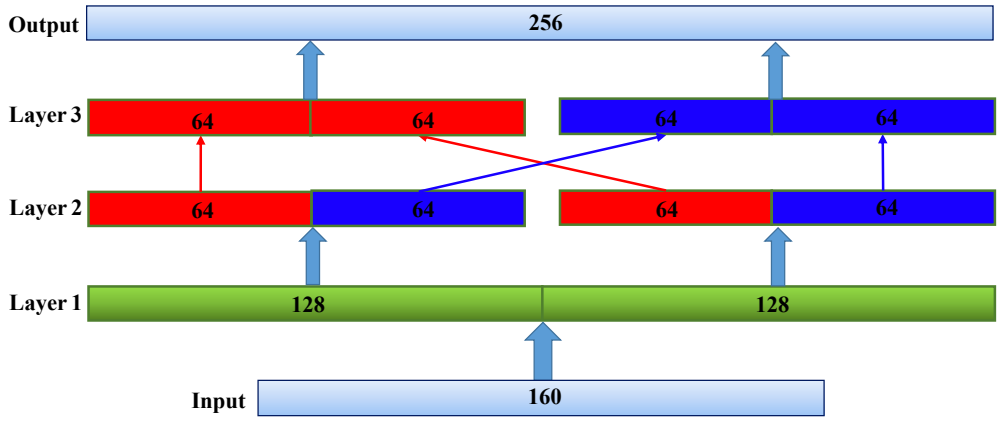
(a)


Output
256
Layer 3
64
64
64
64
Layer 2
64
64
64
64
Layer 1
64
64
64
64
Input
80
80

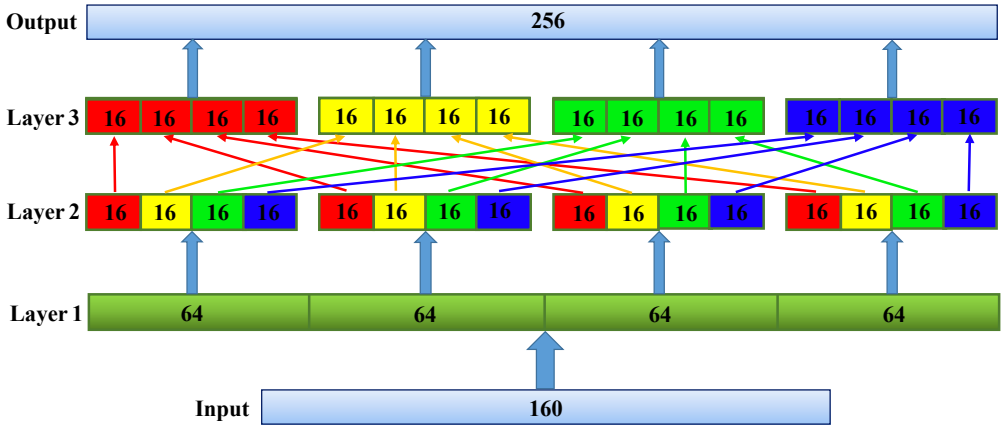
(b)


Output
256
Layer 3
16
Layer 2
16
Layer 1
16
Input
40
40
40
40

(c)



(d)



(e)

Figure 31: LSTM network with grouping strategy and representation rearrangement, (a) a standard LSTM network, (b) LSTM network with 2 groups and representation rearrangement for input and all layers, (c) LSTM network with 4 groups and representation rearrangement for input and all layers, (d) LSTM network with 2 groups and representation rearrangement between layers 2 and 3, (e) LSTM network with 4 groups and representation rearrangement between layers 2 and 3.

same results concerning quality but also cuts the number of whole model parameters by 35%. Also, using the grouping strategy between every contiguous layer (b) gives 0.61 for quality with only 0.79 M parameters which means the number of whole model parameters is cut by 52%. As shown, grouping by 4 does not give good results despite whether grouping for all layers (c) or two layers (e). In this paper, we choose the grouping strategy with two groups between layers 2 and 3 (d).

3.3.6 Training Targets Comparison and Analysis

Label Compression

A neural network would be better and easier trained if input and output are in the same range. Since the mask values (equation (25)) might have a wide range, a compression function should be adopted to make these values amenable to a neural network. The most straightforward compression method might be to limit the values within $[-1, 1]$. This technique's problem is that some mask values can go very high because of a small denominator. As such, normalization to unity with respect to these large values will result in undesired TF cells' over-compression. Other methods are hyperbolic tangent, and a variation of it introduced in [121] which we call QC, as shown in Fig.32 (a) and (b). Figure32 (c) illustrates a slice of the label vector showing how different compression techniques influence label magnitude. As shown, employing hyperbolic tangent compression gives a better resolution while limits the label values to -1 and 1. To show the impact of label compression on the enhancement performance, we evaluated the hybrid model with different compression methods to compare the average PESQ score improvement. As shown in Fig. 32 (d), hyperbolic tangent gives the best results for our model.

Comparison of Different Training Targets

As mentioned in Section 3.2.3, there are different claims in the literature about which training target is preferred for a DNN-based speech enhancement. As such, we compare different training targets

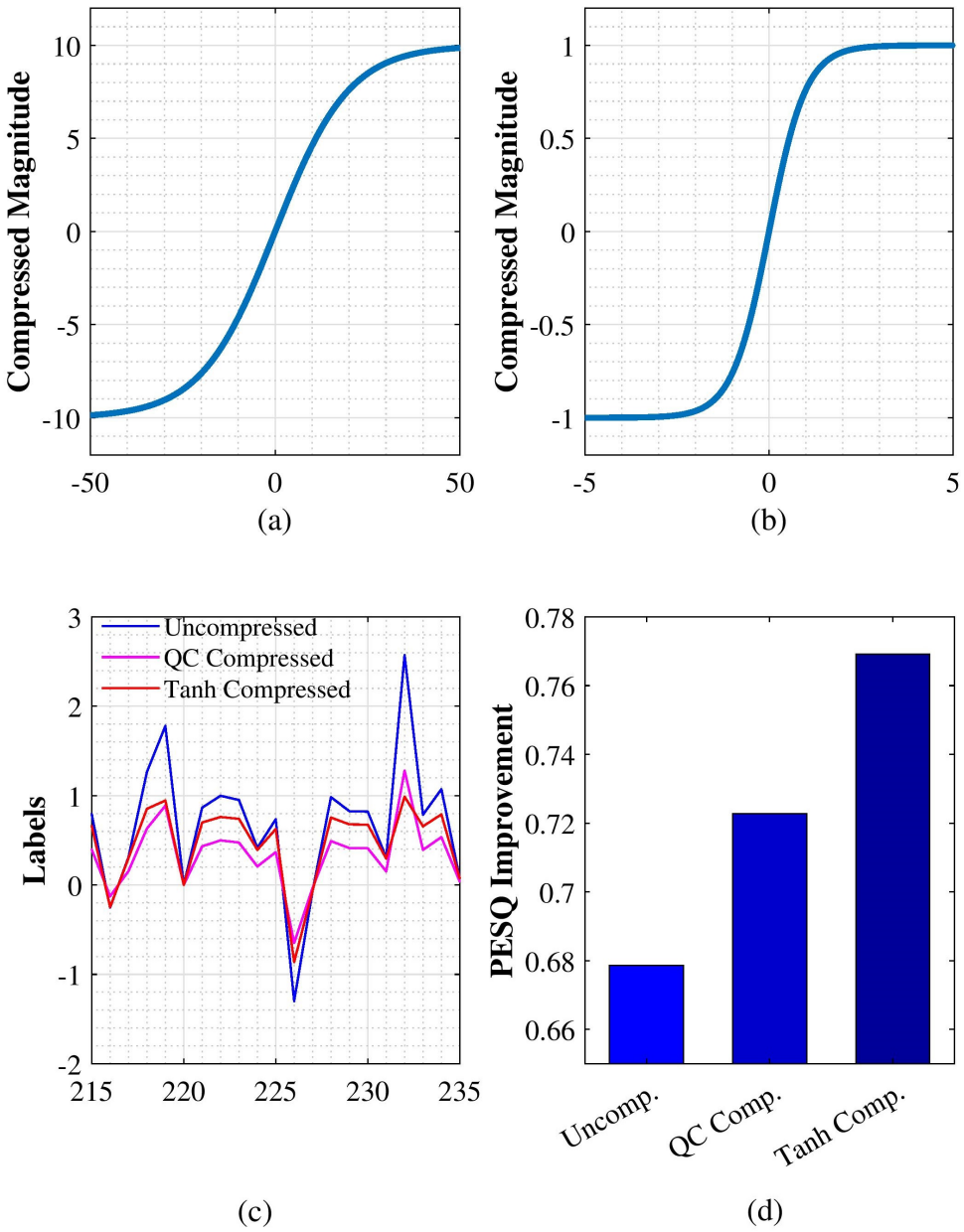


Figure 32: Label compression. (a) QC compression methods, (b) hyperbolic tangent, (c) a cut of mask values, (d) Average PESQ score improvement of different compression methods.

including IRM [66], cIRM [6], complex spectrogram (CS) [12], and PSM [68] in terms of average PESQ score improvement in the hybrid model with IEEE dataset and 20 noises, as mentioned in Section 3.3.1. As known, all these training targets consider phase information alongside magnitude enhancement except for IRM.

Figure 33 shows the results of comparison. Comparing IRM with other training targets reveals

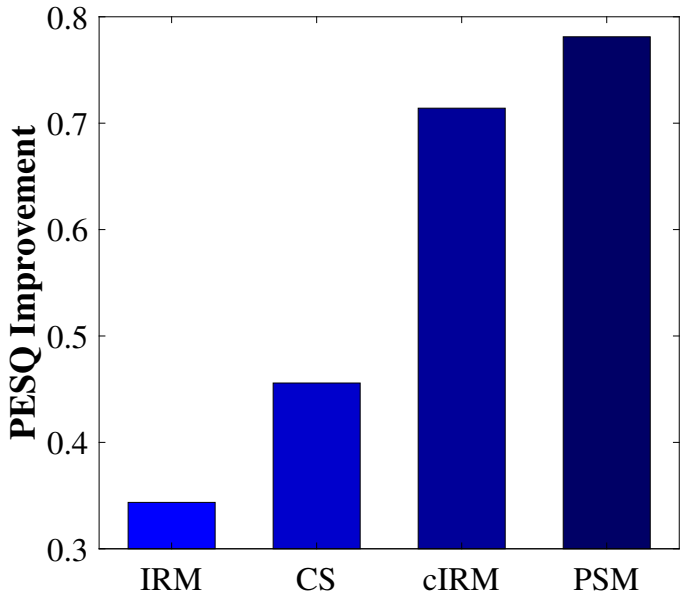


Figure 33: Comparison of training targets with hybrid model.

the advantage of incorporating phase information and its direct impact on the quality of results. Also, we can see that the quality improvement using cIRM as a mask is better than the direct estimation of complex spectrogram using the hybrid model. It is because complex spectrogram estimation is more challenging as the network has to precisely estimate every single element of the complex spectrogram, leading to more cumulative error while the network amounts to a subset of TF cells in the cIRM case. However, the hybrid model using PSM performs the best compared to using other training targets, while the number of model parameters using PSM is almost 5% less than using complex spectrogram and cIRM. This reduction in the number of model parameters stems from the PSM training target size, which is one-half of that of other training targets.

3.3.7 Comparison with other DNN-based Methods

We compare the proposed model with some other mapping- and masking-based techniques. For brevity's sake, we call different methods based on their training targets. FFT-Mag and target magnitude spectrum (TMS) are two direct mapping-based methods introduced in [66] and [96], respectively. Both methods use FC networks with three hidden layers with 1024 and 2048 units per layer,

Table 4: The number of trainable parameters in each method (in Million).

Method	FFT-Mag	TMS	IRM	SMM	PSSA	cIRMC	cIRM	Proposed
Number of Parameters	2.66	12.35	2.66	2.66	0.91	0.99	2.82	1.00

respectively. The former captures 5 frames to exploit contextual information and uses a set of complementary features as the neural network input, while the latter uses 11 frames and log-power spectral magnitude as the neural network input. FFT-MAG and TMS predict the STFT magnitude and log-power spectral magnitude of clean speech, while both methods utilize the noisy phase to resynthesize the clean speech.

SMM, IRM, and cIRM are three masking-based methods first introduced in [66] and [6], each tested with a 3 hidden layer network and 1024 units in each layer. For all of them, 5 frames are used to capture temporal contextual information, and the input is a complementary set of features. cIRM estimation is also performed using a composite model in [98]. The model uses Mel-frequency cepstral coefficients (MFCCs) features and STFT of the noisy speech as input, and a parallel model. We call this model cIRMC in comparison tables. Also, a two-layer LSTM is used for phase-sensitive spectrum approximation (PSSA) in [68]. This network's input is 100-bin log-Mel filterbank features, and a sigmoid function is used as the activation function of the output FC layer. IRM, SMM, and PSSA resynthesize the estimated speech signal with the noisy phase. All networks are evaluated with the same datasets, noises, and SNR levels for a fair comparison.

The number of trainable parameters in each method is presented in this table 4. As illustrated, the number of trainable parameters of the proposed framework is less than that of other models, except for PSSA and cIRMC. The methods are evaluated on the TIMIT dataset and four noises, as mentioned in Section 3.3.1. Tables 5, 6, 7, and 8 present performance scores of the mentioned methods for different noises and SNR levels where BBE, FTRY, STRT, and RTRT denote babble, factory, street, and restaurant, respectively. The top number in each table cell represents the average PESQ score with all aforementioned noises at different SNR levels for males and the bottom one for females. As shown, the proposed framework prioritizes other models for every noise at almost

Table 5: Performance of Different Methods at -6 dB.

Method	PESQ				SSNR			
	BBE	**FTRY**	**STRT**	**RTRT**	**BBE**	**FTRY**	**STRT**	**RTRT**
Unprocessed	1.24	1.10	1.22	1.29	-10.26	-9.65	-9.48	-9.57
	0.94	0.83	0.98	0.90	-9.50	-9.14	-9.09	-8.97
FFT-Mag	1.57	1.78	2.10	1.72	-0.25	0.38	1.39	0.34
	1.22	1.34	1.54	1.22	0.03	0.20	1.35	0.46
TMS	1.49	1.62	1.88	1.61	0.05	0.27	1.33	0.38
	1.29	1.37	1.67	1.31	0.45	0.60	1.77	0.80
IRM	1.61	1.73	2.11	1.91	-3.25	-1.95	1.05	0.06
	1.29	1.34	1.66	1.46	-2.60	-1.77	1.21	0.67
SMM	1.61	1.72	2.05	1.86	-2.70	-1.79	-0.19	-1.25
	1.23	1.32	1.66	1.37	-2.36	-1.71	0.42	-0.85
PSSA	1.56	1.73	2.06	1.74	-1.94	-0.47	1.75	0.43
	1.22	1.42	1.73	1.35	-1.59	-0.22	2.37	0.78
cIRMC	1.73	1.84	2.28	**1.89**	-0.29	0.20	2.53	0.74
	1.53	1.62	2.05	1.61	0.07	0.42	2.90	1.48
cIRM	1.60	1.81	2.30	1.94	-0.14	0.79	2.19	0.66
	1.35	1.50	1.90	1.60	-0.16	0.66	2.51	1.05
Proposed	1.69	**1.85**	**2.34**	**2.03**	**0.47**	**0.80**	**2.94**	**1.44**
	1.59	**1.75**	**2.11**	**1.76**	**1.03**	**1.25**	**3.46**	**2.10**

all SNR levels regarding PESQ score. With reference to SSNR, the proposed model outperforms other models at SNR levels of -6 and 0 dB, while IRM shows higher scores at SNR 6 dB and IRM and PSSA yield slightly better results at 12 dB SNR levels.

We also evaluated the aforementioned methods using the IEEE corpus where they are trained with TIMIT dataset at unmatched SNR levels. Results can be seen in Table 9, where PESQ and SSNR scores of the proposed model are higher than others, except for SMM that outperforms others at SNR level of -6 dB.

Table 10 shows a comparison of different models trained with IEEE corpus mixed with 20 different noises at unmatched SNR levels. Obviously, the proposed model again outperforms others in almost all the cases except for the SNR level of 12 dB, where PSSA yields marginally better results. Furthermore, the model is evaluated with unmatched utterances mixed with unseen noises, *Coffee Shop* and *Busy City Street* represented by CF and BCS, respectively, at unmatched SNR levels. The results are shown in Table 11. Clearly, the proposed model outperforms other methods in terms of both PESQ and SSNR scores.

Table 6: Performance of Different Methods at 0 dB.

Method	PESQ				SSNR			
	BBE	**FTRY**	**STRT**	**RTRT**	**BBE**	**FTRY**	**STRT**	**RTRT**
Unprocessed	1.64	1.52	1.74	1.64	-5.33	-5.20	-4.90	-4.82
	1.38	1.30	1.48	1.32	-4.95	-5.01	-4.77	-4.71
FFT-Mag	2.10	2.16	2.43	2.19	1.63	1.99	2.54	1.81
	1.61	1.73	1.86	1.66	1.40	1.71	2.40	1.88
TMS	1.92	2.06	2.28	2.04	1.90	2.16	2.85	1.89
	1.72	1.86	2.07	1.77	2.17	2.55	3.39	2.53
IRM	2.22	2.30	2.66	2.42	1.59	2.66	4.80	3.49
	1.86	1.95	2.29	2.01	2.07	2.82	5.45	4.11
SMM	2.13	2.20	2.43	2.27	0.89	1.78	2.68	1.72
	1.79	1.86	2.13	1.87	1.20	1.93	3.69	2.32
PSSA	2.12	2.27	2.56	2.27	1.93	2.97	4.51	3.12
	1.86	1.98	2.25	1.91	2.50	3.14	5.25	3.72
cIRMC	**2.30**	**2.40**	2.77	2.41	2.84	3.12	4.91	3.52
	2.08	2.15	2.48	2.12	3.12	3.32	5.33	4.15
cIRM	2.21	2.28	2.70	2.45	2.95	3.34	4.74	3.51
	1.90	2.02	2.35	2.06	3.04	3.34	4.93	3.73
Proposed	**2.30**	**2.40**	**2.83**	**2.50**	**3.46**	**3.68**	**5.36**	**4.01**
	2.12	**2.21**	**2.54**	**2.26**	**3.71**	**4.06**	**5.93**	**4.38**

Table 7: Performance of Different Methods at 6 dB.

Method	PESQ				SSNR			
	BBE	**FTRY**	**STRT**	**RTRT**	**BBE**	**FTRY**	**STRT**	**RTRT**
Unprocessed	2.10	1.99	2.23	2.06	0.34	0.16	0.82	0.65
	1.88	1.78	2.01	1.82	0.30	0.28	0.81	0.67
FFT-Mag	2.50	2.52	2.65	2.56	3.13	3.25	3.50	3.33
	1.97	1.99	2.06	1.96	2.84	2.90	3.26	2.94
TMS	2.31	2.40	2.58	2.40	3.64	3.89	4.22	3.68
	2.17	2.25	2.38	2.17	3.99	4.27	4.52	4.12
IRM	2.79	2.86	3.10	2.95	6.16	**6.71**	7.78	**6.95**
	2.44	2.54	2.83	2.58	6.59	**7.07**	**8.70**	**7.48**
SMM	2.59	2.64	2.84	2.65	4.56	5.32	5.95	5.14
	2.29	2.33	2.53	2.29	5.15	5.59	6.72	5.70
PSSA	2.73	2.78	3.04	2.80	5.97	6.41	7.76	6.43
	2.41	2.49	2.67	2.40	6.47	6.55	8.23	6.84
cIRMC	**2.87**	2.89	3.16	2.90	6.33	6.38	7.32	6.44
	2.61	2.66	2.89	2.61	6.60	6.59	8.08	6.76
cIRM	2.82	2.82	3.10	2.92	6.03	6.24	7.27	6.45
	2.49	2.57	2.79	2.54	5.64	5.95	7.29	6.19
Proposed	2.82	**2.90**	**3.22**	**2.96**	**6.43**	6.63	**7.82**	6.85
	2.66	**2.70**	**3.01**	**2.72**	**6.67**	6.83	8.57	7.32

Table 8: Performance of Different Methods at 12 dB.

Method	PESQ				SSNR			
	BBE	**FTRY**	**STRT**	**RTRT**	**BBE**	**FTRY**	**STRT**	**RTRT**
Unprocessed	2.55	2.45	2.71	2.51	6.05	6.09	6.77	6.40
	2.35	2.28	2.51	2.31	6.00	5.95	6.76	6.36
FFT-Mag	2.71	2.74	2.80	2.76	4.13	4.24	4.08	4.24
	2.14	2.14	2.16	2.14	3.59	3.64	3.72	3.75
TMS	2.66	2.66	2.79	2.69	5.23	5.24	5.34	5.21
	2.47	2.55	2.64	2.50	5.28	5.56	5.53	5.19
IRM	3.28	3.33	3.47	3.33	9.35	9.95	10.04	9.40
	3.05	3.13	3.27	3.05	**10.24**	**10.71**	11.02	**10.26**
SMM	3.05	3.04	3.26	3.05	8.90	9.41	9.98	9.28
	2.76	2.82	2.97	2.74	9.21	9.61	10.49	9.55
PSSA	3.23	3.19	3.38	3.21	**9.97**	**10.08**	**10.89**	**9.93**
	2.90	2.96	3.12	2.89	10.07	10.23	**11.47**	10.20
cIRMC	3.30	3.09	3.28	3.03	9.13	9.19	9.78	9.21
	2.02	3.00	3.17	2.98	9.25	9.40	10.46	9.52
cIRM	3.26	3.27	3.48	3.31	8.80	8.81	9.78	9.02
	2.94	3.00	3.17	2.98	8.08	8.23	9.29	8.31
Proposed	**3.31**	**3.35**	**3.55**	**3.40**	9.58	9.66	10.29	9.89
	3.18	**3.16**	**3.35**	**3.14**	9.76	9.63	10.80	9.94

Table 9: Average SSNR and PESQ Score of Different Methods Trained with TIMIT Dataset and Tested with IEEE Corpus.

Method	PESQ				SSNR			
	-6	**0**	**6**	**12**	**-6**	**0**	**6**	**12**
Unprocessed	1.36	1.73	2.12	2.51	-9.50	-5.34	0.04	5.83
FFT-Mag	1.64	1.97	2.25	2.45	0.06	0.93	1.53	1.92
TMS	1.62	1.90	2.19	2.43	0.62	1.83	2.86	3.67
IRM	1.78	2.16	2.58	2.98	-0.88	2.24	4.96	6.85
PSSA	1.73	2.12	2.51	2.90	-0.39	2.15	4.59	7.00
SMM	**1.81**	2.12	2.44	2.74	-1.26	1.45	4.33	7.41
cIRMC	1.69	2.13	2.52	2.93	0.30	2.57	4.35	5.96
cIRM	1.80	2.19	2.60	2.96	0.18	2.23	4.15	5.89
Proposed	1.73	**2.21**	**2.68**	**3.10**	**0.86**	**3.08**	**5.31**	**7.47**

Table 10: Average SSNR and PESQ Score of Different Methods evaluated with IEEE Corpus.

Method	PESQ				SSNR			
	-6	**0**	**6**	**12**	**-6**	**0**	**6**	**12**
Unprocessed	1.40	1.74	2.14	2.55	-7.79	-3.90	1.40	7.21
FFT-Mag	1.95	2.39	2.76	3.00	1.49	3.53	5.14	6.18
TMS	1.87	2.34	2.73	3.01	1.47	3.74	5.56	6.80
IRM	1.87	2.45	3.00	3.39	-0.81	4.20	8.46	11.65
PSSA	1.90	2.46	2.97	3.37	0.71	4.84	8.47	**11.86**
SMM	1.84	2.32	2.76	3.14	-0.96	2.82	6.37	10.13
cIRMC	2.03	2.56	3.05	3.44	2.02	5.14	8.07	10.88
cIRM	1.90	2.44	2.95	3.34	1.81	4.66	7.50	10.15
Proposed	**2.13**	**2.66**	**3.11**	**3.48**	**2.95**	**5.90**	**8.93**	11.68

Table 11: SSNR and PESQ Score of Different Methods where unseen utterances are mixed with unseen noises at unmatched SNR Levels.

Method	PESQ								SSNR							
	-6		**0**		**6**		**12**		**-6**		**0**		**6**		**12**	
	CF	**BCS**	**CF**	**BCS**	**CF**	**BCS**	**CF**	**BCS**	**CF**	**BCS**	**CF**	**BCS**	**CF**	**BCS**	**CF**	**BCS**
Unprocessed	1.36	1.31	1.70	1.80	2.11	2.25	2.54	2.71	-8.24	-8.10	-4.41	-4.10	0.93	1.00	6.60	6.90
FFT-Mag	1.49	1.92	2.04	2.40	2.60	2.78	2.97	3.04	-0.76	0.98	2.17	3.01	4.65	4.92	6.24	6.28
TMS	1.46	1.96	1.97	2.38	2.49	2.77	2.89	3.06	-0.63	0.79	2.28	3.15	4.68	5.18	6.42	6.73
IRM	1.52	1.80	2.09	2.42	2.70	2.99	3.19	3.38	-3.52	-1.72	1.18	2.91	6.34	7.21	10.51	10.77
PSSA	1.46	1.90	2.08	2.49	2.68	2.99	3.21	3.40	-2.03	0.04	2.39	3.61	6.48	7.25	10.34	10.89
SMM	1.52	1.83	2.06	2.29	2.55	2.78	2.99	3.19	-3.15	-1.53	0.77	2.24	4.76	6.08	8.93	10.05
cIRMC	**1.74**	2.01	2.26	2.53	2.82	3.05	3.28	3.45	**0.03**	1.45	3.72	4.51	7.16	7.56	10.30	10.61
cIRM	1.42	1.95	2.03	2.48	2.69	3.01	3.22	3.39	-0.12	0.89	2.97	3.93	6.13	6.85	9.34	9.72
Proposed	1.68	**2.12**	**2.33**	**2.68**	**2.84**	**3.12**	**3.30**	**3.50**	-0.12	**2.31**	**3.75**	**5.09**	**7.22**	**7.90**	**10.55**	**11.07**

3.4 Conclusion

In this chapter, a serial hybrid model based on the integration of CNN and LSTM was proposed for speech enhancement. First, CNN was employed to extract the most appropriate features from the speech spectrogram. An attention technique is adopted to recalibrate the CNN feature maps. A grouped LSTM network structure was then exploited to map the CNN-extracted features to a PSM training target to benefit from strong temporal dependencies of speech while keeping the complexity low.

CNN as a feature extractor was compared with some high-quality conventional acoustic features to demonstrate CNN's advantage at feature extraction. Also, the most common RNN variations have been considered for the mapping part in the proposed model, where the LSTM was shown to be the best trade-off in terms of the performance, computational time, memory footprint, and the number of model parameters. We also evaluated different grouping strategies within the LSTM to find the hybrid model's best performance.

Moreover, various training targets were compared in the hybrid model to demonstrate the advantage of PSM, which takes into account both magnitude and phase information in the enhancement process.

Finally, the proposed model is compared with some well-known DNN-based speech enhancement methods, showing significant improvement in speech enhancement in the presence of highly non-stationary noise at different SNR levels. It was also shown that the hybrid model has a smaller number of model parameters as compared to some related models in the literature.

Chapter 4

A Low-Complexity Phase-Aware Parallel Deep Neural Network for Speech Enhancement

4.1 Introduction

In the previous chapter, a hybrid serial neural network was introduced, where a CNN extracted the features, and then an LSTM performed the transformation between these features and a phase-sensitive mask. Despite the fact that this model achieved satisfactory speech enhancement results, there is still room for improvement in terms of the number of parameters of the model, computational complexity, and memory footprint. Moreover, the model can be designed in parallel to speed up the process. In addition, the phase information can be more specifically studied and involved in the enhancement process. A brief literature review about two topics will be presented in the following. Specifically, DNN classes and their complexity and some well-known DNN-based speech enhancement methods along with their limitations will be presented. Furthermore, we will categorize and discuss the well-known available phase-aware methods in the literature along with their shortcomings.

Park *et al.* [5] investigated CNN for speech enhancement and compared its required number of parameters with that of FC and LSTM. In particular, they showed that these three methods almost give the same speech enhancement performance, although CNN requires a smaller number of parameters. However, this study only considers the number of parameters, while the actual complexity and implementation cost also depend on the memory footprint, which can be significantly larger for CNN than for LSTM and FC. As mentioned in the previous chapter, CNN was originally conceived to capture local information from an image, while speech spectrograms generally exhibit non-local correlations. Moreover, the CNN network's max-pooling layers only retain the coarse information of its input. Consequently, a generative model with no max-pooling layer but containing a stack of dilated causal convolutional layer instead was introduced in [11]. This model expands the CNN filters' receptive field without adding more complexity to the model. Inspired by this work, a fully convolutional model in the frequency domain was introduced in [12] showing promising speech enhancement results.

In contrast to the aforementioned stand-alone methods, some recent studies have considered a combination of networks as the learning engine for speech enhancement. Tan *et al.* [122] introduced a convolutional recurrent neural network (CRN) as an encoder-decoder network for speech enhancement. They also extended CRN by introducing a gated convolutional recurrent network and obtained better speech enhancement results in [113]. Some other CRN-based networks operating in the frequency and time domain were proposed in [123], and [124], respectively. Although the CRN model yields good speech enhancement results, Strake *et al.* [125] argued that the internal relations and local structures of CNN feature maps are ravished due to the reshaping of data among different CRN components. Thus, they employed convolutional LSTM for speech enhancement, where the fully-connected mappings in LSTM are replaced with convolutional mappings. Based on this argument, another model block named *gruCNN* was recently utilized for speech enhancement in [126], where recurrency is added to feature extracting CNN layers. These combined networks achieved good speech enhancement results; however, they all exhibit very high complexity models, and some of them (due to their non-causal formulation) introduce additional latency. Moreover,

CRN-based methods perform well when the training and testing datasets are the same but break down on unseen datasets [127].

Besides, researchers have proposed several different phase-aware methods. For the DNN-based methods, the earliest attempt was to incorporate the phase information into the magnitude processing, in which the phase spectrum can be considered as implicitly enhanced. Two phase-sensitive masks are proposed in [68, 128], where the mask is defined with the information of the difference between the noisy phase spectrogram and the clean one. However, these methods still process the magnitude only and employ the noisy phase in speech reconstruction.

The second kind of phase-aware methods resorts to processing the real and imaginary part of the speech spectrogram to avoid the difficulty of phase processing. A complex IRM was introduced in [6] where the mask is divided into real and imaginary components to enhance complex spectrogram. Unfortunately, the cIRM algorithm suffers distortion in practice because the imaginary part of the cIRM has no notable trainable pattern and thus is difficult to be estimated as proven in [102, 116]. The direct estimation of the complex spectrogram was then proposed in [12, 104, 113, 122], where the DNN is employed to estimate the real and imaginary parts of the complex spectrogram of the clean speech from those of noisy speech. However, these methods require large datasets to build an accurate mapping function; otherwise, the performance might be worse than a simple spectral magnitude mapping method on unseen databases [127].

The third kind of phase processing aims to recover the phase spectrogram directly. For instance, the authors in [54, 129] adopted a harmonic model to reconstruct the phase spectrogram and achieve better enhancement results over their counterparts without phase reconstruction. Yin *et al.* [102] introduced a phase and harmonics-aware model for noise reduction, where a novel two-stream DNN architecture with information exchange between the magnitude and phase spectra is proposed to recover the complex spectrogram of the clean speech. Since the phase spectrogram itself has an irregular structure, researchers also use the phase variations in phase reconstruction, as these variants exhibit a similar structure as magnitude [130]. Consequently, Zheng *et al.* [131] presented a phase-aware model to joint process the magnitude and phase spectrogram, where the

estimated magnitude is obtained with a spectral mask, and the phase is reconstructed through a phase derivative (PD): i.e., instantaneous frequency derivation (IFD). Experimental results demonstrate that this phase-aware model performs better than both magnitude-only mask and cIRM algorithm, but the limitations of this work are also obvious: 1) FC and LSTM networks are used in the estimation, which might not achieve the most accurate training targets; 2) the paper only used IFD in phase reconstruction; however, there are other phase derivatives such as group delay (GD) should be investigated.

In this chapter, we propose a low-complexity composite model in which carefully designed LSTM and CNN are integrated to extract a complementary set of features by taking full advantage of the temporal and spectral dependencies of input speech. LSTM and CNN perform independently and in parallel to speed up the computation, thereby addressing fundamental concerns from the aspects of real-time processing, limited latency, and low complexity in speech enhancement applications. Moreover, inspired by [131], we present a new model called phase-aware composite deep neural network (PACDNN) that involves two subtasks: magnitude processing with a spectral mask and phase reconstruction with PD, where a DNN estimates both training targets simultaneously. We investigate different types of masks and PDs as well as their possible combinations to select the best training targets for the DNN. Our analysis and experimental studies reveal that the proposed PACDNN model yields a significantly improved speech enhancement performance compared to several existing DNN based methods while exhibiting a significantly lower computational complexity and memory footprint. The advantages of the proposed MBSE model over some well-known DNN-based speech enhancement methods are demonstrated through extensive comparative experiments.

The rest of this chapter is organized as follows: the high-level block diagram of the proposed model is provided in Section 4.2. The composite model structure is discussed in Section 4.2.1. The training target of this model is comprised of a spectral mask and a phase derivative that is expressed in Section 4.2.2. Next, the clean speech is reconstructed using the estimated mask and phase derivative as explained in Section 4.2.3. The detailed model architecture and experimental

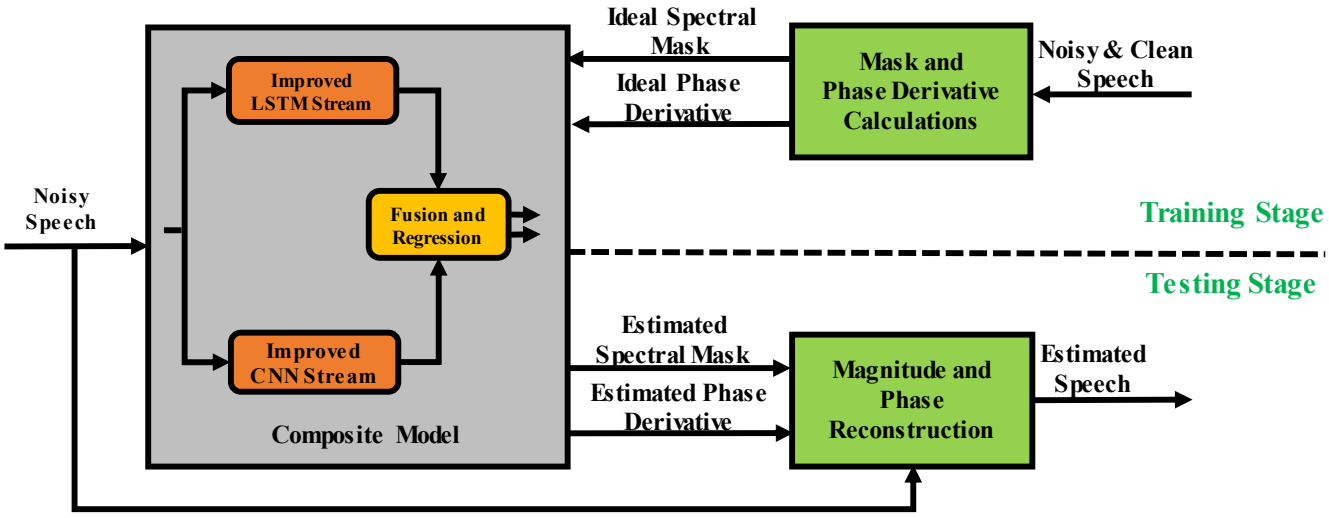


Figure 34: High-level block diagram of the proposed PACDNN model.

results are finally presented in Section 4.2.4 and 4.3, respectively.

4.2 Proposed PACDNN Model

A high-level block diagram of the proposed PACDNN model is shown in Fig. 34. The composite model integrates CNN and LSTM streams to extract a complementary set of features that are then transformed into the network training targets. The composite model input consists of the noisy speech, while its output includes a spectral mask and PD. The mask and PD are calculated and set as the training target model output in the training stage. The clean speech is reconstructed using the estimated mask and PD in the testing stage. The individual components of the PACDNN model are discussed in the following.

4.2.1 Composite Model

Speech exhibits strong dependencies among its samples in both time and frequency domains. The proposed low-complexity composite model integrates carefully designed CNN and LSTM networks to exploit the spectral and temporal information of input speech and extract a complementary set of features. The CNN and LSTM structures are very similar to those in the previous chapter, with some minor modifications. To recap, the CNN is enabled to capture non-local spectral information via dilated frequency convolutions. It also incorporates an attention mechanism to recalibrate its weights without imposing considerable additional complexity. A grouping strategy is adopted for LSTM implementation to reduce its complexity while keeping performance almost unchanged. These modules perform independently and in parallel to speed up the computation, thereby facilitating low-latency and low-complexity real-time applications.

This complementary set of features is then transformed into a spectral mask and PD values. This transformation can be achieved by either a low-complexity CNN or an FC network. These two DNN types have different attributes, although they both can accomplish the required regression task. As stated in [49], CNN can model rapid fluctuations between contiguous elements while FC fails to do so. Besides, CNN requires much fewer model parameters than FC, while the latter requires way fewer computations. We will compare these two networks for the regression task from different perspectives, including speech enhancement performance and computational complexity.

4.2.2 Mask and Phase Derivative Calculation

As mentioned above, the training targets of the composite model in the PACDNN consists of two parts, i.e., spectral mask and PD. The former is applied to the noisy magnitude spectrum to obtain the enhanced one, while the latter is employed to reconstruct the phase spectrum, given the noisy observations. Selecting appropriate training targets is crucial for the final enhancement performance.

The following introduces several popular masks as well as PDs. In our study, the enhancement performance is evaluated by considering different possible combinations of these masks and PDs.

Spectral Mask

Inspired by the masking effects of the human auditory system, masking algorithms aim to retain the speech-dominant regions of the noisy speech in the TF domain while suppressing the noise-dominant ones. To this end, different masks have been introduced in the literature. In this chapter, we investigate four spectral masks as a subtask to the main model, including: IRM, PSM, SMM, and optimal ratio mask (ORM). In Section 3.2.3, IRM and PSM were introduced. SMM and ORM are summarized below.

Spectral Magnitude Mask (SMM) [66], which is conceptually similar to IRM, is defined as the ratio of the spectral magnitude of the clean speech to that of the noisy speech, that is,

$$\mathrm{SMM}(k,l) = \frac{|X(k,l)|}{|Y(k,l)|} \tag{31}$$

where $X(k,l)$ and $Y(k,l)$ denote clean and noisy speech spectrogram, respectively.

Optimal Ratio Mask (ORM) [132] is derived based on the minimization of MSE between the clean and estimated speech. It is given by,

$$\mathrm{ORM}(k,l) = \frac{|X(k,l)|^2 + \Re(X(k,l)N^*(k,l))}{|X(k,l)|^2 + |N(k,l)^2| + 2\Re(X(k,l)N^*(k,l))} \tag{32}$$

where $*$ and $\Re$ denote the conjugate operation and the real part, respectively. The main difference between ORM and IRM is the presence of the term $\Re(X(k,l)N^*(k,l))$ in the former. Accordingly, ORM can be viewed as an improved version of IRM, which takes the correlation between the clean speech and noise into consideration.

Since we use the sigmoid as the output layer's activation function in PACDNN, the values of training targets have to be limited to [0, 1]. Although IRM values fall in the desired range, those of ORM, PSM, and SMM are not limited to this range. Hence, these three masks' outlier values are truncated to [0, 1].

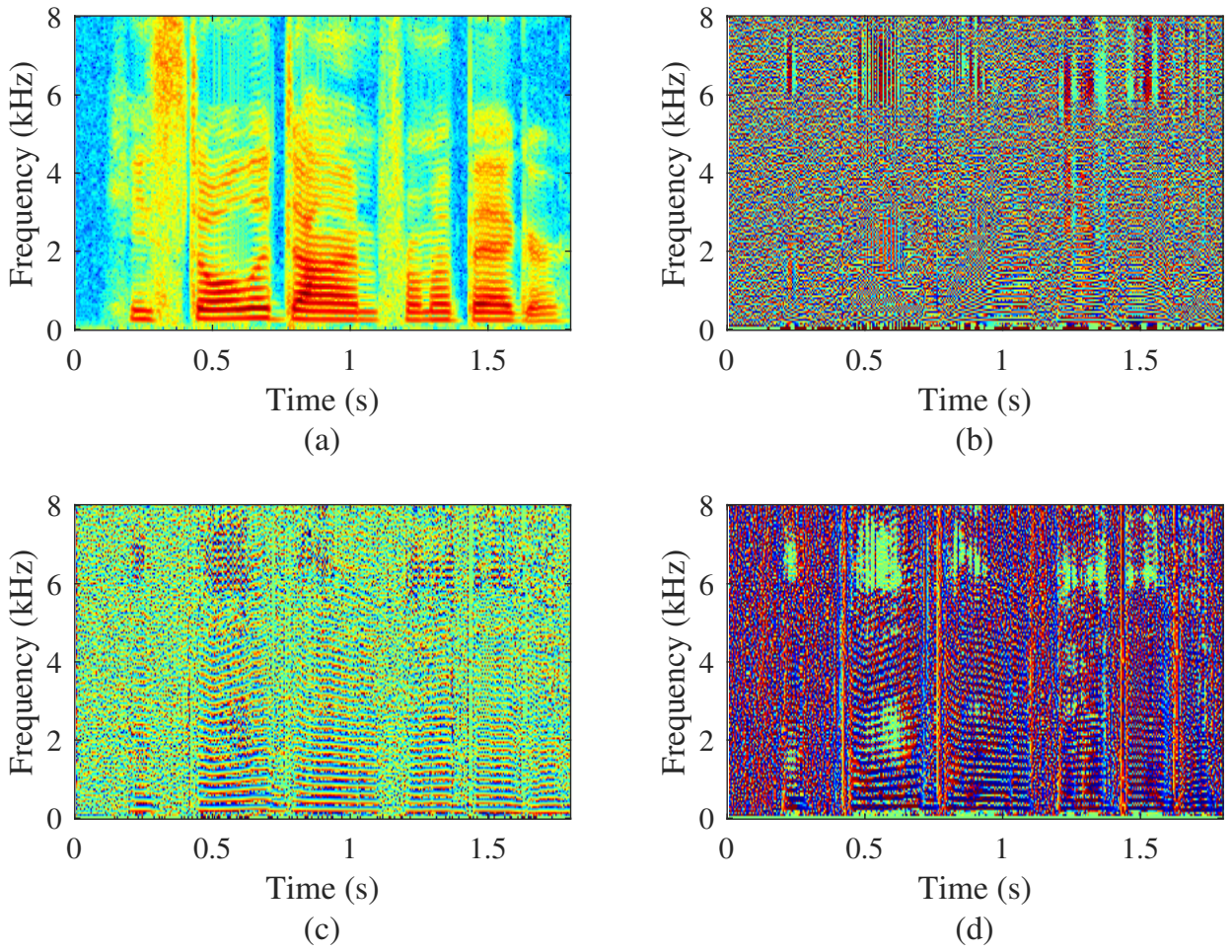


Figure 35: Spectrogram plot of speech at sampling frequency 8 kHz: (a) magnitude; (b) phase; (c) IFD; (d) GD.

Phase Derivative

Processing PD instead of the phase itself has been adopted in some phase-aware speech enhancement methods. In this regard, the instantaneous frequency (IF) [133] and group delay (GD) [134] are two of the most well-known PDs.

Instantaneous frequency (IF) formally defined as the first time-derivative of the phase spectrum. In the case of spectrograms, IF can be approximated by the phase difference between two successive frames as,

$$\mathrm{IF}(k,l) = \mathrm{princ}\left\{\phi(k+1,l) - \phi(k,l)\right\} \tag{33}$$

where the function $\mathrm{princ}\{\cdot\}$ denotes the principal value operator, which projects the phase difference onto $[-\pi, \pi)$. Since IF is limited to its principle value, the wrapping effects would occur along the frequency axis. To alleviate the problem, the instantaneous frequency deviation (IFD) is then adopted,

$$\mathrm{IFD}(k, l) = \mathrm{IF}(k, l) - l \tag{34}$$

It is demonstrated in [133] that the IF values track the frequencies of pitch harmonic peaks, while the IFD values capture pitch and formant structures as in the magnitude spectrogram. Similar findings are presented in [131], in which the authors reconstructed the phase from the estimated IFD for speech enhancement. They also showed that the IFD could be estimated with DNN as it exhibits similar patterns as the magnitude spectrogram, as illustrated in Fig. 35 (a, c).

Group delay (GD) is the negative of the derivative of the spectral phase with respect to frequency, as is given by:

$$\mathrm{GD}(k, l) = -\left[\phi(k, l+1) - \phi(k, l)\right] \tag{35}$$

The authors demonstrated that the GD function behaves like a squared magnitude response at the resonance frequency in [134]. It also exhibits structural patterns similar to the magnitude spectrum, as seen from Fig. 35 (a, d). Moreover, the high-resolution property discussed in [135] reveals that GD has a higher resolving power than the magnitude spectrum. Specifically, the formants are resolved more accurately in the group delay spectrum when compared to the magnitude or linear prediction spectrum. Based on this finding, we infer that GD can also be employed as a training target of the DNN-based speech enhancement, in the same way as the widely-adopted magnitude training targets or their variants.

As the mask and PD are jointly estimated with a single DNN, their values should be in the same range to balance the training process. We adopted the normalization scheme in [131], where the range of the spectral mask is truncated into $[0, 1]$, and the PD values are normalized as follows,

$$\mathrm{PD}_{\mathrm{n}}(k, l) = \frac{1}{2\pi}\mathrm{PD}(k, l) + \frac{1}{2} \tag{36}$$

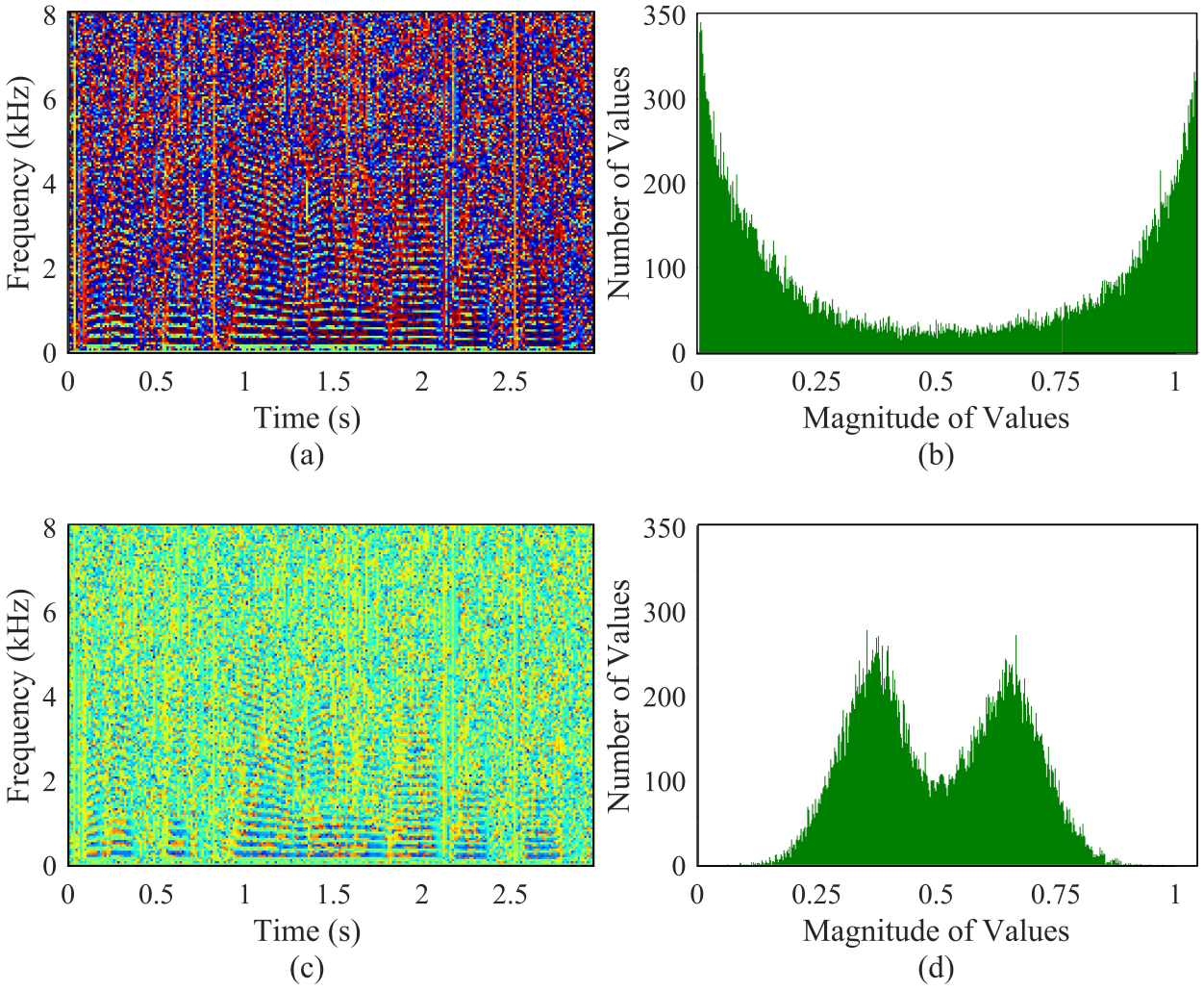


Figure 36: Group delay regularization of 3 seconds clean speech with sampling frequency 16 KHz: (a) GD spectrogram; (b) Distribution of GD values; (c) RGD spectrogram; (d) Distribution of RGD values.

As can be seen from Fig. 36 (a, b), the distribution of the normalized GD values exhibits a U-shaped over the range $[0, 1]$, which renders their accurate estimation with DNN more difficult. As such, we propose to use the following transformation to regularize the normalized GD, namely,

$$\text{RGD}\,(k,l) = \mu + \sqrt{2}\sigma \cdot \text{erfinv}\,(2\text{GD}_n\,(k,l) - 1) \tag{37}$$

where *erfinv*$(\cdot)$ is the inverse error function [1], and σ and μ are set to 0.1 and 0.5, respectively. The RGD and its distribution are shown in Figs. 36 (c, d), where the values are pulled close to the

[1] *erfinv* and *erf* are functions in MATLAB and Scipy (Python library).

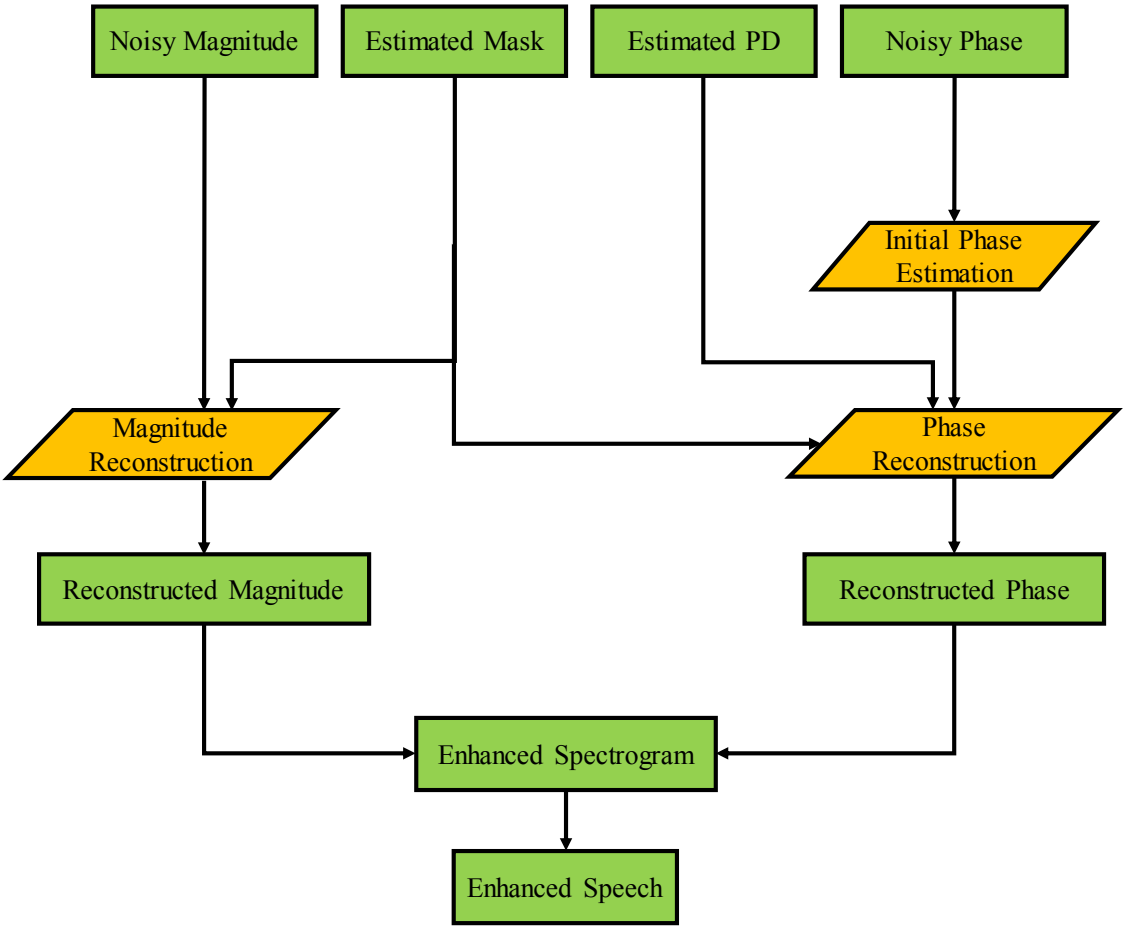


Figure 37: Magnitude and phase reconstruction procedure.

center point (0.5), which makes the RGD a better training target.

4.2.3 Magnitude and Phase Reconstruction

As shown in Fig. 34, there is a magnitude and phase reconstruction block where the magnitude and phase spectra are recovered from the spectral mask and the PD estimates. The whole reconstruction procedure is illustrated in Fig. 37, and will be explained below.

Magnitude Reconstruction

After obtaining the estimated spectral mask $\hat{M}(k,l)$ from the trained DNN, the magnitude reconstruction is accomplished by applying the spectral mask to the magnitude spectrogram of the noisy

speech, i.e.,

$$|\hat{X}(k,l)| = \hat{M}(k,l)\,|Y(k,l))| \tag{38}$$

Typically, if a TF unit is speech dominated, $\hat{M}(k,l)$ will have a large value which helps preserve the speech information in the unit. Otherwise, $\hat{M}(k,l)$ will be small, thereby contributing to suppress the background noise. As mentioned in Section 4.2.2, four types of mask $M(k,l)$, namely IRM, SMM, ORM, and SMM, are investigated in this work.

Phase Reconstruction

Phase reconstruction is performed after obtaining the estimated PDs by the well-trained DNN. Since IF and GD are defined as phase differences between TF units of the spectrogram along the time and frequency axes, respectively, an appropriate initial phase estimate over some chosen TF unit is required to recover the phase spectrogram. Based on the initial estimate, the entire phase spectrogram can be reconstructed along the time and frequency axes through the difference equations in (33) and (35).

1) Initial phase estimation: The noisy phase can be formulated as follows,

$$\phi_y = arg\left(|X|\,e^{j\phi_x} + |N|\,e^{j\phi_n}\right) = \phi_x + arg\left(1 + \frac{|N|}{|X|}e^{j(\phi_n - \phi_x)}\right). \tag{39}$$

In this equation, when the clean speech magnitude, $|X|$, is quite larger than that of the noise, $|N|$, the second term in the argument is close to zero, i.e., the phase of the noisy speech becomes equal to that of the clean one, namely, $\phi_y \approx \phi_x$. This means that the regions wherein the local SNR is high, the phase is less corrupted by noise and thus the noise spectrogram follows the clean one. Hence, using the noisy phase as an initial estimate is justified in TF units with higher SNR. As suggested in [131], we adopt the noisy phase spectrogram as the initial estimate of the clean phase, that is,

$$\hat{\phi}_{init}(k,l) = \phi_Y(k,l), \forall k,l. \tag{40}$$

We then use the local SNR of each TF unit as an index to determine the initial estimate's reliability,

where the local SNR is approximated by the estimated mask $\hat{M}(k,l)$.

2) Phase reconstruction with GD: At first, the estimated RGD, denoted as $\widehat{\mathrm{RGD}}(k,l)$, should be mapped back to $\mathrm{GD}_n(k,l)$ using the following transformation,

$$\widehat{\mathrm{GD}_n}\left(k,l\right)=\frac{1}{2}\left(\mathrm{erf}\left(\frac{\widehat{\mathrm{RGD}}\left(k,l\right)-\mu}{\sqrt{2}\sigma}\right)+1\right) \tag{41}$$

where the *erf*$(\cdot)$ is the error function. Then the estimated GD is obtained by denormalizing $\widehat{\mathrm{GD}_n}$,

$$\widehat{\mathrm{GD}}\left(k,l\right)=2\pi\left(\widehat{\mathrm{GD}_n}\left(k,l\right)-\frac{1}{2}\right) \tag{42}$$

Inspired by the phase reconstruction with IFD in [131], we compute the phase spectrogram using the initial phase estimate and the GD between the initial estimate and the target phase. For each TF unit, we generate $2N_s+1$ frame-conditioned phase estimates, given by,

$$\hat{\phi}^i\left(k,l\right)=\begin{cases}\hat{\phi}_{init}\left(k,l+i\right)+\sum\limits_{n=0}^{i-1}\widehat{\mathbf{GD}}\left(k,l+n\right), & i\neq 0\\ \hat{\phi}_{init}\left(k,l+i\right), & i=0\end{cases}, \tag{43}$$

where $i\in[-N_s,N_s]$ is the frame distance between the initialized TF unit and the target TF unit. These phase estimates are then unwrapped, i.e.,

$$\bar{\phi}^i\left(k,l\right)=\mathrm{unwrap}(\hat{\phi}^i\left(k,l\right)|\hat{\phi}^i\left(k,l-1\right)) \tag{44}$$

The reconstructed phase of the $(k,l)^{\text{th}}$ unit is finally obtained by smoothing the frame-conditioned estimates $\bar{\phi}^i\left(k,l\right)$ with the following weighted average operation,

$$\hat{\phi}\left(k,l\right)=\frac{\sum_{i=-N_s}^{N_s}\left(s(i)\hat{M}\left(k,l+i\right)\right)\bar{\phi}_i\left(k,l\right)}{\sum_{i=-N_s}^{N_s}s(i)\hat{M}\left(k,l+i\right)} \tag{45}$$

where $s(i)$ denotes the proximity weight for $\bar{\phi}^i\left(k,l\right)$, which is inversely related to the absolute

value of the frame distance, that is, a phase estimate $\bar{\phi}^i(k,l)$ with a larger distance $|i|$ is assigned a smaller proximity weight $s(i)$, and to lessen its effect on $\hat{\phi}(k,l)$. In this work, following [131], we chose $s(i)$ as the Hamming window. Moreover, the estimated mask $\hat{M}(k,l)$ is used as an measure of the initial estimate's reliability. For instance, a larger value of $\hat{M}(k,l+i)$ indicates that the local SNR of the i-th TF unit is higher. In this case, the phase estimate $\bar{\phi}^i(k,l)$ is more reliable and contributes more to the final estimate $\hat{\phi}(k,l)$.

3) Phase reconstruction with IFD: The procedure of phase reconstruction with IFD is presented in [131]. To begin with, the estimated $\mathrm{IFD_n}$, denoted as $\widehat{\mathrm{IFD_n}}$*(k,l)*, should be denormalized and converted to $\widehat{\mathrm{IF}}(k,l)$. Then, the phase spectrogram is reconstructed using the noisy phase spectrogram and $\widehat{\mathrm{IF}}$*(k,l)*, with the help of the spectral mask $\hat{M}(k,l)$. Note that phase reconstruction with IFD is similar to the reconstruction with GD. The only difference is that the former is reconstructed along the time axis, while the latter is reconstructed along the frequency axis.

Besides reconstructing the phase with GD only or with IFD only, we also propose the following combination schemes for reconstruction and investigate their corresponding performance in the next section.

- *Two-step reconstruction:* In this scheme, we first take the noisy phase as the initial estimate and use GD/IFD to get the preliminary reconstructed phase. The later is then treated as the initial estimate, which is employed to obtain the final reconstructed phase with IFD/GD.

- *Average reconstruction:* In this scheme, we separately reconstruct the preliminary phase with IFD and GD, respectively. The final reconstructed phase is obtained by averaging the preliminary ones.

With the combination schemes, the final phase estimate $\hat{\phi}_S(k,l)$ is obtained along both time and frequency axes.

Finally, the estimated clean speech spectrogram can be obtained by combining the reconstructed magnitude and phase spectra.

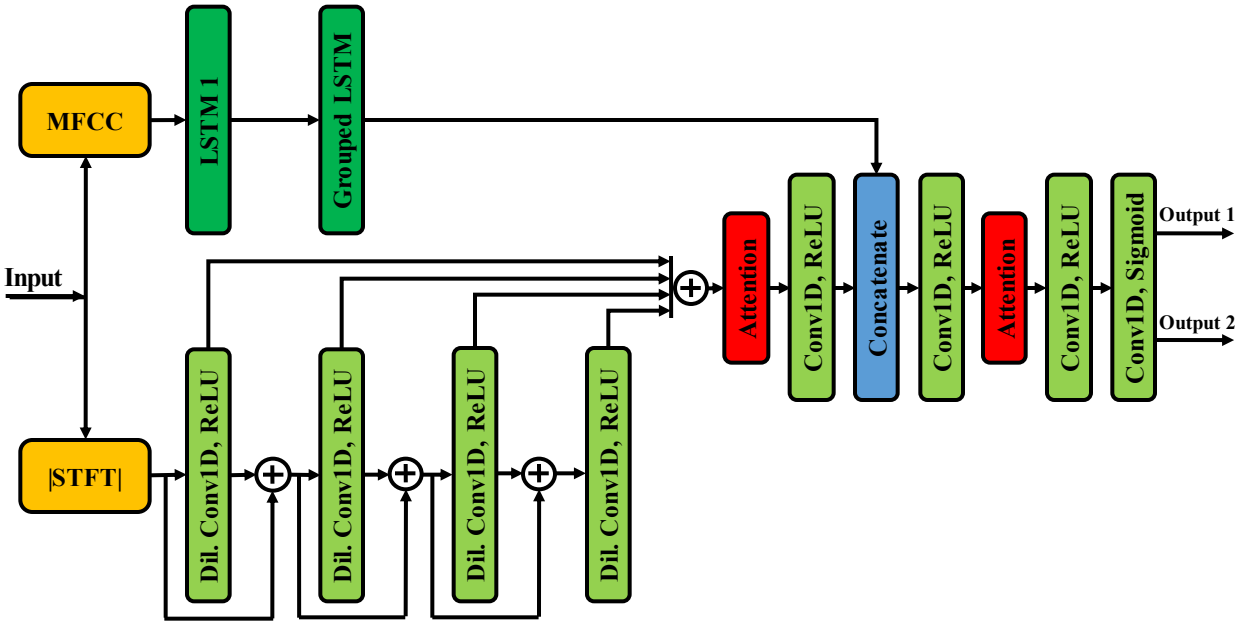


Figure 38: Composite Model Architecture.

4.2.4 Detailed PACDNN Architecture

The composite neural network architecture of our proposed PACDNN model is shown in Fig. 38. The upper stream comprises an LSTM network with two layers, each having 128 LSTM units. We use MFCCs as the LSTM network input, since MFCC is an optimal input for the LSTM network in terms of complexity and performance, as demonstrated in [98]. More specifically, MFCC features are concatenated with their first and second differences, and then normalized to zero mean and unit variance. As mentioned in Section 4.2.1, the grouping strategy is adopted to reduce the LSTM network complexity where the input and hidden layers are divided into K groups. Empirically, we found that grouping only the second layer with K=2 leads to the best speech enhancement results.

In the bottom stream of Fig. 38, the noisy speech STFT magnitude is used as input to the CNN network, which consists of a stack of four dilated-frequency convolutional layers with increasing dilation rates of 1, 2, 4, and 8. The number of kernels in these layers is 16, 32, 16, and 8, with rectified linear unit (ReLU) activation function. Since we want this stream to capture spectral contextual information, the convolutions are 1D with kernel sizes of 1 along with the time and 7 along the frequency dimension. The feed-forward lines around these layers are residual paths, in

the form of convolutional layers with kernel size (1, 1), are used to improve the training procedure. As shown, the outputs of each layer are added up (with a skip connection) to make the output of the CNN network. The output then goes to an attention block, which is spatial with max and average-pooling. The spatial information gathered by max and average pooling are combined using a convolutional layer with sigmoid activation function and kernel size (1, 7), and the final matrix will be element-wise multiplied with the original signal.

The outputs of the LSTM and CNN networks are then concatenated along the channel dimension to form the complementary feature set. Subsequently, another low complexity attention-driven CNN transforms this feature set into the desired targets. This CNN is made up of three convolutional layers with kernel size (1,3) where the number of channels is 32, 16, and 2. The first two layers are followed by ReLU, while the activation function of the output layer is the sigmoid. As explained before, the network estimates a spectral mask and PDs in two channels. Since the structures of these estimators are similar, they are included as two subtasks for the same network through a parameter sharing mechanism. This mechanism provides better generalization and improves learning because it induces a regularization effect between the two subtasks [113]. In the signal reconstruction block, the information from these two channels is used to resynthesize both magnitude and phase as explained in Section 4.2.3. Finally, the clean speech samples in the time domain are generated using inverse STFT and overlap-add operation.

4.3 Experimental Results

4.3.1 Experimental Setup

To evaluate the performance of the proposed PACDNN model, clean utterances are selected from the TIMIT database [81] and IEEE corpus [82], and the noise files from NOISEX-92 [91]. The same setting as described in Section 3.3.1 is used here to prepare the pairs of noisy and clean speech. Four unseen highly-nonstationary noises, namely, *Coffee Shop, Busy City Street, Car Interior,* and *Street Traffic*, are selected from [117] to evaluate the generalization capability of the

proposed model.

The speech STFT is directly fed to the CNN as its input, and used to extract 26 MFCCs as the LSTM input, using a suitable mel-scale filter bank. The MFCCs are finally concatenated with their first and second time differences. Hence, the total length of the feature vector used as input to the LSTM network is 78 (i.e., 26×3).

The MSE is selected as the cost function, while the ADAM optimizer is used as an extension to the stochastic gradient descent [118] to minimize the error between ideal (ground truth) and estimated values of the desired mask and PD, as follows,

$$\text{MSE} = \frac{1}{LK} \sum_{l} \sum_{k} [(M(k,l) - \hat{M}(k,l))^2 + (PD(k,l) - \widehat{PD}(k,l))^2] \tag{46}$$

where L and K respectively denote the number of time frames and frequency bins.

PESQ, STOI, and SSNR are used as the objective evaluation metrics, with the higher these scores are, the better speech enhancement.

4.3.2 Phase-Aware Method Evaluation

The proposed DNN aims to simultaneously estimate the values of both PD and spectral mask. We treat IFD, GD, and their combinations as general PDs. Besides, we investigate four spectral masks, i.e., IRM, ORM, PSM, and SMM.

The comparative performance of the PACDNN model using different combinations of the masks and PDs is shown in Table 12. The experiments are performed using the TIMIT dataset and *restaurant*, *factory*, *street*, and *babble* as noises. The numbers in the table are averaged over all noises and SNR levels. The table is made up of six parts as explained below.

A. This part shows the evaluation metric scores when only a mask is considered as the network's training target with no PD. As seen, PSM yields the best PESQ score while SMM and IRM lead to better STOI and SSNR scores, respectively.

B. This part compares the use of different masks alongside IFD. The results are better than the

Table 12: Comparison of different model targets.

Cases	Objective	PESQ	STOI	SSNR
A	**IRM**	2.61	0.818	4.33
	ORM	2.61	0.814	4.10
	PSM	2.66	0.824	4.28
	SMM	2.54	0.831	4.26
B	**IFD+IRM**	2.67	0.847	5.51
	IFD+ORM	2.66	0.837	4.52
	IFD+PSM	2.71	0.853	6.42
	IFD+SMM	2.64	0.860	5.73
C	**GD+IRM**	2.67	0.848	5.60
	GD+ORM	2.68	0.844	5.64
	GD+PSM	**2.75**	0.853	**6.47**
	GD+SMM	2.66	**0.861**	5.62
D	**(GD-IFD)+IRM**	2.70	0.849	5.69
	(GD-IFD)+ORM	2.70	0.843	5.73
	(GD-IFD)+PSM	2.74	0.854	6.43
	(GD-IFD)+SMM	2.65	0.860	5.58
E	**(IFD-GD)+IRM**	2.70	0.850	5.72
	(IFD-GD)+ORM	2.70	0.844	5.74
	(IFD-GD)+PSM	2.74	0.854	6.42
	(IFD-GD)+SMM	2.65	0.860	5.56
F	**Avg(IFD&GD)+IRM**	2.70	0.849	5.67
	Avg(IFD&GD)+ORM	2.70	0.843	5.72
	Avg(IFD&GD)+PSM	2.74	0.853	6.41
	Avg(IFD&GD)+SMM	2.65	0.860	5.58

previous scenario, showing the advantage of enhancing phase alongside magnitude. In this case, IFD+PSM performs better in terms of PESQ and SSNR, while IFD+SMM yields a slightly better STOI score.

C. This part compares the use of different spectral masks alongside GD. The results are better than both previous scenarios illustrating GD's advantage over IFD. GD+PSM outperforms other combinations in this group in terms of PESQ and SSNR, but not STOI.

D. In this part, a two-stage phase reconstruction is investigated where the noisy phase and GD estimation are used to reconstruct the phase in the first stage, and then the reconstructed phase and IFD estimation are used to obtain the final clean phase estimate in the second stage.

E. This part is similar to the previous one, but in a reverses order, i.e.: the noisy phase and IFD

are first used to reconstruct the phase, and the reconstructed phase is the used with GD estimation to obtain the final phase estimate.

F. This part shows the results when the average of the reconstructed phase using IFD and GD estimation is considered as the clean phase. Although these combinations give good results, the best PESQ and SSNR are obtained using GD+PSM, and the best STOI with GD+SMM.

Hence, we can conclude that the model using PSM+GD as the training target outperforms other scenarios, and thus we adopt it for the rest of our experiments.

4.3.3 Advantages of Grouped LSTM

In the PACDNN model, the LSTM stream exploits the input speech spectrogram's temporal contextual information. LSTM is the most common RNN variation, which is used in this work to avoid the exploding and vanishing gradient problems [55]. Other RNN variations are also considered, such as, GRU and bidirectional forms called BLSTM and BGRU. Furthermore, we adopt the grouping strategy in the LSTM stream to reduce its complexity. This section evaluates the PACDNN model performance using the above-mentioned RNN variations with and without the grouping strategy.

In addition to the metrics mentioned in Section 4.3.1, we compare these variations in terms of: the number of parameters and the required memory to store them; computation time for processing one second of input noisy speech; and the amount of computations measured in terms of the required floating-points operations (FLOPs). These additional measurements are essential for characterizing the implementation complexity of speech enhancement algorithms. These measurements are all made during the testing stage since the trained model parameters are to be saved in the device hardware.

Fig. 39 presents the performance results of the PACDNN model using GRU, LSTM, BGRU, BLSTM, and their grouped versions. In this figure, *M* and *MB* denote million and megabyte, respectively. Note that the dataset is the same as in Section 4.3.2, and the values for PESQ, STOI, and SSNR (dB) show the average improvement over all the noises and SNR levels. As shown in

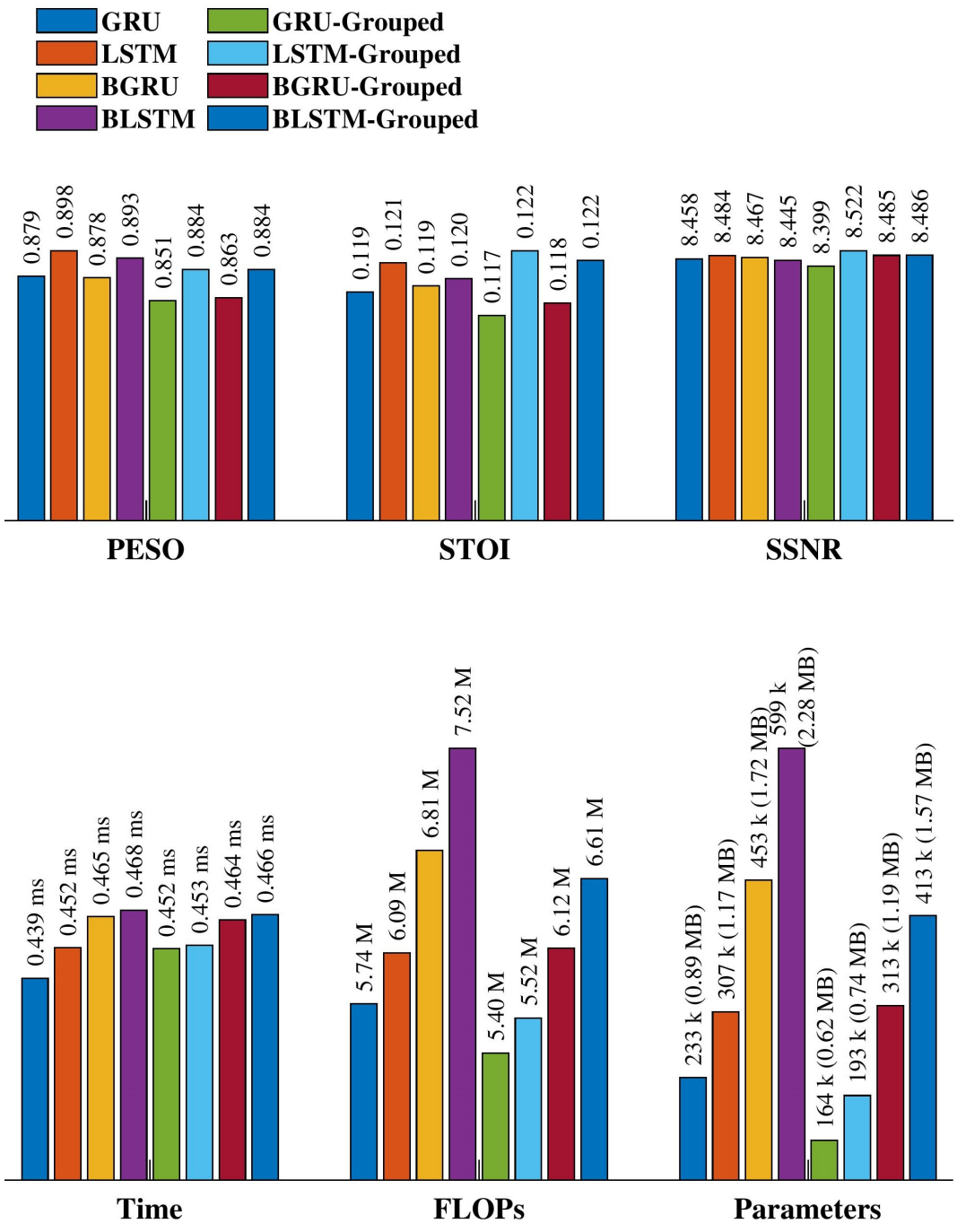


Figure 39: Comparison of PACDNN performance when using different RNN variations.

the figure, using grouped-LSTM yields the best STOI and SSNR scores, while LSTM outperforms others in terms of PESQ score. While the objective results do not show a considerable difference, the results for the complexity measures, especially FLOPs and number of parameters, display

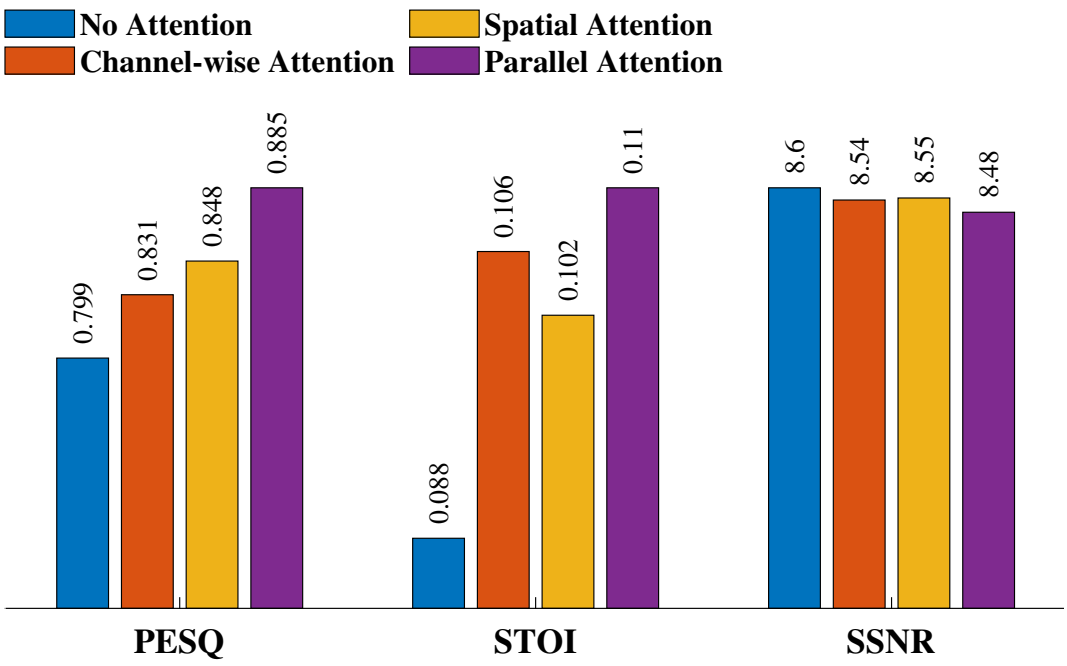


Figure 40: Comparison of PACDNN performance when embedding different attention methods.

huge variations. With respect to processing time, GRU is clearly the fastest while BLSTM is the slowest approach. The grouped variations lead to models with smaller number of parameters and FLOPs, and among them, grouped-GRU requires the least number of parameters and FLOPs, while grouped-LSTM ranks second. Considering both objective speech quality and computational complexity metrics, the grouped-LSTM offers the best trade-off among the RNN variations in the PACDNN model.

4.3.4 Benefits of Attention-Driven CNN

CNN generates many feature maps, each containing some spectrogram characteristics. These feature maps mostly convey noise or speech information. In the PACDNN model, the attention technique is embedded in CNNs to recalibrate feature map weights and emphasize the speech-bearing ones. As mentioned in Section 4.2.1, we consider three attention techniques, i.e., channel-wise, spatial, and parallel, to be embedded in the PACDNN model, and compare the overall model performance. The results of the different cases, using the same dataset as in Section 4.3.2, are illustrated in Fig. 40, where the values show the average improvement over all the noises and SNR levels.

Considering the PESQ score, the PACDNN model with no attention gives the lowest score, while embedding the parallel attention technique yields the highest score. Regarding STOI, the model with the parallel attention again outperforms others, while that with no attention leads to the lowest score. These results demonstrate the effectiveness of attention techniques in emphasizing the informative feature maps. The parallel attention technique made up of average and max pooling can also capture important information of the input feature maps from different perspectives and further improve their representation power. Regarding SSNR, the use of attention model tends to reduce, although marginally, the attainable values.

4.3.5 Investigation of the Regression Model

Two parallel streams in the PACDNN model exploit a complementary set of features, which is to be transformed into the spectral mask and phase derivative(s). This transformer could be both a CNN as mentioned in Section 4.2.1 or an FC network. These two DNNs have different attributes though they both accomplish the required regression task. As stated in [49], CNN can model rapid fluctuations between contiguous elements while DNN fails to do such. Besides, CNN requires much less model parameters than FC network, while the latter requires way less memory for its computations.

This section evaluates the CNN against an FC network for the final regression part of the PACDNN model. The FC network contains three layers, each having 512 nodes with a ReLU activation function. A dropout at the rate of 0.3 is also applied to avoid over-fitting. The output layer consists of 322 nodes with sigmoid activation functions to build the desired mask and PD.

The comparative performance of the two networks in terms of objective speech quality and computational complexity metrics is presented in Fig. 41. As shown, FC network yields slightly better results in terms of objective measurements. This marginal advantage of FC network stems from its number of parameters. Using FC network in PACDNN requires around five times more trainable parameters than using CNN, which means the model with FC network can learn more specific patterns of the training dataset. It is worth mentioning that a low complexity model is

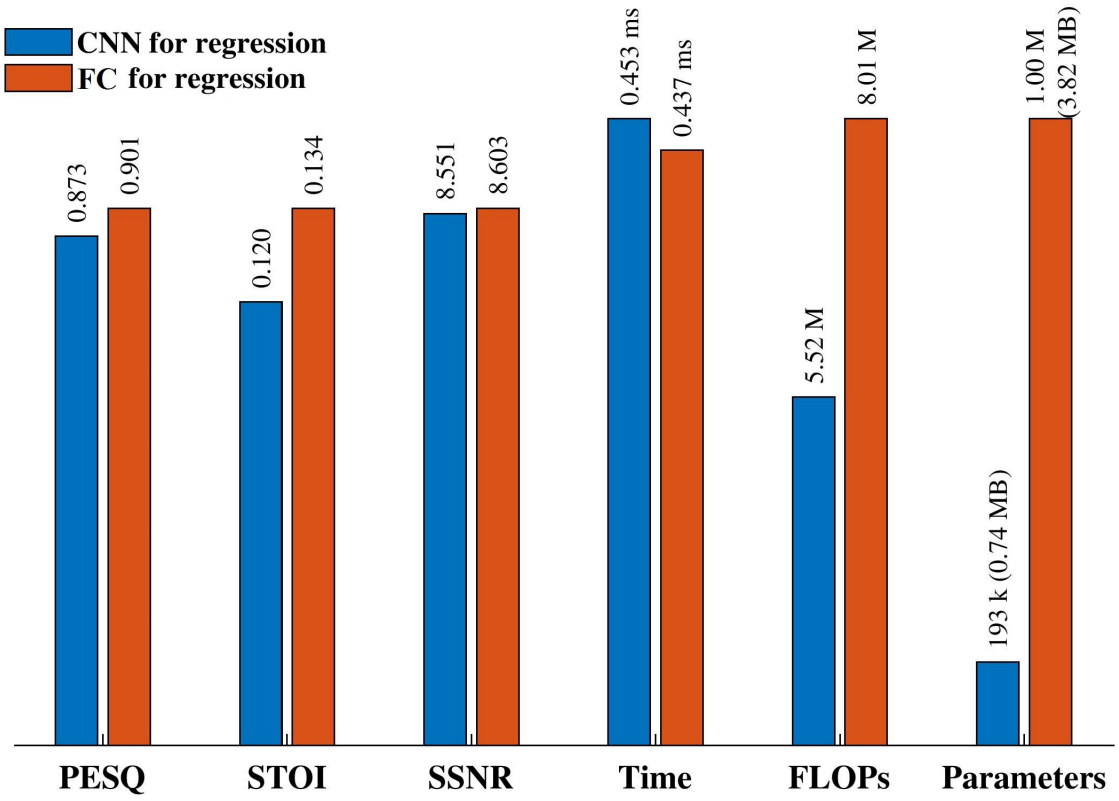


Figure 41: Comparison of PACDNN model performance while using CNN or FC for the final regression.

preferable from the implementation and generalization perspectives. While a model with a low number of parameters does not have the capacity to learn specific patterns or detailed information about noise and speech utterances in the training dataset, and it can perform very well under unseen acoustic conditions. Apart from that, using CNN and FC network in the model respectively requires 0.74 MB and 3.82 MB of memory to store the fixed model parameters, which is proportional to the number of parameters. While the basic computations in FC network are conceptually simpler than in CNN, the former still requires 1.46 times more FLOPs than the former, which is due to the larger number of model parameters. At last, the computation time of CNN, which performs a large number of matrix multiplications, slightly exceeds that of FC network.

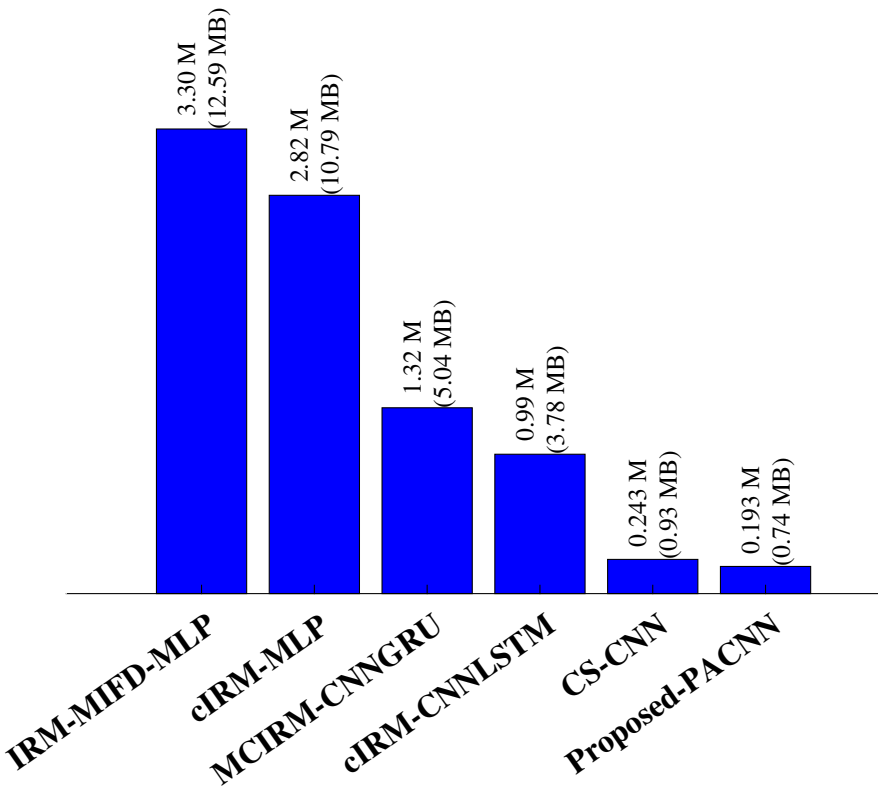


Figure 42: Comparison of the number of trainable parameters of different methods.

4.3.6 Comparison with other DNN-Based Methods

This section compares the proposed PACDNN model with some well-known DNN models in the speech enhancement task. The selected models have moderate complexity. All the selected methods consider phase information for speech enhancement along with magnitude enhancement. All the models, including PACDNN, are trained and tested with the same dataset under the same condition to ensure a fair comparison. The selected methods are summarized below:

1. IRM-MIFD-FC [131]: A FC network with three layers is employed in this multi-objective DNN method. Each hidden layer contains 1024 nodes with ReLU activation function while the output layer contains 512 nodes with sigmoid activation function. IRM and IFD are used as training targets.

2. cIRM-FC [6]: In this method, three-layer FC network is employed to approximate cIRM. Each layer has 1024 nodes with ReLU activation function. The output layer with linear

activation function estimates the real and and imaginary parts of cIRM. The input to the network is a complementary set of acoustic features. To incorporate temporal information, the features from 5 frames are concatenated and fed to the network at once.

3. MCIRM-CNNGRU [116]: In this method, a hybrid model is used to estimate the real and imaginary parts of a modified cIRM. The network is made up of a CNN for feature extraction and a GRU network for regression. The complex spectrogram is used as the input and a 322-node output layer with linear activation function generates the desired mask values.

4. cIRM-CNNLSTM [98]: Here, CNN, LSTM, and FC networks are integrated to estimate cIRM. The feature extraction is performed by the CNN and LSTM networks while the regression is accomplished by the FC network, which maps the features into the real and imaginary components of cIRM.

5. CS-CNN [12]: A fully-convolutional CNN is utilized to estimate the real and imaginary parts of the clean speech complex spectrogram. The input consists of 13 frames of the noisy speech complex spectrogram presented to the network. The middle frame of the output (frame 7) is considered as the enhanced output frame.

Fig. 42 illustrates the number of trainable parameters of each method. As expected, the FC network based models, i.e., IRM-MIFD-FC and cIRM-FC, contain the highest number of parameters and, consequently, require a large memory to store them. Two other hybrid models, i.e., MCIRM-CNNGRU and cIRM-CNNLSTM, have a fair number of parameters, each around 1 million. The lowest number of parameters belongs to CS-CNN and the proposed model, with the latter requiring slightly less parameters. Although the number of model parameter of PACDNN is 6% that of IRM-MIFD-FC, it outperforms all these aforementioned models in the speech enhancement task, as further discussed below.

The comparison results for male test utterances from the TIMIT dataset are shown in Table 13 where bble, ftry, rtrt, and strt denote babble, factory, restaurant, and street noises, respectively. As shown, the proposed model outperforms all the other ones in terms of the various objective

Table 13: Comparison of different methods with unseen **male** utterances from TIMIT dataset.

SNR	Method	PESQ				STOI				SSNR			
		bble	**ftry**	**rtrt**	**strt**	**bble**	**ftry**	**rtrt**	**strt**	**bble**	**ftry**	**rtrt**	**strt**
-6 dB	**Unprocessed**	1.23	1.08	1.29	1.25	0.522	0.509	0.516	0.609	-10.6	-10.3	-9.97	-9.64
	IRM-MIFD-FC	1.57	1.57	1.68	1.95	0.588	**0.591**	0.647	0.724	-3.83	-2.52	-1.78	-0.30
	cIRM-FC	1.57	1.74	1.68	2.06	0.562	0.568	0.624	0.712	**-1.02**	**0.29**	0.14	1.65
	MCIRM-CNNGRU	1.57	1.71	1.53	1.90	0.523	0.544	0.545	0.657	-2.42	-0.80	-2.01	0.45
	cIRM-CNNLSTM	1.67	1.80	1.84	2.04	0.578	0.589	**0.667**	0.716	-1.29	-0.80	-0.26	1.53
	CS-CNN	1.53	1.41	1.48	1.72	0.515	0.518	0.506	0.612	-4.25	-3.86	-5.22	-1.95
	Proposed	**1.72**	**1.81**	**1.87**	**2.19**	**0.591**	**0.591**	0.660	**0.740**	-1.10	-0.46	**0.17**	**1.66**
0 dB	**Unprocessed**	1.66	1.52	1.65	1.74	0.659	0.647	0.651	0.718	-5.97	-5.72	-5.45	-4.84
	IRM-MIFD-FC	2.12	2.15	2.24	2.49	0.732	0.738	0.762	0.803	0.97	1.96	2.22	3.98
	cIRM-FC	2.17	2.25	2.24	2.53	0.724	0.716	0.744	0.796	2.21	2.84	2.76	4.18
	MCIRM-CNNGRU	2.10	2.17	2.03	2.42	0.696	0.700	0.698	0.767	1.36	1.65	1.44	3.22
	cIRM-CNNLSTM	2.24	2.31	2.32	2.53	0.736	0.743	0.773	0.813	1.84	2.12	2.52	3.91
	CS-CNN	2.01	1.87	1.88	2.12	0.667	0.662	0.652	0.722	0.17	0.28	-1.23	1.35
	Proposed	**2.28**	**2.34**	**2.34**	**2.67**	**0.751**	**0.750**	**0.778**	**0.824**	**2.48**	**3.04**	**3.23**	**4.60**
6 dB	**Unprocessed**	2.12	1.98	2.08	2.23	0.778	0.780	0.789	0.806	-0.17	-0.21	0.21	0.29
	IRM-MIFD-FC	2.71	2.70	2.73	2.97	0.837	0.840	0.852	0.862	5.46	5.83	5.73	6.79
	cIRM-FC	2.74	2.73	2.74	2.94	0.828	0.824	0.845	0.854	5.49	5.35	5.59	6.49
	MCIRM-CNNGRU	2.60	2.63	2.57	2.82	0.811	0.810	0.819	0.836	4.13	4.13	4.33	5.40
	cIRM-CNNLSTM	2.79	2.86	2.84	**3.13**	0.850	0.850	0.859	0.872	5.33	5.38	6.32	5.44
	CS-CNN	2.41	2.24	2.33	2.51	0.780	0.770	0.776	0.800	3.48	3.48	2.87	4.06
	Proposed	**2.83**	**2.89**	**2.85**	3.08	**0.860**	**0.855**	**0.875**	**0.882**	**6.26**	**6.66**	**6.51**	**7.72**
12 dB	**Unprocessed**	2.53	2.46	2.51	2.66	0.877	0.890	0.896	0.886	5.51	5.46	5.88	6.05
	IRM-MIFD-FC	3.22	3.16	3.20	3.37	0.906	0.907	0.918	0.913	8.42	8.56	8.28	9.09
	cIRM-FC	3.16	3.17	3.19	3.33	0.897	0.896	0.909	0.903	7.86	7.80	7.79	8.49
	MCIRM-CNNGRU	3.02	3.08	3.04	3.25	0.888	0.882	0.896	0.891	6.73	6.31	6.72	7.51
	cIRM-CNNLSTM	3.27	3.25	3.26	**3.49**	0.910	0.910	0.920	0.916	8.13	8.19	8.21	8.78
	CS-CNN	2.77	2.63	2.74	2.86	0.854	0.848	0.856	0.860	6.01	5.96	5.83	6.42
	Proposed	**3.30**	**3.26**	**3.27**	3.47	**0.920**	**0.921**	**0.930**	**0.927**	**10.08**	**10.33**	**10.18**	**11.11**

quality metrics, except for a few cases, including PESQ at SNR levels of 6 and 12 dB for street noise where MCIRM-CNNGRU give slightly better scores. Also, at the SNR level of -6, cIRM-FC yields marginally better SSNR for babble and factory noises. Table 14 illustrates results for the female utterances from the TIMIT dataset. Again, we can see that the proposed model outperforms the others in nearly all cases, except for STOI at SNR level of -6 for factory and restaurant noises where IRM-MIFD-FC gives better results.

Table 14: Comparison of different methods with unseen **female** utterances from TIMIT dataset.

SNR	Method	PESQ				STOI				SSNR			
		bble	**ftry**	**rtrt**	**strt**	**bble**	**ftry**	**rtrt**	**strt**	**bble**	**ftry**	**rtrt**	**strt**
	Unprocessed	0.95	0.87	0.93	0.93	0.512	0.504	0.497	0.588	-9.97	-10.0	-9.51	-9.51
	IRM-MIFD-FC	1.22	1.23	1.28	1.54	0.568	**0.579**	**0.631**	0.693	-3.51	-2.26	-1.19	-0.13
	cIRM-FC	1.27	1.43	1.31	1.67	0.545	0.549	0.601	0.679	-0.92	**0.40**	0.32	1.97
-6 dB	**MCIRM-CNNGRU**	1.28	1.37	1.23	1.60	0.506	0.526	0.544	0.635	-2.15	-0.95	-1.90	0.56
	cIRM-CNNLSTM	1.38	1.45	1.44	1.79	0.551	0.564	0.635	0.690	-0.93	-0.58	0.48	1.90
	CS-CNN	1.27	1.23	1.23	1.45	0.505	0.517	0.497	0.588	-3.85	-3.04	-5.03	-1.58
	Proposed	**1.40**	**1.48**	**1.46**	**1.80**	**0.570**	0.561	0.629	**0.703**	**-0.70**	-0.12	**0.94**	**2.35**
	Unprocessed	1.36	1.28	1.36	1.48	0.640	0.635	0.641	0.694	-5.58	-5.64	-5.00	-4.89
	IRM-MIFD-FC	1.77	1.81	1.82	2.09	0.709	**0.718**	0.743	0.784	1.34	2.13	2.50	3.97
	cIRM-FC	1.81	1.94	1.88	2.15	0.687	0.690	0.729	0.765	2.46	2.90	3.09	4.40
0 dB	**MCIRM-CNNGRU**	1.83	1.92	1.75	2.10	0.674	0.676	0.699	0.754	1.44	1.83	1.63	3.61
	cIRM-CNNLSTM	1.92	1.98	1.93	**2.41**	0.710	0.716	0.749	0.787	2.12	2.32	3.15	4.33
	CS-CNN	1.76	1.65	1.67	1.91	0.653	0.649	0.643	0.702	0.46	0.59	-0.73	1.65
	Proposed	**1.94**	**2.06**	**1.96**	2.30	**0.721**	0.711	**0.750**	**0.798**	**3.00**	**3.50**	**3.80**	**5.35**
	Unprocessed	1.85	1.77	1.84	2.01	0.766	0.763	0.777	0.798	-0.18	-0.30	0.16	0.49
	IRM-MIFD-FC	2.34	2.37	2.35	2.63	0.822	0.824	0.838	0.849	5.61	6.00	6.06	7.31
	cIRM-FC	2.34	2.42	2.36	2.63	0.805	0.802	0.823	0.836	5.31	5.41	5.68	6.94
6 dB	**MCIRM-CNNGRU**	2.31	2.37	2.25	2.54	0.795	0.788	0.810	0.824	4.34	4.23	4.59	5.75
	cIRM-CNNLSTM	2.47	2.52	**2.52**	**2.85**	0.824	0.830	0.845	0.857	5.67	5.59	5.76	7.08
	CS-CNN	2.17	2.11	2.09	2.31	0.768	0.763	0.770	0.790	3.70	3.84	3.13	4.41
	Proposed	**2.50**	**2.58**	2.47	2.78	**0.836**	**0.837**	**0.849**	**0.869**	**6.82**	**7.09**	**7.06**	**8.75**
	Unprocessed	2.34	2.30	2.31	2.50	0.869	0.883	0.888	0.882	5.53	5.50	5.91	6.11
	IRM-MIFD-FC	2.90	2.92	2.85	3.09	0.896	0.900	0.905	0.899	8.61	8.67	8.42	9.24
	cIRM-FC	2.86	2.96	2.84	3.05	0.882	0.887	0.892	0.891	7.67	7.65	7.71	8.62
12 dB	**MCIRM-CNNGRU**	2.82	2.85	2.76	2.99	0.875	0.873	0.887	0.881	7.23	6.69	7.15	7.74
	cIRM-CNNLSTM	3.00	2.99	2.95	**3.25**	0.898	0.903	0.909	0.907	8.25	8.4	8.52	9.46
	CS-CNN	2.56	2.47	2.52	2.70	0.843	0.840	0.847	0.853	6.16	6.07	5.95	6.52
	Proposed	**3.02**	**3.07**	**2.97**	3.23	**0.915**	**0.916**	**0.922**	**0.920**	**10.89**	**11.04**	**10.89**	**12.16**

In another experiment, we compare the different methods on the IEEE corpus where 20 noises are mixed with the selected utterances, with unmatched SNR levels between the training and testing stages. As can be seen from Table 15, which presents the average scores for the PESQ, STOI and SSNR metrics, the proposed model clearly outperforms all the other methods in all cases, except for the SSNR scores at SNR levels of -6 and 6 dB, where CS-CNN gives slightly better results. This experiment demonstrates that although the proposed model has a very small number of parameters,

Table 15: Comparison of different methods with unseen utterances from IEEE corpus and 20 different noises.

Method	PESQ				STOI				SSNR			
	-6	**0**	**6**	**12**	**-6**	**0**	**6**	**12**	**-6**	**0**	**6**	**12**
Unprocessed	1.40	1.76	2.13	2.54	0.588	0.708	0.825	0.913	-8.99	-5.17	0.01	5.75
IRM-MIFD-FC	1.83	2.36	2.88	3.30	0.711	0.824	0.898	0.942	-1.77	3.19	7.30	10.16
cIRM-FC	1.85	2.37	2.86	3.27	0.690	0.810	0.889	0.938	1.03	4.35	7.26	10.02
MCIRM-CNNGRU	1.85	2.34	2.78	3.15	0.658	0.782	0.869	0.922	0.22	3.45	6.14	8.26
cIRM-CNNLSTM	2.06	2.58	3.06	3.44	0.720	0.832	0.907	0.949	1.02	4.54	8.07	10.88
CS-CNN	1.98	2.46	2.82	3.09	0.685	0.817	0.896	0.939	**2.11**	4.98	**9.23**	11.01
Proposed	**2.07**	**2.60**	**3.08**	**3.46**	**0.724**	**0.838**	**0.911**	**0.955**	1.63	**5.08**	8.34	**11.76**

it can perform well under different noise conditions.

Under the same training conditions as for Table 15, we tested the different methods with unseen highly-nonstationary noises mixed with unseen utterances from IEEE corpus at unmatched SNR levels to evaluate their generalization capability in unseen conditions. The comparison results are shown in Table 16 where bscs, cair, cfsp, and sttc denote *Coffee Shop, Busy City Street, Car Interior,* and *Street Traffic*. It can be seen that the proposed model generally outperforms all the other methods, except for a few cases. This experiment demonstrates that the proposed model has very good generalization capability thanks to its careful design and the small number of parameters making it not learn specific patterns of the training dataset but instead rely on the general information of speech and noise.

As shown in [127], there can be a considerable performance degradation with DNN methods when the training and testing datasets are different, especially at low SNR levels. This study reveals that some well-known but highly complex speech enhancement methods do not perform well on untrained corpora. In this last experiment, we compare the cross-corpus generalization capability of different methods. To this end, we trained different models with the TIMIT dataset and tested them with the IEEE corpus. The results, shown in Table 17 for different SNR levels, reveal that the proposed model outperforms the other ones when the training and testing datasets are different, except at SNR -6 dB, where other methods yield somehow better results. Hence, we can conclude

Table 16: Comparison of different methods with unseen utterances from IEEE corpus mixed with **unseen** noises at unmatched SNR levels.

SNR	Method	PESQ				STOI				SSNR			
		bscs	**cair**	**cfsp**	**sttc**	**bscs**	**cair**	**cfsp**	**sttc**	**bscs**	**cair**	**cfsp**	**sttc**
-6 dB	**Unprocessed**	1.30	1.12	1.71	1.18	0.587	0.506	0.715	0.497	-7.06	-8.66	-9.16	-9.26
	IRM-MIFD-FC	1.76	1.37	2.20	1.42	0.693	0.579	0.817	**0.584**	-2.01	-3.35	-1.28	-3.50
	cIRM-FC	1.73	1.25	2.23	1.34	0.678	0.557	0.814	0.544	0.58	-0.97	2.14	**-1.00**
	MCIRM-CNNGRU	1.69	1.42	2.09	1.37	0.632	0.539	0.756	0.518	-0.71	-2.21	1.03	-2.85
	cIRM-CNNLSTM	1.74	1.30	2.37	1.30	0.677	0.562	0.815	0.542	0.23	-1.18	2.73	-1.77
	CS-CNN	1.64	1.46	2.14	1.53	0.657	0.554	0.786	0.535	**0.99**	-0.95	3.12	-2.08
	Proposed	**1.96**	**1.51**	**2.59**	**1.55**	**0.701**	**0.601**	**0.845**	0.578	0.90	**-0.65**	**3.86**	-1.08
0 dB	**Unprocessed**	1.81	1.60	2.17	1.52	0.727	0.632	0.807	0.629	-4.33	-4.97	-4.96	-5.38
	IRM-MIFD-FC	2.27	1.96	2.70	1.93	0.815	0.742	0.876	0.728	2.68	1.35	4.05	1.36
	cIRM-FC	2.28	1.91	2.80	1.91	0.803	0.736	0.873	0.722	3.85	2.77	5.23	2.81
	MCIRM-CNNGRU	2.18	1.89	2.58	1.84	0.777	0.686	0.852	0.682	2.55	1.36	3.90	1.36
	cIRM-CNNLSTM	2.34	1.99	2.84	1.97	0.815	0.733	0.882	0.728	4.04	2.38	5.68	2.76
	CS-CNN	2.29	1.96	2.63	2.05	0.827	0.731	0.882	0.725	**5.66**	3.35	6.09	2.65
	Proposed	**2.49**	**2.10**	**3.02**	**2.14**	**0.837**	**0.758**	**0.897**	**0.753**	4.62	**3.41**	**6.33**	**3.05**
6 dB	**Unprocessed**	2.15	2.00	2.58	1.87	0.837	0.774	0.878	0.761	0.63	0.17	0.29	-0.18
	IRM-MIFD-FC	2.76	2.54	3.17	2.51	0.897	0.853	0.916	0.849	6.85	5.99	7.75	6.16
	cIRM-FC	2.76	2.52	3.24	2.50	0.885	0.853	0.915	0.848	6.89	6.08	7.75	6.04
	MCIRM-CNNGRU	2.64	2.42	3.06	2.37	0.872	0.822	0.904	0.819	5.74	4.71	6.68	4.84
	cIRM-CNNLSTM	2.86	2.61	3.30	2.55	0.903	0.858	0.928	0.854	7.13	6.16	8.90	6.68
	CS-CNN	2.66	2.48	3.05	2.49	0.902	0.860	**0.937**	0.848	**9.17**	6.91	**10.1**	6.54
	Proposed	**2.92**	**2.67**	**3.41**	**2.67**	**0.906**	**0.873**	0.935	**0.866**	7.86	**7.12**	8.92	**6.97**
12 dB	**Unprocessed**	2.54	2.44	2.98	2.30	0.927	0.889	0.933	0.883	6.56	6.10	6.15	5.49
	IRM-MIFD-FC	3.29	3.04	3.59	2.99	0.947	0.922	0.950	0.922	10.07	9.37	10.34	9.62
	cIRM-FC	3.21	3.02	3.63	3.01	0.940	0.920	0.949	0.922	10.00	9.27	10.21	9.25
	MCIRM-CNNGRU	3.05	2.89	3.48	2.87	0.931	0.905	0.939	0.902	8.24	7.80	9.09	7.34
	cIRM-CNNLSTM	3.26	3.10	3.63	3.07	0.950	0.932	0.954	0.923	10.22	9.12	10.15	9.91
	CS-CNN	2.98	2.87	3.38	2.87	0.948	0.930	**0.964**	0.924	**12.04**	10.17	**12.58**	10.64
	Proposed	**3.36**	**3.19**	**3.70**	**3.17**	**0.955**	**0.936**	0.962	**0.936**	11.45	**10.79**	11.80	**10.93**

that the proposed PACDNN model offers very good generalization capability to unseen datasets.

4.4 Conclusion

This chapter proposed a phase-aware composite deep neural network called PACDNN for speech enhancement where both speech magnitude and phase are enhanced. Specifically, we designed a

Table 17: Cross-corpus evaluation, where the training and testing are accomplished with TIMIT dataset and IEEE corpus, respectively.

Method	PESQ				STOI				SSNR			
	-6	**0**	**6**	**12**	**-6**	**0**	**6**	**12**	**-6**	**0**	**6**	**12**
Unprocessed	1.29	1.70	2.10	2.52	0.541	0.676	0.814	0.913	-8.27	-4.64	0.61	6.34
IRM-MIFD-FC	1.57	2.05	2.53	3.01	**0.609**	0.741	0.837	0.922	-2.44	1.63	4.76	6.51
cIRM-FC	1.53	2.03	2.51	2.97	0.591	0.731	0.840	0.907	**-0.17**	2.21	4.22	5.77
MCIRM-CNNGRU	1.55	2.00	2.44	2.87	0.531	0.703	0.822	0.891	-1.69	1.50	3.45	4.57
cIRM-CNNLSTM	**1.66**	2.09	2.58	3.01	0.598	0.740	0.843	0.907	-0.72	2.57	4.35	5.96
CS-CNN	1.48	1.84	2.24	2.57	0.513	0.658	0.764	0.827	-3.99	-0.32	2.84	4.88
Proposed	1.64	**2.12**	**2.60**	**3.05**	0.604	**0.752**	**0.858**	**0.926**	-0.30	**2.62**	**5.37**	**7.96**

masking-based method to enhance the magnitude and employed phase derivative to reconstruct the clean speech phase. Due to the structural similarity of the spectral mask and phase derivative, a single neural network was used to estimate both training targets through simultaneous parameter sharing. The proposed network integrates improved LSTM and CNN, which perform in parallel to exploit a complementary set of features. Different potential DNN solutions were investigated and compared in terms of objective speech quality and computational complexity measures in order to optimize the final regression between the features and the desired targets. Through extensive series of experiments, the resulting PACDNN model was evaluated and compared with several known DNN-based speech enhancement methods using different datasets and objective measures. In particular, the capability of the proposed model in dealing with unseen noisy conditions, cross-corpus generalization, and unmatched SNR levels in testing and training were investigated, demonstrating the advantages of PACDNN over other speech enhancement methods, in spite of its lower complexity.

Chapter 5

Multi-Mode Mapping-Based Speech Enhancement in Discrete Cosine Transform Domain

5.1 Introduction

In the previous chapters, the importance of processing speech phase along with the magnitude to speech enhancement was discussed. Also, an overview of the available approaches for simultaneous magnitude and phase enhancement along with their pros and cons was presented, including masking-based methods like PSM [68] and mapping-based complex spectrogram ones like CRN [114]. All these methods use DFT to transform time-domain input speech to the frequency domain that is more discriminative and appropriate for DNN-based methods than the time domain. Since speech spectrogram contains complex values, its real and imaginary components have to be processed simultaneously, leading to additional computational burden. As such, STFT variations like log-Mel, magnitude, or energy spectrum are usually processed instead of complex STFT itself, stressing only magnitude enhancement, and thus, limiting the speech enhancement performance.

Discrete cosine transform (DCT) is another transformation that can be used in speech enhancement. Unlike DFT, DCT is a real-valued transform that includes phase and magnitude information of the signal; thus, no information is lost using this transformation. Very recently, Li *et al.* [136] presented a DNN-based monaural speech enhancement methods in the DCT domain. They adopted a CRN network as the learning machine that is fed by short-time DCT (STDCT) of input speech. As the output, a ratio mask in the DCT domain is adopted. Since the length of input and output of the DCT transformation is equal, a modified DCT (MDCT) was introduced in [137] and adopted for speech enhancement in [138]. MDCT is a lapped transform where the output size is half of the input. As such, the size of the training target in the MDCT domain is half compared to that in the DCT domain. Thus, estimation of the training target with a small length in the MDCT domain is easier for DNN, and the amount of computations is also reduced.

Another important factor that has to be taken into account while designing a speech enhancement system is the generalization capability of the DNN to different noise types, SNR levels, and speakers, as discussed in Section 1.3.3. To improve the generalization capability of DNN, a two-stage speech enhancement system called modular neural network was proposed in [139]. In the first stage, various well-trained speech enhancement modules specialized in dealing with a specific SNR, noise, or speaker perform in parallel. In the second stage, an arbitrator, which is another DNN, selects the best module from the previous stage. The authors showed that this system performs better than an arbitrarily chosen DNN. In [140], a deep recurrent mixture-of-experts algorithm for speech enhancement was proposed. In order to reduce the large speech variability, the network is split into several networks, each specialized in a specific and smaller task considering speech phonemes. The authors therein demonstrated the superiority of this architecture over some common DNNs. In [59], the authors investigated speech enhancement performance using DNNs trained to be noise, SNR level, or speaker-selective. They showed that a DNN specifically trained to handle specific speakers, SNR levels, or noises achieve significant improvement for seen and unseen acoustic conditions. The authors also showed that SNR level, gender, and noise are three significant factors that impact speech enhancement performance. Based on the last statement, Yu

et al. [13] proposed a speech enhancement system using an auto-encoder with a multi-branched encoder. This model is made up of two stages. A decision tree algorithm is employed to categorize input speech signal based on the utterance-level and signal-level attributes in the first stage. In the second stage, all the branches of the encoder perform speech enhancement, and the results are then integrated into the decoder to determine the final enhanced speech. The authors showed the superiority of this model over several baseline models in terms of objective metrics and ASR results. Although these methods achieved good speech enhancement results, they suffer from two major issues. First, all these methods have a very high computational complexity since they employ a number of sub-DNNs, performing simultaneously. Second, the contributions of different components are unclear and difficult to describe.

According to the above observations, a system made up of a combination of sub-DNNs designed for a specific and simpler task could generalize better to unseen conditions. Besides, utterance-level attributes of voice signals vastly differ depending on the SNR level, noise, and speaker. Thus, it is expected that a single DNN should be highly complex to model all these modes, while a combination of several low-complexity sub-DNNs that are specifically designed to perform a particular task could handle different scenarios. Hence, we propose a multi-mode mapping-based speech enhancement framework, called MBSE for simplicity, performing in the MDCT domain. MBSE framework has two main stages: classification and mapping. In the former stage, input speech is classified regarding its utterance-level attributes, i.e., SNR level and gender. It is worth pointing out that we only consider SNR level and speaker gender to design the classifier since there is a wide range of noises in the real-world environment, which cannot be included in the training dataset. Four well-trained DNNs specialized for different specific and simple tasks, called expert DNNs, perform the mapping in the latter stage.

Unlike the previous chapters, where the methods were masking-based, the expert DNNs in the MBSE framework directly map noisy speech short-time modified discrete cosine transform (STMDCT) to the clean one. Furthermore, we reduced the latency by %55 than the models in previous chapters. In addition, since the MBSE framework is designed to perform in the STMDCT

domain, there is no need to deal with the phase information, i.e., no phase-related computation is required. Also, the training target length is only half of those in the previous chapters, which means lower computations and less demand for the expert DNNs. Moreover, only one of the low-complexity expert DNNs is active at each time, i.e., the computational burden is tied to only a single branch at each time step, although there are multiple branches in the model. Moreover, the expert DNNs are fully convolutional, and their computations are performed in parallel, unlike recurrent models. Finally, all the convolutions in this framework are causal in the sense that no future information is used in their calculations, as shown in Fig. 18 in Section 3.2.1. Our analysis and experimental studies reveal that the proposed MBSE framework yields a significantly improved speech enhancement performance compared to several existing DNN based methods while exhibiting a significantly lower computational complexity and memory footprint. The advantages of the proposed MBSE model over some well-known DNN-based speech enhancement methods are demonstrated through extensive comparative experiments.

The rest of this chapter is organized as follows: the transformation from the time domain to the MDCT domain and then the reconstruction of the time-domain speech based on MDCT values are introduced in Section 5.2.1. The speech classifier along with different classification approaches are explained in 5.2.2. Section 5.2.3 explains the fully-convolutional expert DNNs architecture. The complete system description is then provided in Section 5.2.4. Finally, the experimental results and comparisons are provided in Section 5.3.

5.2 Proposed MBSE Framework

A high-level block diagram of the proposed MBSE framework is shown in Fig. 43. The MDCT of framed input signal is first calculated and passed to the next block. At the classification stage, the input speech is categorized based on gender and SNR level, i.e., "male at high SNR level", "male at low SNR level", "female at high SNR level", or "female at low SNR level". According to the classification results, one of the expert DNNs then performs the mapping between noisy and

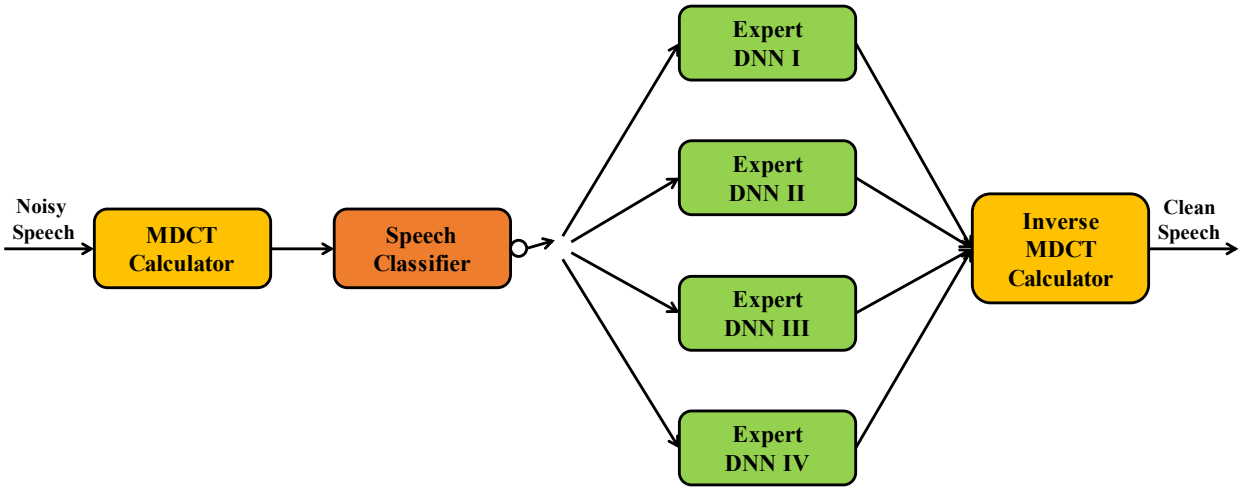


Figure 43: High-level block diagram of the proposed framework.

clean speech MDCT. Finally, the inverse MDCT (IMDCT) is computed and the clean speech in time domain will be delivered. The individual components of the MBSE framework are discussed in the following.

5.2.1 Modified Discrete Cosine Transform

Unlike Fourier transform, the output length of MDCT as a lapped transform is half of its input, i.e., $F : \mathbb{R}^{2N} \rightarrow \mathbb{R}^{N}$, where $\mathbb{R}$ is the set of real numbers. Consider a signal in time domain x_t framed into K segments each with length L. The k^{th} segment can thus be defined below,

$$x_k = (x_{(k-1)L+1}, x_{(k-1)L+2}, ..., x_{kL})^{\intercal} \tag{47}$$

where $(\square)^{\intercal}$ denotes transposition. If two consecutive segments are concatenated, the MDCT and IMDCT for the k^{th} time frame is computed as follows,

$$X_k^C = CW \begin{bmatrix} x_{k-1} \\ x_k \end{bmatrix} \tag{48}$$

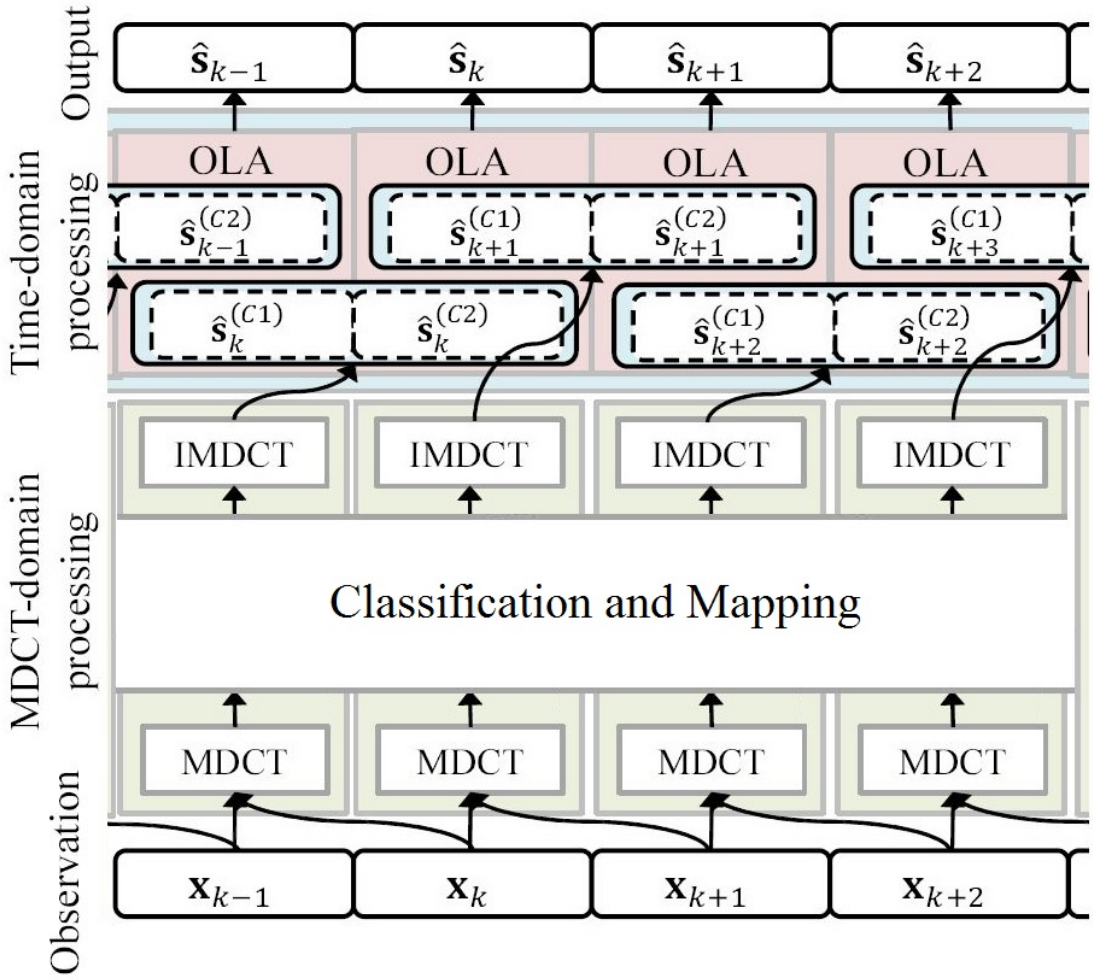


Figure 44: Speech enhancement in MDCT domain.

$$x_k^C = \begin{bmatrix} x_k^{(C1)} \\ x_k^{(C2)} \end{bmatrix} = WC^\intercal X_k^C \tag{49}$$

where $X_k^C = (X_{1,k}^C, ..., X_{L,k}^C)^\intercal$ defines the vectorized MDCT coefficients. The MDCT transformation is denoted by matrix $C \in \mathbb{R}^{L \times 2L}$ with its $(p,q)^{th}$ element being given by,

$$C_{p,q} = \sqrt{\frac{2}{L}} cos\left[\frac{\pi}{L}\left(p + \frac{1}{2}\right)\left(q + \frac{L+1}{2}\right)\right] \tag{50}$$

$$p = 0,\ 1,\ ...,\ L-1\,;\ q = 0,\ 1,\ ...,\ 2L-1$$

where p and q are, respectively, indices in MDCT and time domain. In addition, $W \in \mathbb{R}^{2L \times 2L}$ is a diagonal matrix representing the analysis/synthesis window, whose diagonal elements are defined as,

$$W_{l,l} = sin\left[\left(l + \frac{1}{2}\right)\frac{\pi}{2L}\right]. \tag{51}$$

The IMDCT vector components, $x_k^{(C1)}$ and $x_k^{(C2)}$, are corrupted by time-domain aliasing since C is an $L \times 2L$ matrix. To avoid this aliasing and perfectly reconstruct the original signal, two subsequent IMDCT vector components have to be added as follows,

$$x_k = x_k^{(C2)} + x_{k+1}^{(C1)} = O \begin{bmatrix} x_k^{(C)} \\ x_{k+1}^{(C)} \end{bmatrix} \tag{52}$$

where $O = [0, I, I, 0]$ is the overlap-add (OLA) matrix, in which I and 0 denote identity and zero matrices with size $L \times L$ [138]. Hence, speech signal can be easily transformed to a low-dimensional space, MDCT, without any information loss, and then be easily transformed back to the time domain. The entire mentioned process is shown in Fig. 44.

5.2.2 Speech Classification

Speech signals have different acoustic features considering their utterance-level attributes, such as noise, speaker, and SNR level. In [13], a t-distributed stochastic neighbor embedding (t-SNE) analysis [141] is conducted on the utterances from WSJ dataset [142]. t-SNE is a non-linear technique for dimensionality reduction on original data, and is commonly used to visualize high-dimensional datasets. Fig. 45 illustrates the analysis results for the gender and SNR attributes. It is worth mentioning that the axes do not refer to any physical meanings, just like other dimensionality reduction techniques such as principal component analysis (PCA). As shown in Fig. 45 (a), where yellow and purple dots represent male and female, respectively, which shows a clear distinction between speech attributes of males and females. There is also a clear separation between attributes of speech with high and low SNR levels, as shown in Fig. 45 (b, c) where yellow dots show SNR

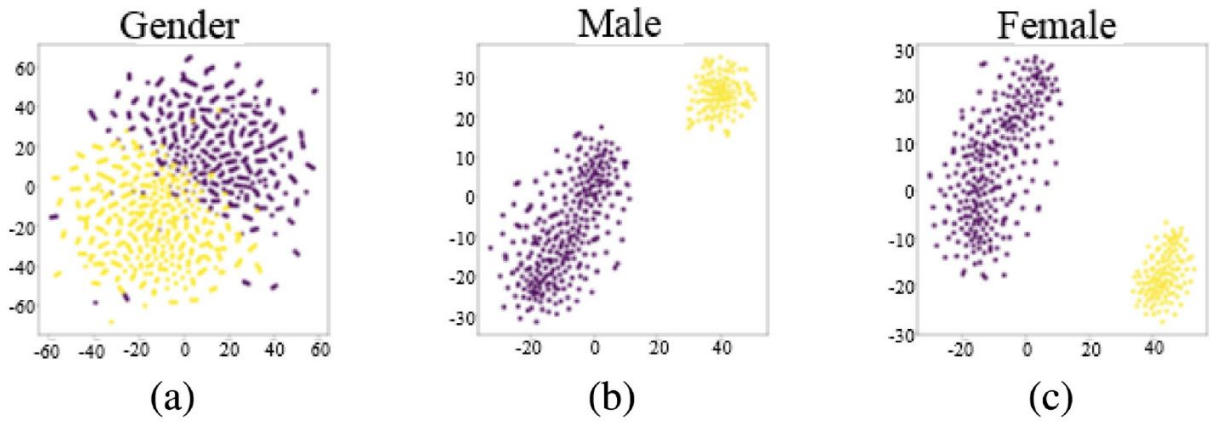


Figure 45: Clear distinction of utterances with different utterance-level attributes, (a) distinction based on gender, male and female: yellow and purple dots, respectively, (b) distinction based on SNR-level for male, high and low SNR levels: yellow and purple dots, respectively, (c) distinction based on SNR-level for female, high and low SNR levels: yellow and purple dots, respectively [13].

level of 10 dB and above, while purple ones represent that of 10 dB and below in both gender partitions.

Speech enhancement algorithms can benefit from the above-mentioned distinctive attributes to improve their performance. As such, we propose using low-complexity expert DNNs specialized on different tasks to perform speech enhancement for utterances with different attributes, i.e., "male at high SNR level", "male at low SNR level", "female at high SNR level", or "female at low SNR level". To this end, a speech classifier is required to determine in which category the input signal falls.

There are two types of classification: binary and multi-class. The former refers to the data with only two possible outcomes, like benign and malignant tumors. The latter yields with multiple classes, like object detection where the object could be any different types of objects. Besides, there are many types of supervised classification algorithms in machine learning, such as naive bayes (NB) [143], support vector machine (SVM) [144], k-nearest neighbors (KNN) [145], decision tree (DT) [146], linear discriminant analysis (LDA) [147], ensemble learning (EL) [148], deep neural networks (DNN), etc. We categorize these algorithms into two classes based on whether they are

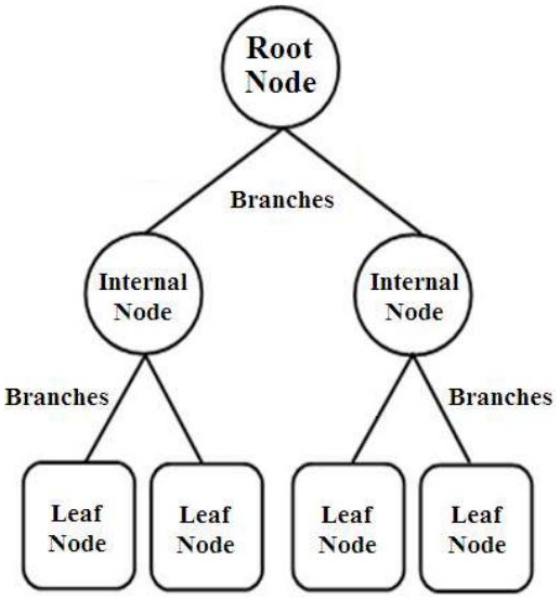


Figure 46: DT made up of three main components: nodes, branches, and leaves [14].

DNN-based or not. We aim to compare these two classes for the desired speech classification task. Some of these most commonly used algorithms are first briefly explained below.

Machine Learning Algorithms for Classification

Naive Bayes (NB): This technique is one of the first and most popular algorithms presented as a machine learning method for classification. NB is based on Bayes' theorem with the assumption that the presence of a particular feature in a class is unrelated to that of any other feature, which is the reason behind calling it naive. This classifier is suitable for both binary and multiclass classification and predicts data based on historical results. This family of classifiers is highly scalable which makes them useful for very large data sets. In this thesis, we use Gaussian NB where the likelihood of features is assumed to be Gaussian.

Decision Tree (DT): DTs, recently referred to as classification and regression trees, build models in the form of a tree structure to evaluate an instance of data. DTs are easy to understand, interpret, and visualize since they mimic a decision-making model in the form of a tree.

The process starts at the tree root with the provided dataset, and moves down to the leaves,

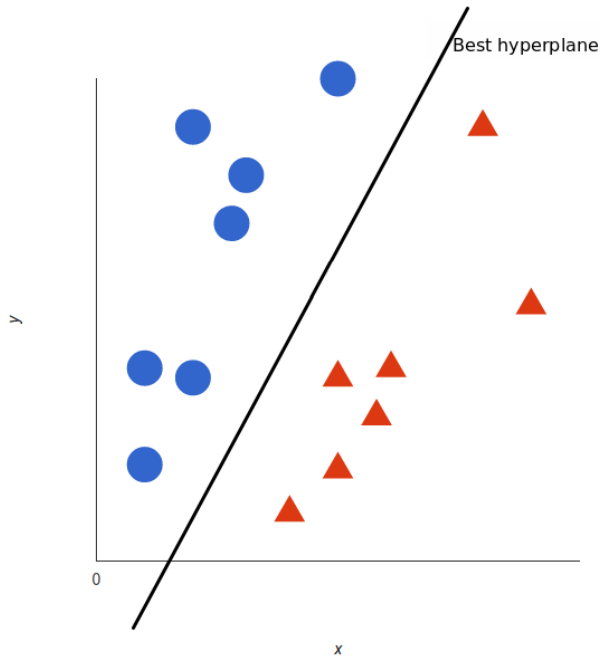


Figure 47: SVM for binary classification [15].

small subsets, until a forecast can be made. An example of a DT is shown in Fig. 46. It is made up of three main components: nodes, branches, and leaves, which represent the possible attributes associated with an event, attribute values, and classes, respectively [14]. The best predictor that asks the main question is the topmost decision node in a tree which is called the root node, and the leaf nodes represent the final decision or classification.

Support Vector Machine (SVM): SVM is one of the most popular classification algorithms because of the quite accurate results while requiring low computational power. It is effective for high-dimensional data and when the number of samples is smaller than feature dimensions. SVM aims to find an M-dimensional hyperplane, where M is the feature size of the data, that makes the furthest distance between data points of different classes. The accuracy of SVM directly depends on how accurate this hyperplane is calculated. As shown in Fig. 47, the samples of different classes place in both sides of the estimated hyperplane, in the case of binary classification [149].

K-Nearest Neighbors (KNN): One of the simplest supervised classification algorithms is KNN that has been used for both classification and regression tasks. This technique aims to label samples in dataset based on their proximity to each other, i.e., proxy is considered as sameness. The

algorithm requires a set of data with predefined classes and use them to label other samples. In other words, a new sample is labeled similar to k of its nearest neighbor samples [150]. It is worth noting that the choice of k depends on the dataset. Typically, a large k suppresses noise while making the classification boundaries less distinct.

Linear Discriminant Analysis (LDA): LDA is a linear multiclass classifier. The representation of LDA contains the statistical properties of different classes in the desired dataset. In the simplest case where the input is a single variable, the statistical properties are the variable's mean and variance of each class. The statistical properties are means and the covariance matrix calculated over the multivariate Gaussian for multiple variables. These properties are computed over the desired dataset, and the LDA equation then makes the predictions. The model employs Bayes' theorem for probability estimation [151]. In effect, LDA performs supervised dimensionality reduction of input data features by projecting the input data to a linear subspace while maximizing the discrimination among different classes. Dimensionality reduction lessens computations for a classification task and helps avoiding over-fitting through minimizing the error in parameter estimation [152].

Ensemble Learning (EL): EL combines multiple learning algorithms to achieve better performance than a single one. EL was first introduced to improve the performance of the classification and prediction models. Then, it was also used to check the predictions, select optimal parameters of the model, etc. In this thesis, we employ an extra-trees classifier [153] where a meta estimator is implemented that fits a number of extra-trees on various sub-samples of the desired dataset. It then takes an average over the models to improve the performance and control the over-fitting.

CNN for Classification

CNN has been widely used in classification tasks due to its remarkable capability in extracting important features and thus its high performance. It has been first used for image classification, leading to promising accuracy. The advent of large image datasets such as ImageNet [154], CNN-based frameworks like AlexNet [155], VGG [156], Inception [157], and ResNet [72] have achieved remarkable improvement in image classification. Since the speech spectrogram can be considered

as a 2D image, CNN has also been used for audio scene classification in many studies, such as [158, 159] showing good classification accuracy. In our work, we aim to compare the performance of a CNN as the multiclass classifier with that of the aforementioned non-DNN-based ML algorithms for classifying speech in terms of the SNR level and speakers' gender.

The employed CNN network comprises three convolutional layers to exploit useful patterns in their inputs and thus provide a set of appropriate features for the desired classification task. The number of CNN channels and their size are decreased along the network to lower the dimensionality of CNN feature maps, like in [160]. Afterward, a global max-pooling layer is employed to decrease the spatial dimensionality of the CNN feature maps. This layer selects the maximum value of the features in the last convolutional layer. The max-pooling operation also removes the noise and unnecessary information from its input. Finally, a very small FC network is utilized to perform the final prediction based on the extracted features. A dropout technique is also adopted in this FC network to avoid over-fitting. It is worth mentioning that all the activation functions in this classifier are ReLU except for the last layer where softmax is utilized. The details of this network configuration will be explained in Section 5.2.4.

5.2.3 Mapping-Based Expert CNNs

CNN can perform well regression in speech enhancement due to its low complexity and high performance. Also, it is amenable to parallel computations that leads to low processing time. CNN has been used as the mapping function to transfer the noisy speech spectrogram to the clean one in [5, 12], demonstrating promising speech enhancement results.

In this work, we need very low-complexity CNNs that are specialized on different tasks for mapping the noisy STMDCT to the clean one. Hence, we employ a fully-convolutional CNN comprising two stages: feature extraction and regression. In the first stage, a CNN made up of five layers with 2D dilated frequency convolutions is used to extract necessary features considering contextual information of input speech spectrogram. In the second stage, another three-layer CNN with 1D convolutions is employed to transform the extracted features in the previous stage to the

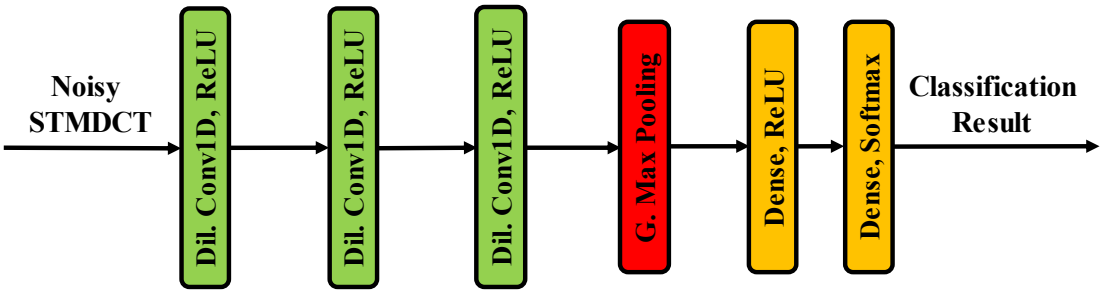


Figure 48: CNN classifier network architecture.

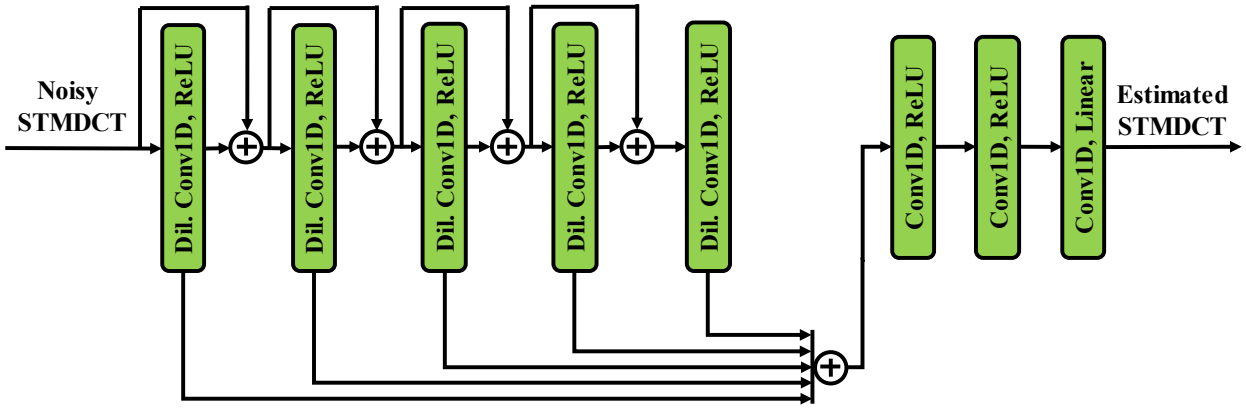


Figure 49: CNN expert network architecture for mapping.

clean STMDCT. Residual learning and skip connection techniques are also used to facilitate learning and accelerate the convergence. The details of this fully-convolutional CNN will be explained in the next section.

5.2.4 MBSE Framework Architecture

Two main components of the MBSE framework are the classifier and expert networks. The proposed classifier network is shown in Fig. 48. It is basically a stack of convolutional layers, max-pooling, and FC layers. In particular, the first convolutional layer contains N kernels of size (9 $\times$ 1) followed by a ReLU activation function. To reduce the size of feature maps and lessen computations, padding *valid* is chosen, which means that no padding is added to the inputs of the convolutional layers. Assuming the shape of input spectrogram is ($F \times T$), the output of this convolutional layer will contain N feature maps with the shape of ((F-9)/2, (T-1)/2). It is worth

mentioning that the stride is 1 in all the convolutional layers, i.e., the jump size of the kernels on their input is 1. Everything is the same for the second and third layers except that the kernel size and the number of kernels are (7×1) and $N/2$ for the second layer, and (5×1) and $N/4$ for the third one. Afterward, a max-pooling layer reduces the spatial dimension of the feature maps of the previous layer. As such, the output size of this layer will be a vector of size $N/4$ to be passed to the next layer. A two-layer FC network performs the classification with the information extracted by CNN and the max-pooling layer. The first layer contains 16 nodes with a ReLU activation function and a dropout rate of 0.3. The final layer comprises four nodes, the same as the number of decisions that the classifier has to make. The activation function of the last layer is softmax that calculates a probability for every possible class. Based on the results of this classifier, one of the expert networks is selected to perform the main mapping between the noisy input STMDCT and clean one.

Another main component of the MBSE framework is the mapping network. Four identical very low-complexity CNNs are trained with different training datasets. The structure of these networks is shown in Fig. 49. Each of these CNNs is thus specialized on different mapping-based tasks. These fully-convolutional CNNs are comprised of two stages: feature extraction and regression. In the former stage, there is a stack of five CNN layers with dilated convolutions and increasing dilation rates of 1, 2, 4, 8, and 16 which leads a receptive field of 125. Padding in these layers is *same*, i.e., the input size and feature maps are the same size. The number of kernels in all these layers is 32, with ReLU activation function. The convolutions are 2D with kernel sizes of (3×5). The feed-forward lines around these layers are residual paths, in the form of convolutional layers with kernel size (1×1), are used to improve the training procedure. As shown, the outputs of each layer are added up (with a skip connection) to make the output of the feature extraction part of the CNN network. Then, there is a three-layer CNN with 1D convolutions and kernel size (1×7) to perform the regression. The activation function of first two layers is ReLU and that of the last layer is linear. The number of channels of these layers is 48, 48, and 1 where the last one transforms the output of the middle layer to the final output of the CNN network. Finally, the time-domain clean

speech is reconstructed based on the estimated STMDCT matrix as explained in Section 5.2.1.

5.3 Experiments

5.3.1 Experimental Setup

To evaluate the performance of the proposed MBSE model, clean utterances are selected from the TIMIT database [81], with 1050 clean utterances spoken by males and the same number by females randomly selected, and the duration of each utterance is between 2.5 and 5 seconds. Note that no utterance is repeated in this random utterance selection from the TIMIT database. From these 1050 selected utterances, 1000 speech files are used for training and 50 for testing. Also, a set of utterances from LibriSpeech [87] is chosen for the testing stage. This testing set contains 100 utterances, with the length of 3 seconds, half spoken by males and another half by females. The sampling rate is 16 kHz. The noise files are selected from NOISEX-92 [91]. The same setting as described in Section 3.3.1 is used to prepare the pairs of noisy and clean speech. Four unseen highly-nonstationary noises, namely, *Coffee Shop, Busy City Street, Car Interior,* and *Street Traffic*, are selected from [117] to evaluate the generalization capability of the proposed model. The speech STMDCT is directly fed to the classifier and expert networks as input.

For the purpose of training and testing the classifier, each of the 1050 utterances spoken by males in the dataset is mixed with different noises at SNR levels of -6, -3, and 0 dB to create the *male utterances with low SNR* sub-dataset and mixed at SNR levels of 3 and 6 dB to create the *male utterances with high SNR* sub-dataset. Via the same procedure, the mixtures for *female utterances with low SNR* and *female utterances with high SNR* sub-datasets are obtained. Then, the STMDCT of these utterances are calculated where the time-domain input length is twice the length of each frame, i.e., 320 samples, since the input of MDCT should be the concatenation of two frames as explained in Section 5.2.1, and the output length is 160 samples in STMDCT domain. Since we want the latency to be lower than the methods in Chapters 3 and 4, random chunks with the length of 5 frames from these mixtures are selected for the training, validation, and testing purposes.

Finally, the classifier and expert networks are separately trained with the associated datasets and then jointly tested.

The MSE is selected as the cost function for the expert networks, while the Adam optimizer is used as an extension to the stochastic gradient descent [118] to minimize the error between ideal (ground truth) and estimated STMDCT values. For the classifier, categorical cross entropy [161] is adopted. We will discuss some modifications about the loss function in Section 5.3.2.

PESQ, STOI, and SSNR are used as objective evaluation metrics; the higher these scores are, the better the speech enhancement.

The performance of the classifier is evaluated with accuracy and F1 scores. The accuracy score calculates the percentage of correctly predicted labels over all the predictions. The F1 score is valued between 0 and 1 and is defined as a weighted average of the precision and recall. In a binary classification analysis, precision is the number of true positive labels divided by all positive labels, and recall is defined as the number of true positive labels divided by that of all labels should have been identified as positive. These two definitions are shown in Fig. 50. Specifically, the F1 score is defined as,

$$F1 = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \tag{53}$$

In the multiclass case, the F1 score is calculated by averaging over F1 scores of different classes [16, 153].

Since the accuracy score can be misleading when the number of observations is unequal for different classes, or in the case of multiclass problems, a confusion matrix [162] is usually employed to summarize the classification performance. The confusion matrix has a square shape where the rows represent the true instances and the columns represent the predicted ones. The confusion matrix indicates what type of errors the classifier is making and where the model is confused.

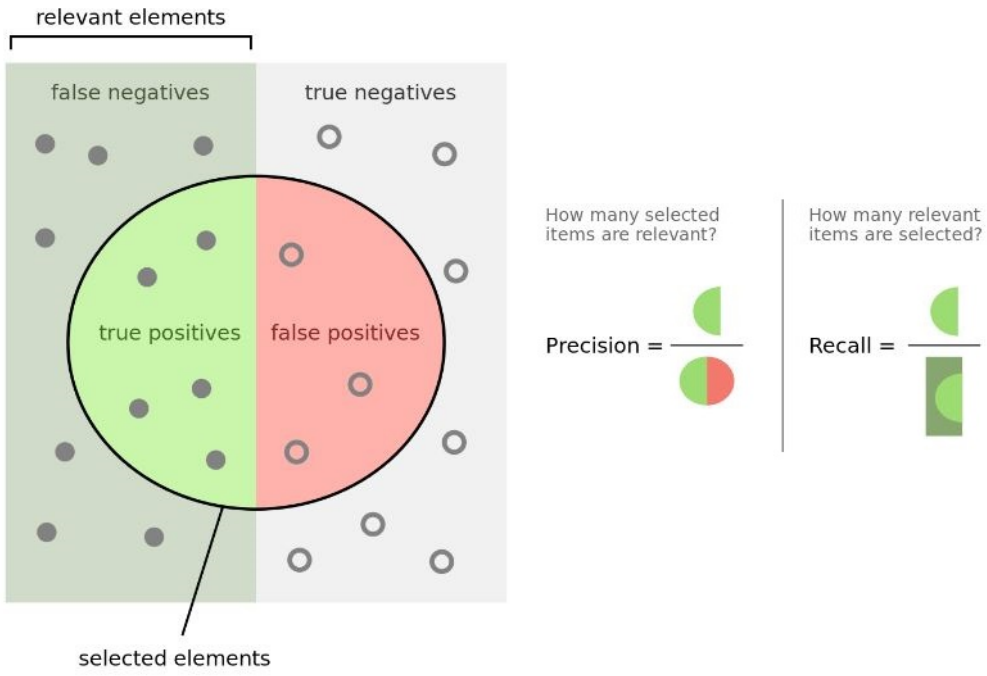


Figure 50: Precision and Recall [16].

5.3.2 Low-Complexity High-Performance Classifier

The classifier in the MBSE framework aims to classify input speech in terms of SNR level and speaker gender. We design this classifier with a CNN network under several considerations in order to improve its performance regarding accuracy and computations. In the following, we investigate some techniques to optimize the CNN for the desired classification task.

Complexity Reduction Using Depth-Wise Separable Convolution in Classifier

Convolution measures the amount of overlap of two functions as one slides over another. Since the standard convolution is slow to perform, an alternative technique called depth-wise separable convolution has been introduced in [163] to speed up this operation.

A standard convolution operation is shown in Fig. 51. Consider an input with dimensions of $D_f \times D_f \times M$ where D_f is the input's height and width, and M is the number of channels. If a kernel with the shape of $D_k \times D_k \times M$ is convolved with this input, the output's shape will be $D_g \times D_g \times 1$. If we apply N such kernels to the input, the output will have the shape of $D_g \times D_g \times N$.

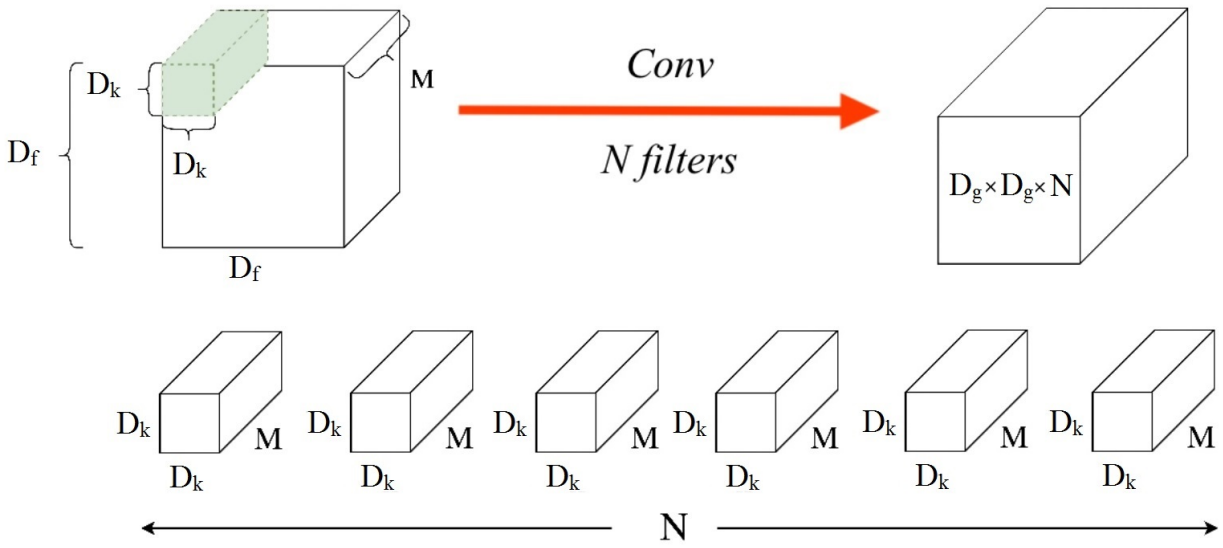


Figure 51: A standard convolution operation.

To measure the computations of this convolution, we count the number of multiplications. The number of multiplication for one convolution is $D_k^2 \times M$ in the standard convolution. When this kernel slides over the input, the number of multiplications will be $D_g^2 \times D_k^2 \times M$. And, having N such a kernel leads to $D_g^2 \times D_k^2 \times M \times N$ multiplications.

Depth-wise separable convolution comprises of two stages: filtering (depth-wise) and combination (point-wise). The process is shown in Fig. 52. Depth-wise convolution applies convolutions to a single input channel each time while the standard convolution does it to all the channels. As shown in the figure, each of M kernels with the shape of $D_k \times D_k \times 1$ is applied to only one channel. As such, by stacking the output of these M convolutions, the final output will have a shape of $D_g \times D_g \times M$. In the latter stage, a point-wise convolution involves the linear combination of each of its input layers. If a kernel with the shape of $1 \times 1 \times M$ is applied to the previous stage's output, the resulting output will have the shape of $D_g \times D_g \times M$ that is the desired output shape.

Here we calculate the number of multiplications of the depth-wise separable convolution. The number of multiplications for one convolution is D_k^2 in the first stage. When these kernels are applied over all the input channels, the multiplications will be $D_k^2 \times D_g^2$. Applying such kernels over all the input channels will result in a total number of multiplications of $D_k^2 \times D_g^2 \times M$ in the first stage. In the second stage, the number of multiplications for one instance of convolutions

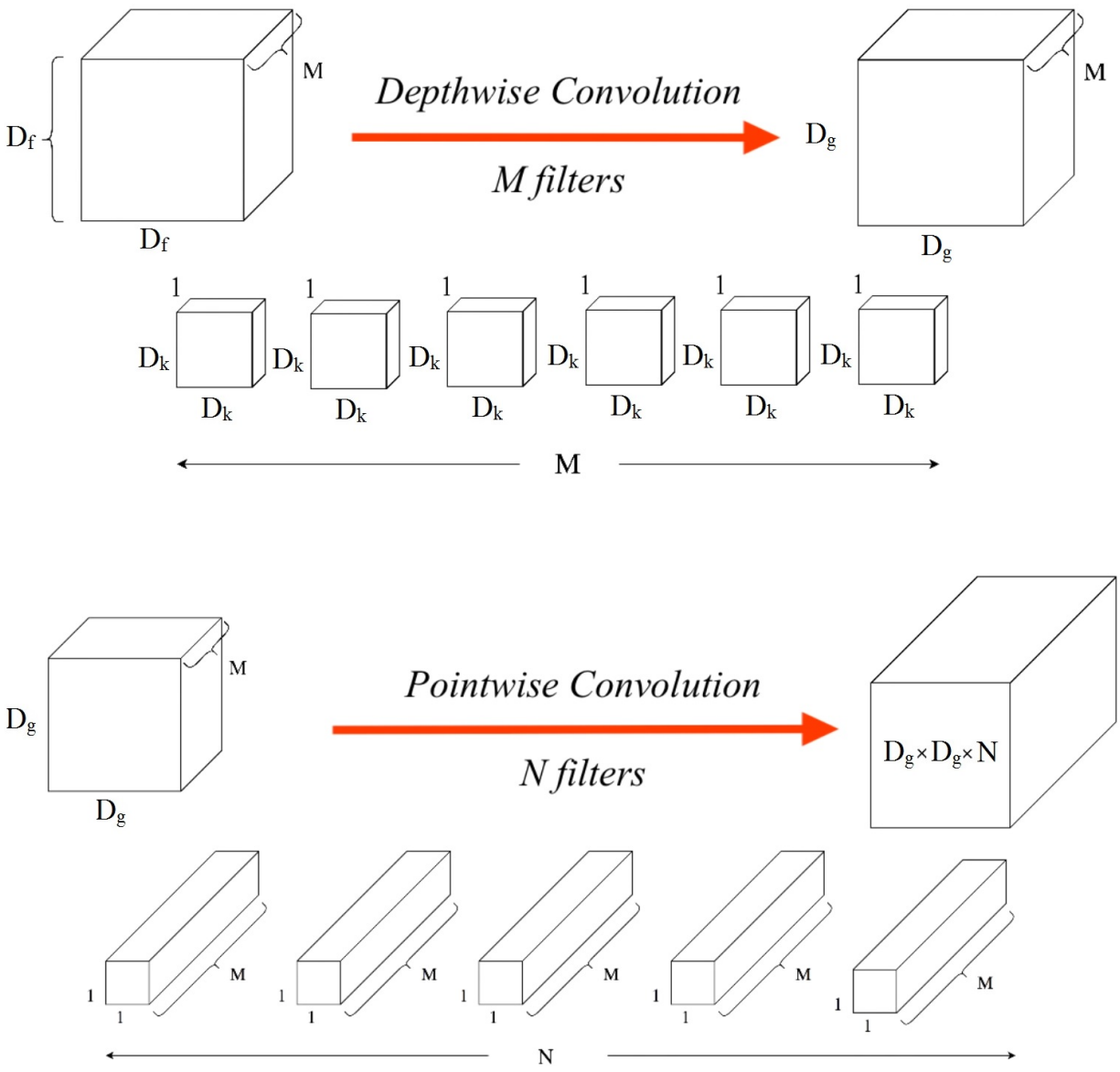


Figure 52: Depth-wise separable convolution comprising two stages: filtering (depth-wise convolution) and combination (point-wise convolution).

with the kernel size of $1 \times 1 \times M$ is M. Applying this kernel to the entire output of the first stage with the shape of D_g^2 will result in $D_g^2 \times M$ multiplications. Hence, it requires $D_g^2 \times M \times N$ multiplications for N kernels. Thus, the total number of multiplications for the entire convolution will be $D_g^2 \times D_k^2 \times M + D_g^2 \times M \times N$.

Table 18: Comparison of different convolution processes.

Method	Accuracy	F1 Score	Parameters (k)	FLOPs (M)
Standard Convolution	**75.6**	**73.8**	17.9	0.68
Depth-Wise Separable Convolution	72.5	69.5	**4.2**	**0.15**
Difference (in %)	-3.1	-4.3	+75.5	+77.9

To compare the number of multiplications of the standard and depth-wise separable convolutions, we can compute the ratio of the total number of their multiplications, as below,

$$\frac{No.\ multiplications\ in\ depthwise\ separable\ convolution}{No.\ multiplications\ in\ standard\ convolution} = \frac{D_g^2 \times (D_k^2 + N) \times M}{N \times D_g^2 \times D_k^2 \times M} = \frac{1}{N} + \frac{1}{D_k^2} \tag{54}$$

To get a better perspective on how much the depth-wise separable convolution decreases the number of multiplications, we present an example. Assume $N = 1024$ and $D_k = 3$. Plugging these values into the above equation gives a ratio of 0.112 which means the number of multiplications is reduced by almost %90 [164].

This section compares the standard and depth-wise separable convolutions in the classifier in terms of accuracy, F1 score, the number of parameters, and FLOPs. The results of this comparison are shown in Table 18, where k and M denote kilo and million. As seen, using depth-wise separable convolution causes %3.1 and %4.3 deterioration in accuracy and F1 score. However, the number of parameters using the standard convolution is about 18 k while that using depth-wise separable convolution is only 4.2 k, which means %75.5 reduction in the number of parameters. In addition, the required computation that we measure in terms of FLOPs drops from 0.68 M to 0.15 M, i.e., %77.9, which saves a tremendous amount of computation. Hence, depth-wise separable convolution remarkably improves the model performance by reducing the computations.

All in all, we can conclude that the designed classifier for classifying the input speech into four different categories based on input speech segments with the length of only five frames is super low complexity. In the following, some techniques are presented to vastly improve the accuracy and F1 score.

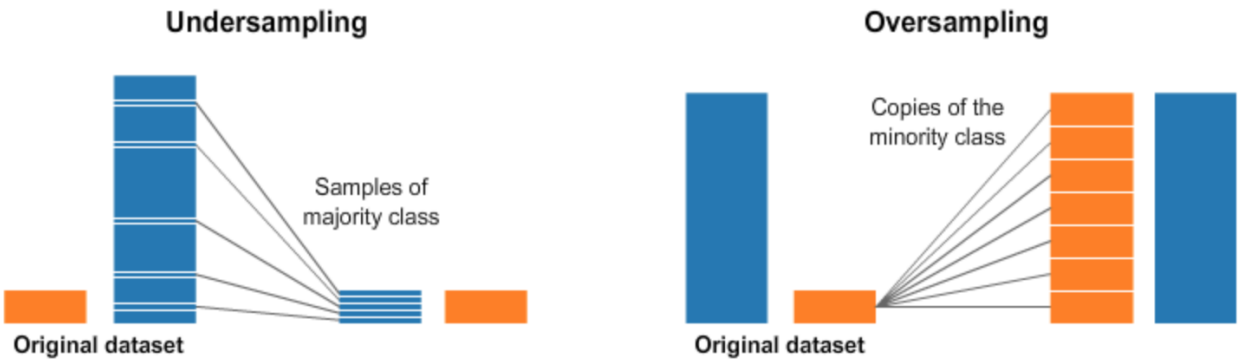


Figure 53: Visualization of random oversampling and undersampling.

Dealing with Imbalanced Dataset

Imbalanced datasets refer to those with a skew in the class distribution. The imbalanced number of samples of different classes in the training dataset can impact the machine learning algorithm performance in a way that the machine becomes biased to the classes with more samples. There are two main approaches to address this issue: random resampling of the training dataset and using a weighted loss function.

Resampling of the training dataset refers to creating a transformed version of the dataset wherein the numbers of samples of different classes are almost equal. Random resampling is to choose samples for the transformed dataset randomly. There are two main approaches to create the balanced dataset: random oversampling and undersampling, as shown in Fig. 53. In the former, some instances from classes with lower number of samples are randomly chosen and duplicated. In the latter, some instances from classes with higher number of samples are randomly chosen and removed [165]. Random oversampling and undersampling can also be combined to create a balanced dataset through randomly duplicating or removing samples from different classes.

Another approach to address the imbalanced dataset problem is to introduce different weights to the loss function for different classes. Depending on the number of samples of each class in the training dataset, the computed loss for different samples is weighted differently in a way that the higher weight is given to the computed loss for samples from the classes with less number of instances.

Table 19: Comparison of different techniques to address imbalanced dataset issue.

Method	Accuracy	F1 Score
Weighted Loss Function	78.1	75.8
Random Resampling	**89.5**	**89.5**
Difference (in %)	+11.4	+13.7

Table 20: Comparison of different attention techniques in terms of accuracy and F1 score..

Method	Accuracy	F1 Score
No Attention	89.5	89.5
Channel-Wise Attention	88.4	88.5
Spatial Attention	**91.4**	**91.5**

In this work, we mix the clean utterances of males and females with different noises at different SNR levels to create four categories "male at high SNR level", "male at low SNR level", "female at high SNR level", or "female at low SNR level". However, the SNR levels of -6, -3, and 0 are considered as the *low SNR* while those of 3 and 6 are considered as *high SNR*. Hence, the number of samples in the *high SNR* categories is two-third of that in the *low SNR* ones. As such, the training dataset is imbalanced. In this section, we compare both above-mentioned techniques to address the imbalanced dataset problem. Table 19 shows the comparison results with both techniques in terms of accuracy and F1 score. As seen, the random resampling technique improves the accuracy by %11.4 and more importantly improves the F1 score by %13.7. Hence, we can learn the importance of having a balanced dataset to make the leaning machine unbiased to some categories with more number of training samples.

Attention Benefit in Classifier

As mentioned in Section 3.2.1, CNN contains many feature maps that may have different levels of significance. Accordingly, emphasizing informative feature maps improves the model performance. An attention mechanism adaptively emphasizes the informative ones by recalibrating feature maps while suppressing others.

This section investigates if and how using the attention technique will improve the classification results in our work. Two attention techniques are considered in this section: channel-wise and

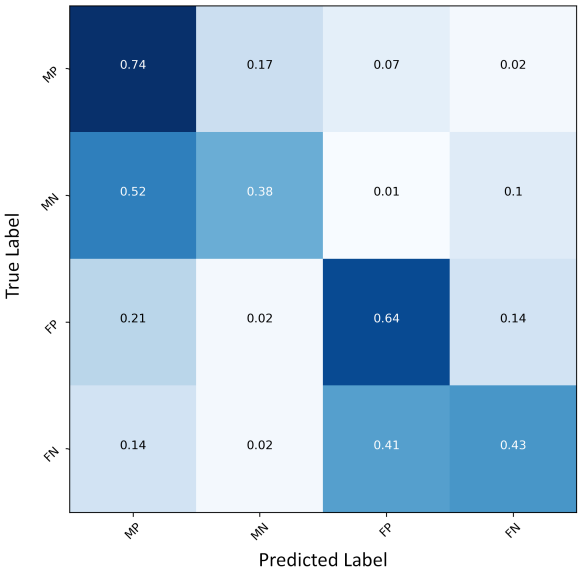
Table 21: Comparison of different classifiers.

Method	Accuracy	F1 Score
NB	54.7	53.7
DT	63.8	63.5
SVM	77.5	77.2
KNN	76.8	75.5
LDA	72.7	72.7
EL	52.3	68.8
CNN	**91.4**	**91.5**

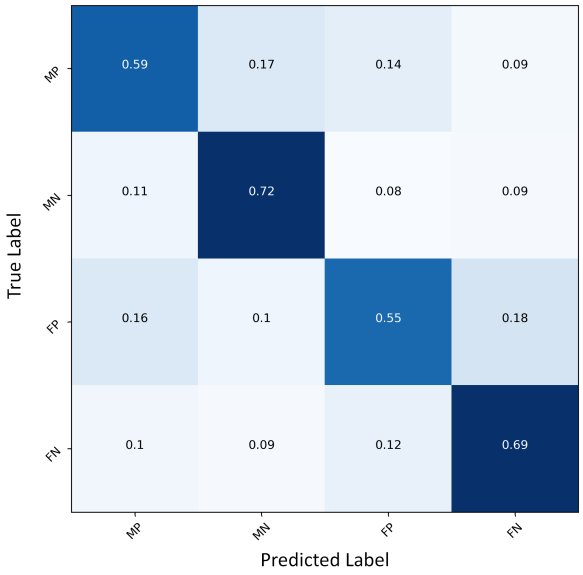
spatial, as explained in Section 3.2.1. These attention blocks are placed after the first and second convolutional layers. Table 20 shows the comparison results with both attention techniques along with using no attention in terms of accuracy and F1 score. As seen, although the channel-wise attention technique leads to no improvement, the spatial attention yields improvement in terms of both accuracy and F1 score.

Compare CNN with with non-DNN-based Machine Learning classifiers

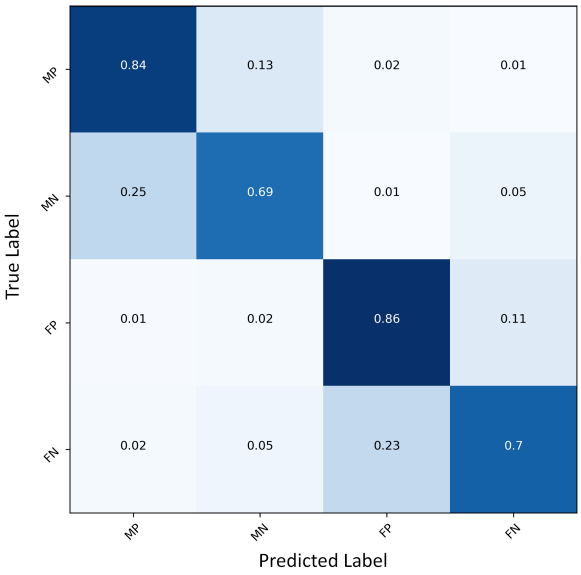
As mentioned in Section 5.2.2, there are many types of supervised classification algorithms in machine learning, such as NB, DT, SVM, KNN, LDA, EL, and DNN. This section evaluates and compares these methods for our desired speech classification task. The comparison results are shown in Table 21. In addition, the confusion matrices for different methods are illustrated in Fig. 54. As shown in the table, the accuracy and F1 score achieved by CNN are quite better than those obtained by other methods. After CNN, the results by SVM are ranked second, and NB obtains the lowest results. In the confusion matrices, we can get a perspective about the type of errors the classifiers are making and where the model is confused. For example, in the SVM confusion matrix, Fig. 54 (c), it is clear that most of the misclassification results are because the model is confused between high and low SNR. As shown in this figure, %25 of the *male with low SNR* samples are classified as *male with high SNR*. In addition, %23 of the *female with low SNR* samples are classified as *female with high SNR*. However, in the CNN confusion matrix, we can see that all the classes are almost accurately classified while there are tiny misclassification results


MP
MN
FP
FN
0.74 0.17 0.07 0.02
0.52 0.38 0.01 0.1
0.21 0.02 0.64 0.14
0.14 0.02 0.41 0.43
True Label
Predicted Label

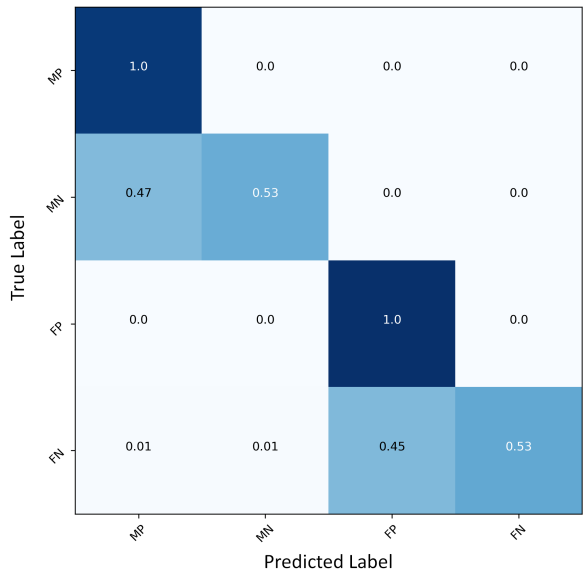
(a)


MP
MN
FP
FN
0.59 0.17 0.14 0.09
0.11 0.72 0.08 0.09
0.16 0.1 0.55 0.18
0.1 0.09 0.12 0.69
True Label
Predicted Label

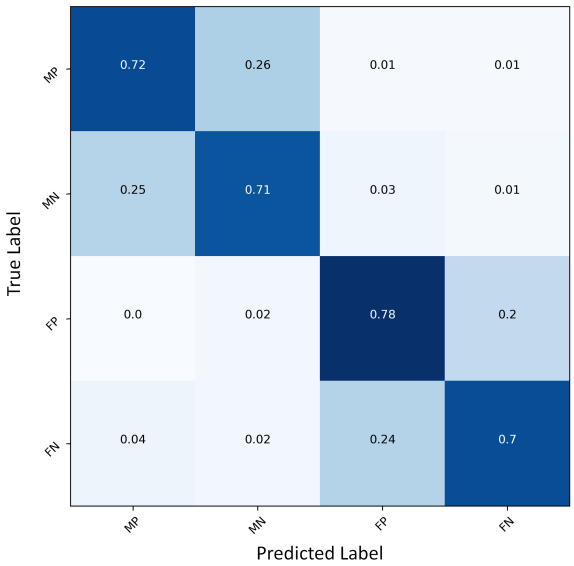
(b)


MP
MN
FP
FN
0.84 0.13 0.02 0.01
0.25 0.69 0.01 0.05
0.01 0.02 0.86 0.11
0.02 0.05 0.23 0.7
True Label
Predicted Label

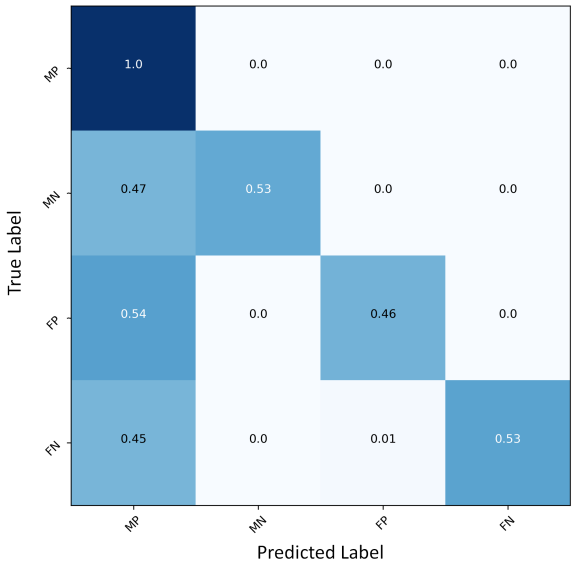
(c)


MP
MN
FP
FN
1.0 0.0 0.0 0.0
0.47 0.53 0.0 0.0
0.0 0.0 1.0 0.0
0.01 0.01 0.45 0.53
True Label
Predicted Label

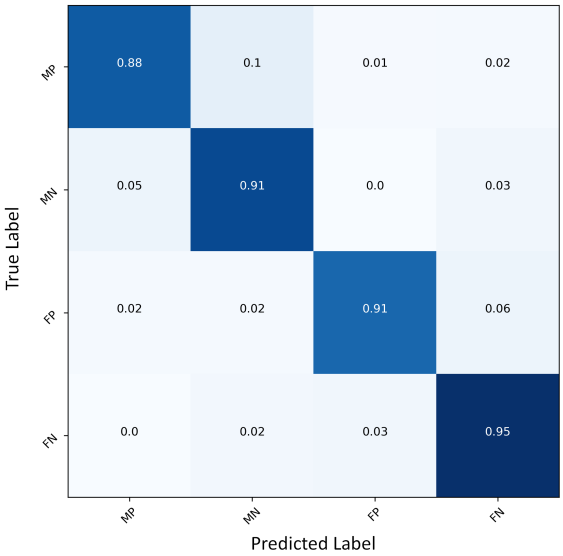
(d)



(e)



(f)



(g)

Figure 54: Comparison of different machine learning methods for classification: (a) NB, (b) DT, (c) SVM, (d) KNN, (e) LDA, (f) EL, (g) CNN. MP, MN, FP, and FN denote "male at high SNR level", "male at low SNR level", "female at high SNR level", or "female at low SNR level", respectively.

which is because the model is confused between the negative and positive SNRs. However, these misclassification results are negligible. All in all, we can see that the CNN classifier is the best choice to classify the input speech into four different categories based on input speech segments with the length of only five frames.

5.3.3 Comparison with Previous Methods

Here, we compare the performance of the MBSE framework with that of the neural network models developed in the previous chapters, i.e., serial hybrid network and PACDNN. Table 22 shows the domain and training target of these three methods along with their number of parameters, latency, and computational time. Note that *M* and *ms* denote million and millisecond. Obviously all the proposed models have a lower number of parameters as compared to those in the literature. Specifically, the number of parameters in the serial hybrid model was 1 million, while it was remarkably reduced by more than %80 in the PACDNN, and by more than %91 in the MBSE framework. Besides, the latency of the MBSE model is less than half of that of the serial hybrid and PACDNN. Moreover, the computational time of all the three different methods proposed in this thesis is very small, especially the MBSE framework takes only 0.22 ms per frame.

Table 22: Comparison of different methods in this thesis.

Method	Domain	Training Target	No. of Parameters (M)	Latency (ms)	Computational Time (ms)
Serial Hybrid	STFT	PSM	1.00	110	0.62
PACDNN	STFT	PSM+GD	0.19	110	0.45
MBSE	STMDCT	Clean Spectrogram	0.09	50	0.22

Tables 23 and 24 present the comparison of the three methods in terms of speech enhancement objective metrics, evaluated with utterances from the TIMIT dataset and 20 noises from the NOISEX dataset, as explained in Section 5.3.1. Table 23 shows the results for male utterances. As seen, the MBSE framework yields better results in terms of all the metrics at almost all SNR levels, except for the SNR level of -3 dB, where PACDNN performs better. The same comparison

for the female utterances is shown in Table 24, where again MBSE outperforms other methods.

Tables 25 and 26 compare the performances of the three methods where the testing noises and utterances are unseen. The noises are from [117] and the utterances are from LibriSpeech [87], as explained in Section 5.3.1. Note that bscs, cair, cfsp, and sttc denote *Coffee Shop, Busy City Street, Car Interior,* and *Street Traffic* noises. As shown, the MBSE framework achieves better results in almost all the cases, except for a few cases such as SNR level of -3 dB where PACDNN yields slightly better results. It is worth mentioning that the MBSE method performs very well while the number of its trainable parameters is quite lower than other methods.

Table 23: Comparison of the three proposed methods with **unseen male** utterances from TIMIT dataset mixed with different noises noises.

Method	PESQ					STOI					SSNR				
	-6	**-3**	**0**	**3**	**6**	**-6**	**-3**	**0**	**3**	**6**	**-6**	**-3**	**0**	**3**	**6**
Unprocessed	1.39	1.47	1.66	1.87	2.18	0.568	0.632	0.697	0.713	0.760	-13.2	-11.0	-8.61	-5.91	-3.14
Serial Hybrid	2.11	2.35	2.55	2.75	2.94	0.696	0.759	0.812	0.854	0.888	0.36	**1.88**	3.23	4.67	5.97
PACDNN	2.13	**2.38**	2.58	2.78	2.99	0.710	**0.771**	0.822	0.863	0.897	-0.3	1.86	3.30	4.85	6.53
Proposed	**2.20**	2.33	**2.61**	**2.80**	**3.00**	**0.750**	0.750	**0.834**	**0.879**	**0.911**	**1.72**	1.77	**3.50**	**5.45**	**7.19**

Table 24: Comparison of the three proposed methods with **unseen female** utterances from TIMIT dataset mixed with different noises noises.

Method	PESQ					STOI					SSNR				
	-6	**-3**	**0**	**3**	**6**	**-6**	**-3**	**0**	**3**	**6**	**-6**	**-3**	**0**	**3**	**6**
Unprocessed	1.03	1.13	1.34	1.57	1.77	0.532	0.592	0.656	0.717	0.720	-11.8	-9.97	-7.62	-5.00	-2.25
Serial Hybrid	1.84	2.09	2.34	2.57	2.79	0.669	0.729	0.784	0.829	0.866	1.35	**2.78**	4.15	5.42	6.63
PACDNN	1.95	**2.18**	2.40	2.61	2.82	0.682	**0.742**	0.796	0.841	0.877	0.84	2.70	4.45	6.18	7.89
Proposed	**2.11**	2.13	**2.42**	**2.68**	**2.91**	**0.722**	0.704	**0.809**	**0.847**	**0.884**	**2.53**	2.00	**4.47**	**6.45**	**8.10**

5.4 Conclusion

This chapter proposed a multi-mode and mapping-based MBSE framework for speech enhancement in the MDCT domain. The proposed framework comprises two main stages: classification and mapping. The former classifies the input speech based on its utterance-level attributes, i.e., SNR level and gender. The latter containing four well-trained CNNs performs mapping between

Table 25: Comparison of the three proposed methods with **unseen male** utterances from **LibriSpeech** dataset mixed with **unseen** noises.

SNR	Method	PESQ				STOI				SSNR			
		bscs	cair	cfsp	sttc	bscs	cair	cfsp	sttc	bscs	cair	cfsp	sttc
-6 dB	Unprocessed	1.15	1.75	1.23	1.38	0.434	0.680	0.474	0.556	-13.3	-13.2	-12.9	-11.8
	Serial Hybrid	1.44	2.54	1.47	1.67	0.499	0.783	0.530	0.611	-6.64	0.48	-3.93	-2.92
	PACDNN	1.54	**2.70**	1.59	**1.95**	0.545	**0.842**	0.601	0.695	-4.98	2.94	-3.36	-1.37
	Proposed	**1.76**	2.53	**1.81**	**1.95**	**0.628**	0.831	**0.670**	**0.735**	**-2.30**	**3.10**	**-0.32**	**1.19**
-3 dB	Unprocessed	1.31	1.96	1.41	1.58	0.507	0.716	0.537	0.611	-11.3	-10.7	-10.8	-9.73
	Serial Hybrid	1.74	2.70	1.73	1.99	0.593	0.814	0.626	0.721	-4.39	1.95	-1.81	-0.55
	Unprocessed	**1.84**	**2.91**	**1.84**	**2.23**	**0.649**	0.869	**0.679**	0.754	**-2.98**	**4.60**	-1.67	0.83
	Proposed	1.75	2.77	1.78	2.02	0.608	**0.883**	0.600	**0.762**	-4.91	2.68	**-1.33**	**1.25**
0 dB	Unprocessed	1.50	2.14	1.64	1.80	0.586	0.746	0.609	0.679	-8.73	-8.32	-8.05	-7.53
	Serial Hybrid	1.98	2.94	1.97	2.19	0.681	0.852	0.697	0.775	-2.26	3.66	-0.25	0.78
	PACDNN	2.13	**3.07**	**2.10**	2.25	0.739	0.885	0.750	0.815	-0.43	5.86	0.29	2.76
	Proposed	**2.27**	3.00	**2.10**	**2.46**	**0.757**	**0.901**	**0.786**	**0.835**	**-0.22**	**5.95**	**1.85**	**3.66**
3 dB	Unprocessed	1.70	2.35	1.85	1.99	0.660	0.781	0.673	0.743	-6.08	-5.56	-5.62	-4.82
	Serial Hybrid	2.25	3.08	2.28	2.43	0.769	0.876	0.784	0.829	0.09	4.91	1.79	2.72
	PACDNN	2.30	3.07	2.35	2.68	0.804	0.904	0.817	0.860	1.71	7.39	2.51	4.44
	Proposed	**2.38**	**3.22**	**2.43**	**2.75**	**0.830**	**0.926**	**0.848**	**0.877**	**2.67**	**7.79**	**3.98**	**5.42**
6 dB	Unprocessed	1.92	2.50	2.06	2.19	0.731	0.804	0.728	0.788	-3.18	-2.79	-2.89	-2.24
	Serial Hybrid	2.48	3.27	2.53	2.62	0.825	0.899	0.833	0.867	2.49	6.46	3.59	4.15
	PACDNN	**2.62**	3.40	2.68	**2.89**	0.853	0.917	0.855	0.890	3.73	8.45	4.46	6.15
	Proposed	2.60	**3.47**	**2.78**	2.76	**0.879**	**0.941**	**0.890**	**0.916**	**4.85**	**9.22**	**5.87**	**7.51**

Table 26: Comparison of the three proposed methods with **unseen female** utterances from **LibriSpeech** dataset mixed with **unseen** noises.

SNR	Method	PESQ				STOI				SSNR			
		bscs	cair	cfsp	sttc	bscs	cair	cfsp	sttc	bscs	cair	cfsp	sttc
-6 dB	Unprocessed	0.86	1.40	0.93	1.18	0.424	0.668	0.462	0.559	-12.1	-11.8	-11.5	-9.68
	Serial Hybrid	1.32	2.30	1.38	1.79	0.498	0.760	0.517	0.651	-4.66	1.69	-2.82	-0.30
	PACDNN	1.33	2.37	1.42	1.77	0.530	0.806	0.575	0.686	-3.67	4.11	-2.33	0.26
	Proposed	**1.63**	**2.47**	**1.60**	**1.93**	**0.587**	**0.816**	**0.619**	**0.708**	**-2.23**	**4.18**	**-0.91**	**1.45**
-3 dB	Unprocessed	1.06	1.66	1.16	1.31	0.493	0.704	0.524	0.600	-10.1	-9.80	-9.36	-8.26
	Serial Hybrid	1.52	2.49	1.66	1.88	0.579	0.793	0.619	0.698	-2.98	2.93	-0.90	0.48
	Unprocessed	**1.64**	**2.59**	1.71	1.95	**0.632**	0.832	0.659	0.726	**-1.37**	**5.63**	-0.20	1.73
	Proposed	1.54	2.54	**1.85**	**1.98**	0.617	**0.851**	**0.683**	**0.758**	-1.52	4.37	**0.57**	**1.76**
0 dB	Unprocessed	1.24	1.92	1.38	1.62	0.554	0.740	0.592	0.683	-7.76	-7.31	-7.01	-6.71
	Serial Hybrid	1.83	2.75	1.89	2.14	0.666	0.831	0.689	0.759	-0.54	4.55	0.72	2.15
	PACDNN	**1.90**	2.82	1.99	2.35	0.701	0.857	0.732	0.807	1.08	7.33	**1.85**	4.58
	Proposed	**1.90**	**2.86**	**2.08**	**2.41**	**0.727**	**0.865**	**0.742**	**0.813**	**1.25**	**7.62**	1.81	**4.87**
3 dB	Unprocessed	1.45	2.11	1.62	1.77	0.631	0.766	0.650	0.728	-5.19	-4.74	-4.64	-4.11
	Serial Hybrid	2.08	2.92	2.18	2.37	0.742	0.860	0.762	0.811	1.14	5.69	2.64	3.77
	PACDNN	**2.14**	2.86	2.23	2.38	0.774	0.877	0.784	**0.843**	2.98	8.41	3.71	5.77
	Proposed	**2.14**	**2.88**	**2.47**	**2.40**	**0.777**	**0.895**	**0.802**	0.832	**3.00**	**8.47**	**3.99**	**5.83**
6 dB	Unprocessed	1.71	2.35	1.85	1.97	0.712	0.798	0.714	0.766	-2.36	-1.83	-1.82	-1.16
	Serial Hybrid	2.32	3.14	2.43	2.66	0.802	0.882	0.811	0.859	3.64	6.88	4.36	5.39
	PACDNN	2.44	3.15	2.46	2.67	0.837	0.899	0.834	0.873	5.11	10.05	5.59	7.58
	Proposed	**2.45**	**3.21**	**2.58**	**2.87**	**0.839**	**0.918**	**0.852**	**0.884**	**5.58**	**10.29**	**6.26**	**7.78**

noisy and clean speech MDCT. The classifier is a very low complexity CNN made up of convolutional, global max-poling, and dense layers. A depth-wise separable convolution technique was adopted to reduce the CNN's number of parameters and FLOPs. Also, some methods were investigated to deal with the imbalanced dataset issue. In addition, the performance of the classifier was improved by adopting attention techniques between convolutional layers.

It is worth mentioning that the computations in the CNNs are performed in parallel, reducing the computational time and latency. Moreover, only one of the expert DNNs was active for each time, i.e., the computational burden is related to only a single branch at each time, although there are multiple branches in the model. The latency was also reduced to one-half compared to the methods in previous chapters. In addition, all the convolutions in this framework are causal in the sense that no future information is used in their calculations. Through extensive comparative experimental studies, it was shown that the proposed MBSE framework yields a significantly improved speech enhancement performance compared to existing DNN based methods while exhibiting a significantly reduced complexity and memory footprint.

Chapter 6

Conclusion and Future Work

6.1 Concluding Remarks

Deep learning as a primary tool to develop data driven information systems has led to revolutionary advances in speech enhancement. In this context, speech enhancement solved as a supervised learning problem, does not suffer from issues faced by traditional methods. This supervised learning problem comprises three key components: input features, learning machine, and training target. In this thesis, we proposed and investigated various deep learning structures and techniques to deal with limitations of these three components found in the literature. For the features, we used DNN for feature extraction instead of engineered features. For the learning machines, we explored combinations of different DNNs to take advantage of the contextual information of speech while using different techniques to keep the complexity of DNN very low. For the training targets, we investigated and proposed different categories of them so that both magnitude and phase are enhanced simultaneously. Throughout the whole thesis, we aimed at improving the speech enhancement performance while keeping the DNN complexity low so that the system is implementable on the edge.

In Chapter 3, we proposed a novel serial hybrid neural network that integrates a CNN and LSTM for speech enhancement based on PSM estimation. A new low-complexity fully-convolutional

CNN that facilitates learning, accelerates convergence, and reduces the number of model parameters was proposed to extract the most appropriate features of the input speech. This CNN was empowered by an attention technique to adaptively emphasize the valuable features extracted by CNN and suppress less important ones. In addition, an RNN was employed to take advantage of temporal dependencies of speech and accomplish the regression between the CNN-extracted features and the mask values. Different RNN variations were also evaluated and analyzed to optimize the network structure in terms of performance, computation time, memory footprint, and number of model parameters. Moreover, a grouping strategy was then adopted to reduce the number of RNN parameters. The proposed model was evaluated using different datasets and compared to some related DNN-based methods. Different training targets were also investigated to exploit the phase information alongside magnitude enhancement so as to achieve the best performance. Through extensive comparative experiments, we showed that the proposed model significantly outperforms some known neural network-based speech enhancement methods in the presence of highly non-stationary noises, while it exhibits a relatively small number of model parameters compared to some commonly employed DNN-based methods.

In Chapter 4, we proposed a very low-complexity composite model in which LSTM and CNN were carefully designed and integrated to extract a complementary set of features. LSTM and CNN performed independently and in parallel to speed up the computation, thereby addressing fundamental concerns from the perspective of real-time, low latency and low complexity speech enhancement applications. The new model, called PACDNN, involved two subtasks: magnitude processing with a spectral mask and phase reconstruction with PD, and exploited one DNN to estimate both training targets simultaneously. We investigated different types of masks and PDs as well as their possible combinations to select the best training targets for the DNN. Our analysis and experimental studies revealed that the proposed PACDNN model yields a significantly improved speech enhancement performance compared to several existing DNN based methods while exhibiting a significantly lower computational complexity and memory footprint.

In Chapter 5, we proposed a very low complexity framework called MBSE performing speech

enhancement in STMDCT domain. The MBSE framework consists of two main stages: classification and mapping. In the former stage, input speech is classified upon its utterance-level attributes, i.e., SNR level and gender. Four well-trained DNNs each specialized for a different specific and simple task, called expert DNNs, perform the mapping in the latter stage. Even though there are multi-branches in the model, only one of the low-complexity expert DNNs was active at each time step, i.e., the computational burden is related to only a single branch at each time step. Unlike the previous chapters, where the methods were masking-based, the expert DNNs introduced in the MBSE framework directly map noisy speech STMDCT to the clean one. We showed that the new structure comprising sub-DNNs can better handle different acoustic scenarios while having less overall computational complexity. The latency of MBSE framework was %55 less than the models in previous chapters. Through extensive experimental studies, it was shown that the MBSE framework not only gives a superior speech enhancement performance, but also has a lower complexity compared to some existing deep learning-based methods.

6.2 Scope for Further Work

In the previous chapters, several frameworks were introduced for monaural speech enhancement. The speech signal is usually corrupted by not only additive noise but also reverberation from surface reflections. However, we only studied and proposed methods for suppressing additive noise and did not investigate dereverberation. Besides, speech enhancement methods can be mainly categorized into single- and multi-channel, while monaural speech enhancement was only studied and practiced in this thesis. Last but not least, we only studied and investigated supervised methods while there are unsupervised methods based on say GANs, which could also yield good speech enhancement results. In the following, we introduce some topics for further research in the speech enhancement area.

6.2.1 Simultaneous Speech Dereverberation and Denoising

In an enclosed environment, the quality of the perceived sound is severely affected by reverberation which means the perceptual artifacts like echoes and coloration are added to the direct sound signal, thus reducing speech intelligibility [166]. Dereverberation has been a well-recognized challenge and would be more complicated when it is combined with the background noise. Reverberation corresponds to a convolution of the direct signal and a room impulse response (RIR), i.e.,

$$y(t) = h(t) * s(t) \tag{55}$$

where $y(t), h(t),$ and $s(t)$ denote reverberant speech, the RIR, and the clean anechoic speech, respectively, and $*$ indicates the convolution operation. In more sophisticated acoustic conditions, the reverberant speech could be further corrupted by ambient noise. As such, the key to speech enhancement in this situation is to simultaneously suppress the noise and estimate the RIR.

Different DNN-based methods have been proposed for speech dereverberation like spectral mapping on cochleagram [167], reverberation-time-aware model [168], etc. It is demonstrated that spectral mapping is more effective than masking for dereverberation while masking outperforms mapping for denoising [169]. Hence, to benefit from both mapping and masking methods to perform denoising and dereverberation simultaneously, we can study a two-stage network.

In the first stage, features of the noisy and reverberant speech can be passed to an LSTM network. As incorporation of delta and acceleration of the features improves dereverberation in a DNN-based structure [120], we can use the concatenation of the extracted features with their delta and acceleration as the input for the network. The LSTM can estimate the real and imaginary components of a complex IRM which will be multiplied by the noisy speech spectrogram for denoising. Then, the output which is the real and imaginary parts of the reverberant speech spectrogram can be passed as two channels to the second stage which is a CNN network. As CNN is a strong machine for spectral mapping, it will map both spectrogram components of the reverberant speech to the clean ones to accomplish speech dereverberation. The final output will be the real and imaginary

parts of the clean speech spectrogram which means the phase and magnitude are simultaneously enhanced. It is worth mentioning that these two stages can be first trained separately and then be jointly optimized.

The other promising idea is to extend our work by using RASTA-PLP features. RASTA is added to PLP in [170] so as to suppress slowly-changing factors in the noisy speech. RASTA filtering can help suppress the high-frequency components in the spectrum leading to background noise attenuation. It also suppresses the low-frequency components of the spectrum which mitigates reverberation [17].

6.2.2 Multi-Channel Speech Enhancement

An array of microphones providing multiple monaural recordings utilizes a different principle to enhance speech. Traditional approaches to source separation based on spatial information, say beamforming, boost the signal coming from a particular direction and suppress the signals coming from other directions as interferences. However, when the target speech source and noise are near each other, these methods will not work properly, and get worse in the presence of reverberation [7].

In [171], a speech separation approach based on spatial feature extraction is introduced where two spatial feature types are used for IBM estimation using a DNN. The first feature is interaural time difference (ITD) and the second one is interaural level difference (ILD) which measure the phase and level difference, respectively, between signals received by two microphones. The authors have shown interesting results in their paper. First, DNN can generalize well to an infinite number of unseen spatial configurations of the sound sources. Second, DNN can be generalized well to various unseen RIRs and reverberation times. Third, simultaneously using both monaural and spatial speech features can boost the performance of speech enhancement.

The other methodology for multi-channel speech enhancement is to integrate time-frequency masking and beamforming. To explain this methodology, minimum variance distortionless response (MVDR) as a representative beamformer will be first described.

According to [7], MVDR aims to mitigate the signal from non-target direction while maintains the energy from the target direction. The received signals by the microphone array can be written as follows,

$$Y(t,f) = C(t,f)s(t,f) + N(t,f) \tag{56}$$

where $Y(t,f)$ and $N(t,f)$ denote STFT spatial vectors of the noisy speech and noise. $C(f,t)s(t,f)$ is the image of source at each microphone, $C(f,t)$ is the channel between source and microphone, and $s(t,f)$ denotes STFT of the received speech source. MVDR leads to an optimization problem as follows,

$$w_{opt} = argmin\{w^H \Phi_n w\}, \quad subject\ to \quad w^H c = 1 \tag{57}$$

in which w is the weight vector of the beamformer that is to be determined to maximize the energy along the target direction while minimizing the rest. Note that H denotes Hermitian transpose, and Φ the spatial covariance matrix of the noise. Solving this optimization problem leads to the following optimized weight,

$$w_{opt} = \frac{\Phi_n^{-1}\ c}{c^H\ \Phi_n^{-1}\ c} \tag{58}$$

So, the enhanced speech will be obtained by,

$$\hat{s}(t) = w_{opt}^H\ y(t) \tag{59}$$

Then, with uncorrelated speech and noise, we have,

$$\Phi_x = \Phi_y - \Phi_n \tag{60}$$

Hence, the accurate estimation of the noise (Φ_n) plays a key role for the MVDR beamformer to perform properly.

In [172], the coefficients of an MVDR beamformer are estimated using an ideal ratio mask which is computed for monaural speech enhancement by an LSTM network. In other words,

LSTM enhancement is used to drive beamforming. The idea is to separately enhance each single channel signal using an LSTM network, which is trained from single-channel data and applied to each channel separately, to obtain several single-channel masks. Afterward, a robust beamformer is designed using the enhanced and the original signal. The noise spatial covariances for MVDR beamforming is estimated by applying a maximum operator to the array of the predicted masks.

As the conventional beamforming approaches rely on a noise estimate, just like the traditional speech enhancement methods, we believe a supervised model instead can be used to exploit the most appropriate information from data without any assumption, like no correlation between noise and target speech, which is indeed valid for stationary noises.

The whole process of speech enhancement and beamforming can be implemented as a fully learnable neural network and be jointly optimized. This way will make the method more robust to various unseen spatial configurations of the sound sources. In other words, instead of a dynamic beamformer requiring noise estimation, a DNN with fixed parameters can be utilized to exploit the most appropriate information of the spatial configuration of the sound sources.

The spatial and monaural features of different channels can be concatenated and normalized, and then fed to a neural network to simultaneously benefit from both feature types. Thus, this network learns how to exploit the most useful information from all the channels and then transform them into one single channel. It is worth mentioning that the optimization of the first stage can be initialized using some assumptions not randomly, and then the network will decide on the rest of the optimization process within training proceedure. Furthermore, this method is real-time and does not have a long delay unlike beamformers that perform on the information of one complete utterance which leads to a long delay.

6.2.3 Semantic Image Inpainting

There is a challenging task called semantic image inpainting which refers to recovery of large missing regions of an image based on the available visual data. Raymond et al. [173] introduced a novel method using deep generative models to generate the missing content by conditioning on the

Figure 55: Original and restored image.

available data as shown in Fig. 55. First, they trained a deep generative model. Then, their model searches for a corrupted image encoding which is closest to the image in the latent space. Next, the image will be reconstructed with the generator using this encoding.

We can use the same idea for speech enhancement. Considering the speech spectrogram as an image, some T-F units in the spectrogram are missing, i.e. noise dominates in these T-F units. So, we can use this model to reconstruct the clean speech given the noisy spectrogram.

Bibliography

[1] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal processing letters*, vol. 21, no. 1, pp. 65–68, 2013.

[2] H. Zhang, X. Zhang, and G. Gao, "Multi-target ensemble learning for monaural speech separation.," in *INTERSPEECH*, pp. 1958–1962, 2017.

[3] Y. Xu, J. Du, Z. Huang, L.-R. Dai, and C.-H. Lee, "Multi-objective learning and mask-based post-processing for deep neural network based speech enhancement," *arXiv preprint arXiv:1703.07172*, 2017.

[4] L. Sun, J. Du, L.-R. Dai, and C.-H. Lee, "Multiple-target deep learning for lstm-rnn based speech enhancement," in *2017 Hands-free Speech Communications and Microphone Arrays (HSCMA)*, pp. 136–140, IEEE, 2017.

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

[6] D. S. Williamson, Y. Wang, and D. Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 24, no. 3, pp. 483–492, 2015.

[7] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.

[8] H. Yu, *DNN-Assisted Speech Enhancement Approaches Incorporating Phase Information*. PhD thesis, Concordia University, 2021.

[9] S. Pascual, A. Bonafonte, and J. Serra, "Segan: Speech enhancement generative adversarial network," *arXiv preprint arXiv:1703.09452*, 2017.

[10] D. Kobran and d. Banys, "Activation functions." `https://docs.paperspace.com/machine-learning/wiki/activation-function`, 2020.

[11] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[12] Z. Ouyang, H. Yu, W.-P. Zhu, and B. Champagne, "A fully convolutional neural network for complex spectrogram processing in speech enhancement," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5756–5760, IEEE, 2019.

[13] C. Yu, R. E. Zezario, S.-S. Wang, J. Sherman, Y.-Y. Hsieh, X. Lu, H.-M. Wang, and Y. Tsao, "Speech enhancement based on denoising autoencoder with multi-branched encoders," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2756–2769, 2020.

[14] A. Sá, A. Almeida, B. Rocha, M. Mota, J. Souza, and L. Dentel, "Lightning forecast using data mining techniques on hourly evolution of the convective available potential energy," in *Brazilian Congress on Computational Intelligence, Fortaleza, November*, pp. 8–11, 2011.

[15] B. Stecanella, "An Introduction to Support Vector Machines (SVM)." `https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/`, 2017.

[16] Wikipedia, "F-score." `https://en.wikipedia.org/wiki/F-score`, 2021.

[17] M. Delfarah and D. Wang, "Features for masking-based monaural speech separation in reverberant conditions," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 1085–1094, 2017.

[18] R. Y. M. Li, H. Li, C. Mak, and T. Tang, "Sustainable smart home and home automation: Big data analytics approach," *Int. Journal of Smart Home*, vol. 10, no. 8, pp. 177–187, 2016.

[19] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Tran. on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, 2014.

[20] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[21] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," in *ICASSP'79. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 208–211, 1979.

[22] W. M. Kushner, V. Goncharoff, C. Wu, V. Nguyen, and J. N. Damoulakis, "The effects of subtractive-type speech enhancement/noise reduction algorithms on parameter estimation for improved recognition and coding in high noise environments," in *International Conference on Acoustics, Speech, and Signal Processing,*, pp. 211–214, 1989.

[23] L. Singh and S. Sridharan, "Speech enhancement using critical band spectral subtraction," in *Fifth International Conference on Spoken Language Processing*, 1998.

[24] S. Kamath, P. Loizou, *et al.*, "A multi-band spectral subtraction method for enhancing speech corrupted by colored noise.," in *ICASSP*, vol. 4, pp. 44164–44164, Citeseer, 2002.

[25] S. S. Haykin, *Adaptive filter theory*. Pearson Education India, 2008.

[26] J. S. Lim and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech," *Proceedings of the IEEE*, vol. 67, no. 12, pp. 1586–1604, 1979.

[27] Y. Hu and P. C. Loizou, "A perceptually motivated approach for speech enhancement," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 5, pp. 457–465, 2003.

[28] S. Srinivasan, J. Samuelsson, and W. B. Kleijn, "Codebook driven short-term predictor parameter estimation for speech enhancement," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 163–176, 2005.

[29] M. A. Abd El-Fattah, M. I. Dessouky, A. M. Abbas, S. M. Diab, E.-S. M. El-Rabaie, W. Al-Nuaimy, S. A. Alshebeili, and F. E. Abd El-samie, "Speech enhancement with an adaptive wiener filter," *International Journal of Speech Technology*, vol. 17, no. 1, pp. 53–64, 2014.

[30] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[31] E. Plourde and B. Champagne, "Auditory-based spectral amplitude estimators for speech enhancement," *IEEE transactions on audio, speech, and language processing*, vol. 16, no. 8, pp. 1614–1623, 2008.

[32] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," 2017.

[34] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Convolutional neural network committees for handwritten character classification," in *2011 International Conference on Document Analysis and Recognition*, pp. 1135–1139, IEEE, 2011.

[35] Y. Liu and J. Zhang, "Deep learning in machine translation," in *Deep Learning in Natural Language Processing*, pp. 147–183, Springer, 2018.

[36] L. Deng and J. C. Platt, "Ensemble deep learning for speech recognition," in *Fifteenth annual conference of the international speech communication association*, 2014.

[37] M. Sundermeyer, H. Ney, and R. Schlüter, "From feedforward to recurrent lstm neural networks for language modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 517–529, 2015.

[38] J. Agnew and J. M. Thornton, "Just noticeable and objectionable group delays in digital hearing aids," *Journal of the American Academy of Audiology*, vol. 11, no. 6, pp. 330–336, 2000.

[39] Z. Ouyang, "Single-channel speech enhancement based on deep neural network," Master's thesis, Concordia University, 2020.

[40] Y. Wang and D. Wang, "Boosting classification based speech separation using temporal dynamics," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[41] Y. Wang and D. Wang, "Cocktail party processing via structured prediction," *Advances in Neural Information Processing Systems*, vol. 25, pp. 224–232, 2012.

[42] Y. Wang and D. Wang, "Towards scaling up classification-based speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1381–1390, 2013.

[43] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[44] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder.," in *Interspeech*, vol. 2013, pp. 436–440, 2013.

[45] F. Weninger, J. R. Hershey, J. Le Roux, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 577–581, IEEE, 2014.

[46] F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. Le Roux, J. R. Hershey, and B. Schuller, "Speech enhancement with lstm recurrent neural networks and its application to noise-robust asr," in *International conference on latent variable analysis and signal separation*, pp. 91–99, Springer, 2015.

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

[48] E. M. Grais and M. D. Plumbley, "Single channel audio source separation using convolutional denoising autoencoders," in *2017 IEEE global conference on signal and information processing (GlobalSIP)*, pp. 1265–1269, IEEE, 2017.

[49] S.-W. Fu, Y. Tsao, X. Lu, and H. Kawai, "Raw waveform-based speech enhancement by fully convolutional networks," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 006–012, IEEE, 2017.

[50] A. Pandey and D. Wang, "A new framework for supervised speech enhancement in the time domain.," in *Interspeech*, pp. 1136–1140, 2018.

[51] A. Sugiyama and R. Miyahara, "Phase randomization-a new paradigm for single-channel signal enhancement," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 7487–7491, IEEE, 2013.

[52] D. Gunawan and D. Sen, "Iterative phase estimation for the synthesis of separated sources from single-channel mixtures," *IEEE Signal Processing Letters*, vol. 17, no. 5, pp. 421–424, 2010.

[53] P. Mowlaee, R. Saeidi, and R. Martin, "Phase estimation for signal reconstruction in single-channel source separation," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[54] M. Krawczyk and T. Gerkmann, "Stft phase reconstruction in voiced speech for an improved single-channel speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1931–1940, 2014.

[55] J. Chen and D. Wang, "Long short-term memory for speaker generalization in supervised speech separation," *The Journal of the Acoustical Society of America*, vol. 141, no. 6, pp. 4705–4714, 2017.

[56] T. Gao, J. Du, L.-R. Dai, and C.-H. Lee, "Snr-based progressive learning of deep neural network for speech enhancement.," in *INTERSPEECH*, pp. 3713–3717, 2016.

[57] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "Dynamic noise aware training for speech enhancement based on deep neural networks," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[58] J. Du, Y.-H. Tu, L. Sun, F. Ma, H.-K. Wang, J. Pan, C. Liu, J.-D. Chen, and C.-H. Lee, "The ustc-iflytek system for chime-4 challenge," *Proc. CHiME*, vol. 4, pp. 36–38, 2016.

[59] M. Kolbæk, Z.-H. Tan, and J. Jensen, "Speech intelligibility potential of general and specialized deep neural network based speech enhancement systems," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 153–167, 2016.

[60] L. S. Vailshery, "Smart speaker market revenue worldwide from 2014 to 2025." `https://www.statista.com/statistics/1022823/worldwide-smart-speaker-market-revenue/`, 2021.

[61] M. Parchami, *New Approaches for Speech Enhancement in the Short-Time Fourier Transform Domain*. PhD thesis, Concordia University, 2016.

[62] E. Sejdić, I. Djurović, and J. Jiang, "Time-frequency feature representation using energy concentration: An overview of recent advances," *Digital signal processing*, vol. 19, no. 1, pp. 153–183, 2009.

[63] Y. Wang, K. Han, and D. Wang, "Exploring monaural features for classification-based speech segregation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 2, pp. 270–279, 2012.

[64] K. Han and D. Wang, "A classification based approach to speech segregation," *The Journal of the Acoustical Society of America*, vol. 132, no. 5, pp. 3475–3483, 2012.

[65] C. Hummersone, T. Stokes, and T. Brookes, "On the ideal ratio mask as the goal of computational auditory scene analysis," in *Blind source separation*, pp. 349–368, Springer, 2014.

[66] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 22, no. 12, pp. 1849–1858, 2014.

[67] L. Lightburn and M. Brookes, "Sobm-a binary mask for noisy speech that optimises an objective intelligibility metric," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5078–5082, IEEE, 2015.

[68] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 708–712, IEEE, 2015.

[69] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1562–1566, IEEE, 2014.

[70] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[71] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

[72] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[73] D. Yu and L. Deng, *Automatic Speech Recognition.* Springer, 2016.

[74] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*, pp. 1310–1318, PMLR, 2013.

[75] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[76] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[77] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[78] Y. Bengio, Y. LeCun, *et al.*, "Scaling learning algorithms towards ai," *Large-scale kernel machines*, vol. 34, no. 5, pp. 1–41, 2007.

[79] C. Donahue, B. Li, and R. Prabhavalkar, "Exploring speech enhancement with generative adversarial networks for robust speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5024–5028, IEEE, 2018.

[80] N. Khan, "Activation functions in deep learning." `https://medium.com/@najeebnik21/activation-function-in-deep-learning-587e83d5a681`, 2017.

[81] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *NASA STI/Recon technical report n*, vol. 93, p. 27403, 1993.

[82] E. Rothauser, "Ieee recommended practice for speech quality measurements," *IEEE Trans. on Audio and Electroacoustics*, vol. 17, pp. 225–246, 1969.

[83] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[84] A. Rousseau, P. Deléglise, and Y. Esteve, "Ted-lium: an automatic speech recognition dedicated corpus.," in *LREC*, pp. 125–129, 2012.

[85] A. Rousseau, P. Deléglise, Y. Esteve, *et al.*, "Enhancing the ted-lium corpus with selected data for language modeling and more ted talks.," in *LREC*, pp. 3935–3939, 2014.

[86] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Estève, "Ted-lium 3: twice as much data and corpus repartition for experiments on speaker adaptation," in *International Conference on Speech and Computer*, pp. 198–208, Springer, 2018.

[87] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5206–5210, IEEE, 2015.

[88] H. McGuire, "LibriVox." `https://librivox.org/`, 2005.

[89] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, "Mls: A large-scale multilingual dataset for speech research," *arXiv preprint arXiv:2012.03411*, 2020.

[90] H.-G. Hirsch and D. Pearce, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *ASR2000-Automatic speech recognition: challenges for the new Millenium ISCA tutorial and research workshop (ITRW)*, 2000.

[91] A. Varga and H. J. 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 communication*, vol. 12, no. 3, pp. 247–251, 1993.

[92] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The fifth'chime'speech separation and recognition challenge: dataset, task and baselines," *arXiv preprint arXiv:1803.10609*, 2018.

[93] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, vol. 2, pp. 749–752, IEEE, 2001.

[94] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *2010 IEEE international conference on acoustics, speech and signal processing*, pp. 4214–4217, IEEE, 2010.

[95] J. H. Hansen and B. L. Pellom, "An effective quality evaluation protocol for speech enhancement algorithms," in *Fifth international conference on spoken language processing*, 1998.

[96] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, 2014.

[97] Y. Wang, J. Du, L.-R. Dai, and C.-H. Lee, "A maximum likelihood approach to deep neural network based nonlinear spectral mapping for single-channel speech separation.," in *INTERSPEECH*, pp. 1178–1182, 2017.

[98] M. Hasannezhad, Z. Ouyang, W.-P. Zhu, and B. Champagne, "Speech separation using a composite model for complex mask estimation," in *2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 578–581, IEEE, 2020.

[99] A. Al-Dulaimi, S. Zabihi, A. Asif, and A. Mohammed, "Nblstm: Noisy and hybrid convolutional neural network and blstm-based deep architecture for remaining useful life estimation," *Journal of Computing and Information Science in Engineering*, vol. 20, no. 2, 2020.

[100] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning*, pp. 173–182, PMLR, 2016.

[101] J. Guo, N. Xu, L.-J. Li, and A. Alwan, "Attention based cldnns for short-duration acoustic scene classification.," in *Interspeech*, pp. 469–473, 2017.

[102] D. Yin, C. Luo, Z. Xiong, and W. Zeng, "Phasen: A phase-and-harmonics-aware speech enhancement network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 9458–9465, 2020.

[103] H. Zhang and J. Ma, "Hartley spectral pooling for deep learning," *arXiv preprint arXiv:1810.04028*, 2018.

[104] S.-W. Fu, T.-y. Hu, Y. Tsao, and X. Lu, "Complex spectrogram enhancement by convolutional neural network with multi-metrics learning," in *2017 IEEE 27th international workshop on machine learning for signal processing (MLSP)*, pp. 1–6, IEEE, 2017.

[105] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[106] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[107] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2018.

[108] A. G. Roy, N. Navab, and C. Wachinger, "Concurrent spatial and channel 'squeeze & excitation' in fully convolutional networks," in *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*, pp. 421–429, Springer, 2018.

[109] K.-L. Du and M. Swamy, "Recurrent neural networks," in *Neural networks and statistical learning*, pp. 351–371, Springer, 2019.

[110] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[111] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (gru) neural networks," in *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, pp. 1597–1600, IEEE, 2017.

[112] F. Gao, L. Wu, L. Zhao, T. Qin, X. Cheng, and T.-Y. Liu, "Efficient Sequence Learning with Group Recurrent Networks," in *Proc. of the 2018 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 799–808, 2018.

[113] K. Tan and D. Wang, "Learning Complex Spectral Mapping with Gated Convolutional Recurrent Networks for Monaural Speech Enhancement," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 28, pp. 380–390, 11 2019.

[114] K. Tan and D. Wang, "Complex spectral mapping with a convolutional recurrent network for monaural speech enhancement," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6865–6869, IEEE, 2019.

[115] S. Xia, H. Li, and X. Zhang, "Using optimal ratio mask as training target for supervised speech separation," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 163–166, IEEE, 2017.

[116] M. Hasannezhad, Z. Ouyang, W.-P. Zhu, and B. Champagne, "An integrated cnn-gru framework for complex ratio mask estimation in speech enhancement," in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 764–768, IEEE, 2020.

[117] "Premium beat." `www.premiumbeat.com`.

[118] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[119] J. Chen, Y. Wang, and D. Wang, "A feature study for classification-based speech separation at low signal-to-noise ratios," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1993–2002, 2014.

[120] X. Xiao, S. Zhao, D. H. H. Nguyen, X. Zhong, D. L. Jones, E. S. Chng, and H. Li, "Speech dereverberation for enhancement and recognition using dynamic features constrained deep neural networks and feature adaptation," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, pp. 1–18, 2016.

[121] D. S. Williamson and D. Wang, "Time-frequency masking in the complex domain for speech dereverberation and denoising," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 25, no. 7, pp. 1492–1501, 2017.

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

[123] H. Zhao, S. Zarar, I. Tashev, and C.-H. Lee, "Convolutional-recurrent neural networks for speech enhancement," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2401–2405, April 2018.

[124] T.-A. Hsieh, H.-M. Wang, X. Lu, and Y. Tsao, "Wavecrn: An efficient convolutional recurrent neural network for end-to-end speech enhancement," *arXiv preprint arXiv:2004.04098*, 2020.

[125] M. Strake, B. Defraene, K. Fluyt, W. Tirry, and T. Fingscheidt, "Fully convolutional recurrent networks for speech enhancement," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6674–6678, May 2020.

[126] M. P. Shifas, S. Claudio, Y. Stylianou, *et al.*, "A fully recurrent feature extraction for single channel speech enhancement," *arXiv preprint arXiv:2006.05233*, 2020.

[127] A. Pandey and D. Wang, "Learning complex spectral mapping for speech enhancement with improved cross-corpus generalization," *Proc. Interspeech 2020*, pp. 4511–4515, 2020.

[128] X. Wang and C. Bao, "Mask estimation incorporating phase-sensitive information for speech enhancement," *Applied Acoustics*, vol. 156, pp. 101–112, 2019.

[129] F. Mayer, D. S. Williamson, P. Mowlaee, and D. Wang, "Impact of phase estimation on single-channel speech separation based on time-frequency masking," *The Journal of the Acoustical Society of America*, vol. 141, no. 6, pp. 4668–4679, 2017.

[130] P. Mowlaee and R. Saeidi, "Time-frequency constraints for phase estimation in single-channel speech enhancement," in *Int. Workshop on Acoustic Signal Enhancement (IWAENC)*, pp. 337–341, IEEE, 2014.

[131] N. Zheng and X.-L. Zhang, "Phase-aware speech enhancement based on deep neural networks," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 27, no. 1, pp. 63–76, 2018.

[132] S. Liang, W. Liu, W. Jiang, and W. Xue, "The optimal ratio time-frequency mask for speech separation in terms of the signal-to-noise ratio," *The J. of the Acoustical Society of America*, vol. 134, no. 5, pp. EL452–EL458, 2013.

[133] A. P. Stark and K. K. Paliwal, "Speech analysis using instantaneous frequency deviation," in *INTERSPEECH*, 2008.

[134] R. M. Hegde, H. A. Murthy, and V. R. R. Gadde, "Significance of the modified group delay feature in speech recognition," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 190–202, 2007.

[135] V. K. Prasad, T. Nagarajan, and H. A. Murthy, "Automatic segmentation of continuous speech using minimum phase group delay functions," *speech communication*, vol. 42, no. 3-4, pp. 429–446, 2004.

[136] Q. Li, F. Gao, H. Guan, and K. Ma, "Real-time monaural speech enhancement with short-time discrete cosine transform," *arXiv preprint arXiv:2102.04629*, 2021.

[137] J. Princen, A. Johnson, and A. Bradley, "Subband/transform coding using filter bank designs based on time domain aliasing cancellation," in *ICASSP'87. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 12, pp. 2161–2164, IEEE, 1987.

[138] Y. Koizumi, N. Harada, Y. Haneda, Y. Hioka, and K. Kobayashi, "End-to-end sound source enhancement using deep neural network in the modified discrete cosine transform domain," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 706–710, IEEE, 2018.

[139] M. Kim, "Collaborative deep learning for speech enhancement: A run-time model selection method using autoencoders," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 76–80, IEEE, 2017.

[140] S. E. Chazan, J. Goldberger, and S. Gannot, "Deep recurrent mixture of experts for speech enhancement," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 359–363, IEEE, 2017.

[141] A. Gisbrecht, B. Mokbel, and B. Hammer, "Linear basis-function t-sne for fast nonlinear dimensionality reduction," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2012.

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

[143] T. F. Chan, G. H. Golub, and R. J. LeVeque, "Updating formulae and a pairwise algorithm for computing sample variances," in *COMPSTAT 1982 5th Symposium held at Toulouse 1982*, pp. 30–41, Springer, 1982.

[144] F. Osisanwo, J. Akinsola, O. Awodele, J. Hinmikaiye, O. Olakanmi, and J. Akinjobi, "Supervised machine learning algorithms: classification and comparison," *International Journal of Computer Trends and Technology (IJCTT)*, vol. 48, no. 3, pp. 128–138, 2017.

[145] E. Fix, *Discriminatory analysis: nonparametric discrimination, consistency properties*, vol. 1. USAF school of Aviation Medicine, 1985.

[146] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and regression trees. belmont, ca: Wadsworth," *International Group*, vol. 432, pp. 151–166, 1984.

[147] F. Kemp, "Applied multiple regression/correlation analysis for the behavioral sciences," 2003.

[148] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *Journal of artificial intelligence research*, vol. 11, pp. 169–198, 1999.

[149] Y. Gavrilova and R. Stryungis, "Machine Learning Algorithm Classification for Beginners." `https://serokell.io/blog/machine-learning-algorithm-classification-overview`, 2020.

[150] M. Sidana, "Intro to types of classification algorithms in Machine Learning." `https://medium.com/sifium/machine-learning-types-of-classification-9497bd4f2e14`, 2017.

[151] J. Brownlee, "Linear Discriminant Analysis for Machine Learning." `https://machinelearningmastery.com/linear-discriminant-analysis-for-machine-learning/`, 2016.

[152] S. Raschka, "Linear Discriminant Analysis – Bit by Bit." `https://sebastianraschka.com/Articles/2014_python_lda.html#introduction`, 2014.

[153] scikit-learn developers, "Sklearn: Base Classes and Utility Functions." `https://scikit-learn.org/stable/modules/classes.html#module-sklearn.naive_bayes`, 2020.

[154] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[155] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[156] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[157] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[158] H. Phan, L. Hertel, M. Maass, P. Koch, R. Mazur, and A. Mertins, "Improved audio scene classification based on label-tree embeddings and convolutional neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1278–1290, 2017.

[159] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, *et al.*, "Cnn architectures for large-scale audio classification," in *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pp. 131–135, IEEE, 2017.

[160] H. Sadreazami, M. Bolic, and S. Rajan, "Fall detection using standoff radar-based sensing and deep convolutional neural network," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 1, pp. 197–201, 2019.

[161] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[162] K. Ting, "Confusion matrix, encyclopedia of machine learning and data mining," 2017.

[163] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

[164] CenterForAI, "Depthwise Separable Convolution - A FASTER CONVOLUTION!." `https://www.youtube.com/watch?v=T7o3xvJLuHk&t=480s`, 2018.

[165] P. Branco, L. Torgo, and R. Ribeiro, "A survey of predictive modelling under imbalanced distributions," *arXiv preprint arXiv:1505.01658*, 2015.

[166] A. C. Neuman, M. Wroblewski, J. Hajicek, and A. Rubinstein, "Combined effects of noise and reverberation on speech recognition performance of normal-hearing children and adults," *Ear and hearing*, vol. 31, no. 3, pp. 336–344, 2010.

[167] K. Han, Y. Wang, and D. Wang, "Learning spectral mapping for speech dereverberation," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4628–4632, IEEE, 2014.

[168] B. Wu, K. Li, M. Yang, and C.-H. Lee, "A reverberation-time-aware approach to speech dereverberation based on deep neural networks," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 25, no. 1, pp. 102–111, 2016.

[169] Y. Zhao, Z.-Q. Wang, and D. Wang, "A two-stage algorithm for noisy and reverberant speech enhancement," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5580–5584, IEEE, 2017.

[170] H. Hermansky and N. Morgan, "Rasta processing of speech," *IEEE transactions on speech and audio processing*, vol. 2, no. 4, pp. 578–589, 1994.

[171] Y. Jiang, D. Wang, R. Liu, and Z. Feng, "Binaural classification for reverberant speech segregation using deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 2112–2121, 2014.

[172] H. Erdogan, J. R. Hershey, S. Watanabe, M. I. Mandel, and J. Le Roux, "Improved mvdr beamforming using single-channel mask prediction networks.," in *Interspeech*, pp. 1981–1985, 2016.

[173] R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5485–5493, 2017.

Appendix A

Publications from the Thesis Research

Following is a list of peer-reviewed journal and conference papers based on this thesis that have been published during this research work.

[1] **M. Hasannezhad**, Z. Ouyang, W.-P. Zhu, and B. Champagne, "PACDNN: A Phase-Aware Composite Deep Neural Network for Speech Enhancement", Speech Communication, revised and under further review, 2021.

[2] **M. Hasannezhad**, Z. Ouyang, W.-P. Zhu, and B. Champagne, "Speech Enhancement with Phase Sensitive Mask Estimation using a Novel Hybrid Neural Network", IEEE Open Journal of Signal Processing, vol. 2, pp. 136-150, 2021.

[3] **M. Hasannezhad**, W.-P. Zhu, and B. Champagne, "A Novel Low-Complexity Attention-Driven Composite Model for Speech Enhancement," IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1-5, May 2021.

[4] **M. Hasannezhad**, Z. Ouyang, W.-P. Zhu, and B. Champagne, "A Novel Integrated CNN-GRU Framework for Complex Ratio Mask Estimation in Speech Separation," 12th IEEE Asia-Pacific Signal and Information Processing Association (APSIPA), pp. 764-768, Dec. 2020.

[5] **M. Hasannezhad**, Z. Ouyang, W.-P. Zhu, and B. Champagne, "Speech Separation Using a Composite Model for Complex Mask Estimation," 64th IEEE Midwest Symposium on Circuits and Systems (MWSCS), pp. 578-581, Aug. 2020.

[6] **M. Hasannezhad**, W.-P. Zhu, and B. Champagne, "MBSE: A Multi-Mode Mapping-Based Speech Enhancement in Discrete Cosine Transform Domain", to be submitted, 2021.