Single channel speech enhancement based on U-Net

Bengbeng He

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Electrical and Computer Engineering) at

Concordia University

Montréal, Québec, Canada

July 2022

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Bengbeng He

Entitled: Single channel speech enhancement based on U-Net

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

Master of Applied Science (Electrical and Computer Engineering)

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

Signed by the final examining committee:

_____ Chair
Dr. M. Omair Ahmad

_____ Examiner
Dr. M. Omair Ahmad

_____ Examiner
Dr. Youmin Zhang (MIAE)

_____ Thesis Supervisor(s)
Dr. Wei-Ping Zhu

_____ Thesis Supervisor(s)

_____ Thesis Supervisor(s)

Approved by _____
Dr. Yousef Shayan, Chair
Department of Electrical and Computer Engineering

_____
Dr. Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Date _____

Abstract

Single channel speech enhancement based on U-Net

Bengbeng He

Speech enhancement has found many applications in various fields involving speech processing. It aims to remove the background noise from the acquired speech signals to efficiently improve the speech intelligibility and quality. The development of deep learning approaches in recent years has significantly promoted the advancement of speech enhancement by treating it as an estimation problem with or without supervision. Many existing neural networks are based on U-Net structure, where the encoder of U-Net transforms the input speech into compressed features via removing the noise information and the decoder of U-Net reconstructs the enhanced speech from the speech features with a symmetric structure. However, the contextual information of speech sequences cannot be fully captured due to the intrinsically local operation of commonly used convolution in U-Net, which is extremely crucial for improving the performance of speech enhancement. To improve the capability of U-Net in extracting the long-term dependency of speech sequences, this thesis investigates different attention-based U-Nets to capture the abundant contextual information of long-range speech signals.

In the first contribution of this thesis, a dual-branch attention-assisted U-Net is proposed for single-channel speech enhancement, which consists of a dilated-dense encoder-decoder structure and a dual-branch attention mechanism in between. In the encoder, the high-level speech features are obtained by adopting multiple groups of a dilated-dense block and a down-sampling layer, where the densely dilated convolutions are employed to enlarge the receptive field and aggregate previous features. Next, the dual-branch attention utilizes the spatial-wise attention and channel-wise attention to parallelly extract spatial and channel information of speech features, which are then averaged to form the contextual feature representation. The decoder, as a symmetric structure of encoder, is adopted to transform the features back to denoised speech via multiple pairs of a dilated-dense block and an up-sampling layer, where the skip connection from encoder layers is used to boost the feature reconstruction. By comparing with other U-Net based methods, our

proposed dual-branch attention-assisted U-Net achieves a comparable performance of evaluation metrics but with fairly low trainable parameters.

To further improve the performance of proposed attention-based U-Net, in the second contribution of this thesis, we incorporate the multi-head attention (MHA) mechanism into U-Net for extracting the features from different representation subspaces. First, we propose a two-stage MHA block to replace the dual-branch attention block in the U-Net structure proposed before, where the MHA block employs tandemly connected sample MHA and frame MHA to successively extract sample-level features of each individual frame and frame-level features among different frames, respectively, leading to better contextual speech features. However, the convolutional encoder-decoder used in the proposed U-Net still constrains the potentials of U-Net model to extract long-range dependency of speech sequences because of the local operation performed in convolution. To overcome this drawback, we further replace the convolution-based encoder and decoder layers by proposed two-stage MHA blocks to extract long-range relationship of speech sequences in the encoder-decoder level. Experimental results on a benchmark dataset shows that our two proposed MHA based U-Net models achieve a competitive performance among existing methods in all evaluation metrics while exhibiting a much lighter model complexity than other state-of-art networks.

Related Publications

The work of this thesis led to the following research publications:

1. Kai Wang, **Bengbeng He**, and Wei-Ping Zhu, "CAUNet: Context-aware U-Net for speech enhancement in time domain," in 2021 *IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2021, pp. 1–5.

2. **Bengbeng He**, Kai Wang, and Wei-Ping Zhu, "A novel multi-head attention U-Net for speech enhancement," submitted to 2022 *Asia Pacific Signal and Information Processing Association (APSIPA)*.

3. **Bengbeng He** and Wei-Ping Zhu, "DBUANet: Dual-branch attention U-Net for time-domain speech enhancement", to be submitted in 2023 *IEEE International Symposium on Circuits and Systems (ISCAS)*.

Acknowledgements

First of all, I would like to express greatest thanks to my supervisor, Prof. Wei-Ping Zhu, who provides the valuable research environment of our speech enhancement group and patient guidance as well as a number of inspirations on my way to dive into the research work. And he is always there to listen to anyone in need and offer a support.

Moreover, I would like to give sincere thanks to my intelligent and active research partners, Mr. Kai Wang, Mr. Zhiheng Ouyang, Dr. Hongjiang Yu and Dr. Mojtaba Hasannezhad, the people who also work on the speech enhancement group and always give me useful advice and great encouragements along with my research works. It is a wonderful experience for me to work with them during these couple of years.

Finally, I have been very appreciative of supports from my parents all the time, which builds my original power from heart to face and overcome all obstacles in my life.

Contents

List of Figures

List of Tables

List of Abbreviations

ACB	Attention-based Convolutional Block
ARCN	Attention-based Redundant Convolutional Network
BN	Batch Normalization
Bi-GRU	Bi-directional GRU
CMHAB	Cascaded MHA Block
CNN	Convolutional Neural Network
cIRM	Complex Ideal Ratio Mask
CRN	Convolutional Recurrent Network
DBAUNet	Dual-Branch Attention U-Net
DCN	Dense Convolutional Network
DW	DepthWise
DBAB	Dual-Branch Attention Block
FC	Fully-Connected
FC-NN	Fully-Connected Neural Network
GRU	Gated Recurrent Unit
GN	Group Normalization
MSE	Mean Square Error
MHAUNet	Multi-Head Attention U-Net
MHAB	Multi-Head Attention Block
NLP	Natural Language Processing
ORM	Optimal Ratio Mask
PESQ	Perceptual Evaluation of Speech Quality
PReLU	Parametric ReLU
PW	PointWise
PSM	Phase Sensitive Mask
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
SE	Speech Enhancement

STFT	Short-Term Fourier Transform
SNR	Signal-to-Noise Ratio
STOI	Short-Time Objective Intelligibility
SSNR	Segmental SNR
SDR	Signal Distortion Rate
TCN	Temporal Convolutional Network
TCNN	Temporal Convolutional Neural Network
TSMHAB	Two-Stage Multi-Head Attention Block
T-F	Time-Frequency
TIMIT	Texas Instruments and Massachusetts Institute of Technology

Chapter 1
Introduction

Speech enhancement (SE) can improve speech quality by supressing unexpected noises from noisy environment. It has many important applications in various areas relating to acoustic signal processing, including communications, hearing aids and automatic speech recognition. Researchers have investigated a large number of conventional approaches for speech enhancement including Wiener filter [1], statistical models [2], spectral subtraction [3] and noise-subspace algorithms [4]. In recent years, deep learning algorithms have been broadly explored in speech enhancement as powerful data-driven tools, making an impressive achievement on speech processing.

Deep learning-based SE systems are investigated to learn a complex mapping function from noisy speech to clean one by using neural networks like fully-connected neural network (FCNN) [5-6], convolutional neural networks (CNNs) [7-8], and recurrent neural networks (RNNs) [9-10]. Generally speaking, current deep learning-based SE systems can be divided into time-domain methods and time-frequency (T-F) domain methods. Time-domain methods operate on one-dimensional noisy speech waveform to directly estimate the enhanced speech waveform, which is inspired from sequence models for natural language processing (NLP). T-F domain methods process two-dimensional image-like spectrograms obtained from short-term Fourier Transform (STFT) that contain both temporal and frequency information. Moreover, the attention mechanism is widely introduced to improve the performance of existing SE systems [11-12], which can efficiently capture the long-range global dependency of speech sequences. Compared with CNNs and RNNs, the attention-based system can effectively extract the contextual information.

In this thesis, we explore two attention-based U-Net neural networks for single-channel speech enhancement in the time domain. The U-Net structure adopts the encoder to extract the compressed speech features and then employs a decoder that is symmetric to the encoder to reconstruct the target enhanced speech, where the noise component in the noisy speech is eliminated or suppressed. To capture the long-term dependency of speech signals, we propose attention-based modules as a

part of U-Net architecture to build a contextual sequence modeling for further promoting the performance of speech enhancement. In addition, we attempt to design lightweight SE systems while having promising denoising performance for potential mobile applications. In the following, we will introduce some popular deep neural networks and their application for speech enhancement.

1.1 Deep-learning models for speech enhancement

1.1.1 Brief review of speech enhancement problem

Deep learning based SE system aims to build a mapping function between noisy speech and clean speech, which commonly takes the raw speech waveform or speech spectrogram as the input. The speech waveform is one-dimensional sequential data, where the length of sequence depends on the sampling rate and each sample point represents the strength of voice. The noisy speech waveform can be modelled as:

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

where $y(t)$, $x(t)$, $n(t)$ denote the sequence of noisy speech, clean speech and noise speech at time step t, respectively.

Different from speech waveform, the speech spectrogram has more obvious harmonics of speech signal, which is very useful to distinguish between clean speech and noisy speech. The representation of noisy spectrogram can be obtained by implementing the STFT on the waveform, which can be formulated as:

$$Y(t,f) = X(t,f) + N(t,f) \tag{1-2}$$

where $Y(t,f)$, $X(t,f)$ and $N(t,f)$ denote the spectrogram of noisy speech, clean speech, and noise speech, with t being the frame index and f being the frequency bin.

According to the type of input speech, the SE systems can be categorized into time domain and time-frequency (T-F) domain. The time-domain methods take the noisy speech waveform as input and directly estimate the enhanced speech waveform, where both magnitude and phase information of speech are naturally considered while avoiding the STFT computation. The T-F domain SE systems are implemented on the spectrogram of speech to learn the harmonic structures for

generating the enhanced spectrogram via mapping- or masking-based methods, which can efficiently capture the characteristics of speech signal like temporal and frequency correlations. Magnitude spectrogram is the most popular speech feature for T-F domain SE systems, where the noisy magnitude is enhanced during the training and phase information is retrained for speech reconstruction. To obtain the enhanced spectrogram, the mapping-based methods directly transform the noisy spectrogram into enhanced spectrogram by using regression-based models. Different from that, the masking-based methods predict the T-F masks which are used to suppress the noise components of noisy spectrogram, producing the enhanced spectrogram which will be transformed to the enhanced waveform by inverse STFT.

There are several types of masks commonly used for speech enhancement. The ideal binary mask (IBM) is of value either 0 or 1 via comparing the local criterion (LC) versus signal-to-noise ratio (SNR) of per time-frequency unit [13]. It is assigned 1 when the SNR of T-F units surpasses LC, otherwise, it is 0, defined as:

$$IBM(i,j) = \begin{cases} 1 & |C(t,f)|^2 - |N(t,f)|^2 > LC, \\ 0 & otherwise \end{cases} \tag{1-3}$$

where $C(t,f)$ and $N(t,f)$ denote the clean speech and noises at time index t and frequency index f, respectively. Based on the orthogonal assumption, $C(t,f) \cdot N(t,f) = 0$, the ideal ratio mask (IRM) is the ratio of signal power of the clean and mixture signals [14], formulated as:

$$IRM(t,f) = \left(\frac{|C(t,f)|^2}{|C(t,f)|^2 + |N(t,f)|^2}\right)^{\beta} \tag{1-4}$$

where β is an adjustable factor, generally set as 0.5. The output of IRM is between 0 and 1, where a higher value of IRM means more proportion of the clean speech in the mixture audio at each T-F unit. Additionally, the IRM with $\beta = 2$ is the Wiener filter, which is the best filter at mean square error (MSE). The complex ideal ratio mask (cIRM) can simultaneously enhance the amplitude and phase [15], which is a version of IRM in complex field, expressed as:

$$cIRM = \frac{M_r C_r + M_i C_i}{{M_r}^2 + {M_i}^2} + i\frac{M_r C_i - M_i C_r}{{M_r}^2 + {M_i}^2} \tag{1-5}$$

where M denotes the mixture of clean speech and noises, and r and i are the real and imaginary part, respectively. The ideal amplitude mask (IAM) [16] also depicts the energy ratio of the clean speech and the noisy mixture without orthogonal assumption:

$$IAM(t,f) = \frac{|C(t,f)|}{|C(t,f) + N(t,f)|} \tag{1-6}$$

The theoretic range of IAM is [0, $+\infty$]. The phase sensitive mask (PSM) [17] makes an improvement on IAM by adding a cosine similarity of the clean and noisy speech, defined as:

$$PSM(t,f) = \frac{|C(t,f)|}{|C(t,f)+N(t,f)|}\cos(\theta^C - \theta^M) \tag{1-7}$$

where $\theta^C - \theta^M$ is the phase difference of the clean and mixture speech. The value of PSM varies in [$-\infty$, $+\infty$]. Using PSM tends to obtain a better performance since it takes advantage of the phase information that is very important for the speech recovery. The optimal ratio mask (ORM) [18] is similar to IRM except for the orthogonal assumption of clean speech and the noise:

$$ORM(t,f) = \frac{|C(t,f)|^2 + R(C(t,f)N^*(t,f))}{|C(t,f)|^2 + |N(t,f)|^2 + 2R(C(t,f)N^*(t,f))} \tag{1-8}$$

where R and * denote the real part of the complex and conjugate calculation, respectively.

1.1.2 Commonly used neural networks for speech enhancement

Some popular DNNs are commonly used for speech enhancement, including fully-connected neural network (FC-NN), convolutional neural network (CNNs) and recurrent neural networks (RNNs). Generally, a DNN is comprised of multiple layers, where each one contains some necessary components like neurons, weights, biases, activation functions and normalization operation. By backpropagation algorithm during training stage, the parameters of neurons are updated to build a suitable mapping function between inputs and outputs of a DNN. In the following, we will introduce these commonly used DNNs and their constituent components.

Fully-connected neural network

The fully-connected neural network is the simple and initial architecture of DNNs, simply called as FC-NN, which has been widely used in speech enhancement. The FC-NN normally contains one input layer, multiple hidden layers and one output layers, which are comprised of neurons to perform the basic non-linear computation among its inputs. An example as the Fig. 1-1 shows, $x1$ and $x2$ are inputs of a neuron, which are multiplied with corresponding weights $w1$ and $w2$ to produce the outputs involving the bias. The output of this neuron can be denoted as:

$$Y = f(w1 \times x1 + w2 \times x2 + b) \tag{1-9}$$

where f is a nonlinear activation function to provide the non-linearity for neurons. Since the most

scenarios in real world are nonlinear, it is important to introduce nonlinearity into the output to encourage neural networks to simulate and solve more complex and practical problems.

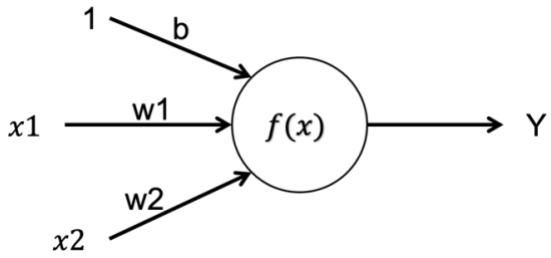


Figure 1-1: An illustration of a neuron

Activation function as an indispensable component for efficient neural networks, is applied for introducing the non-linearity and it should be continuous and differential so as to be incorporated into backpropagation algorithm. Three types of activation functions are commonly used in neural networks, including sigmoid, tanh and ReLU function [19] as shown in Fig. 1-2.

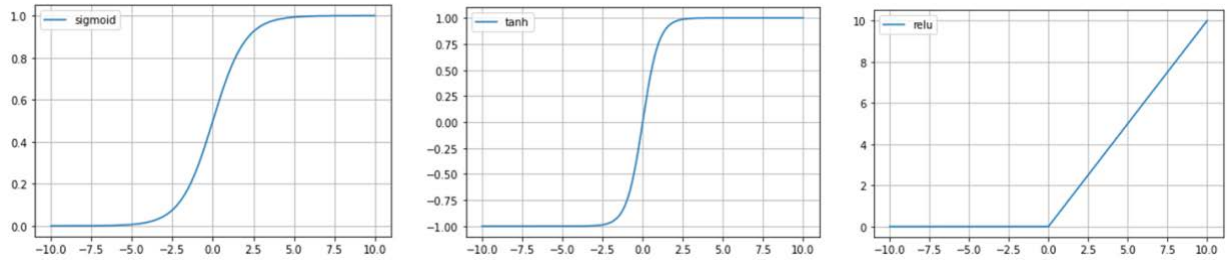


Figure 1-2: Sigmoid, Tanh and ReLU function

The sigmoid function is a smoothed curve with output value ranging from 0 and 1, which can be used for binary classification tasks, formulated as the follow:

$$f(x) = \frac{1}{1+e^{-x}} \tag{1-10}$$

When using it as activation function, training error would be rising as the number of hidden layers increases, where hidden layers close to output layer have big gradient and hidden layers near input layer get small gradient, which may lead to gradient vanishing problem in very deep model. The tanh function depicts a curve of hyperbolic tangent, with an output score between -1

and 1. Compared with sigmoid function, it alleviates gradient vanishing problem and performs faster computationally for training.

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{1-11}$$

The ReLU (rectified linear unit) function steadily outputs zero with input values below zero and output is the same as the input above zero, where gradients are respectively zero and a constant value, avoiding the gradient vanishing. Besides, it has much fewer complexity of computation, leading to a fast training. However, it might deactivate partial neurons if big gradients pass through, which may lose important information during training.

$$f(x) = \max(0, x) \tag{1-12}$$

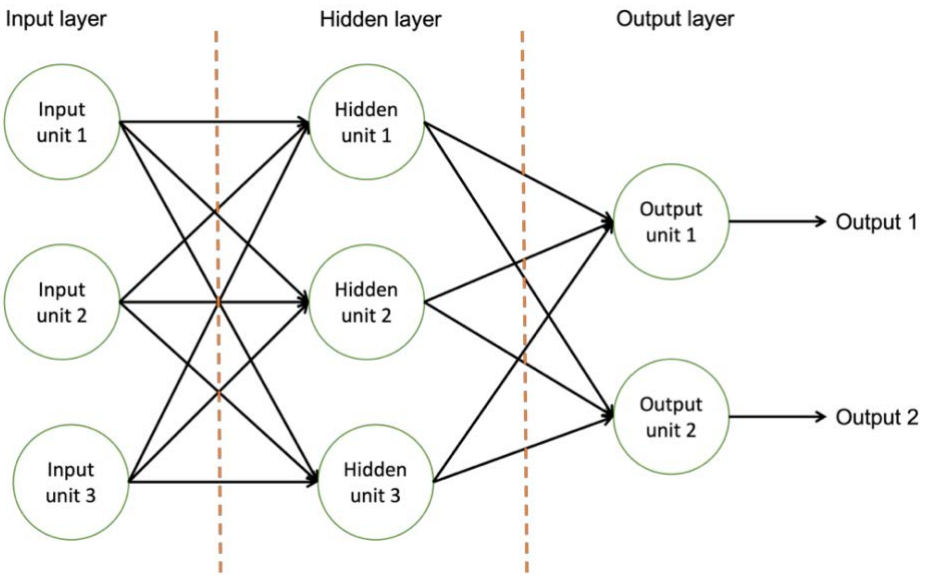


Figure 1-3: An illustration of the feed-forward neural network

As shown in Fig. 1-3, the FC-NN is built based on three types of neurons: input neurons, hidden neurons and output neurons. Input neurons absorb input signals into the neural network without any computation operation, and the output neurons compute and give out the final information to the outside. The hidden neurons are responsible for the computation and transmit computational output from input units to output ones. The three groups of neurons are used to construct the corresponding input layer, hidden layer, and output layer. To learn the complex transformation function between inputs and outputs, a FC-NN typically employs sufficient hidden layers to

explore the high-level features of input signals. It is worth mentioning that the gradient vanishing problems may potentially happen in a very deep FC-NN since the input layer is far from output layer, providing a very smaller gradient for the beginning few layers during the backpropagation.

The FC-NN is a basic learning machine for speech enhancement. Some works have investigated the FC based models to estimate the enhanced magnitude of speech signals or learn the estimated mask for multiplying with noisy speech to generate the enhanced speech. As shown in Fig. 1-4, the authors of [5] adopted a FC-based network for mask estimation in speech enhancement, which includes one input layer, one output layer and five hidden layers in between. In contrast, the authors of [6] directly mapped the noisy speech spectrogram into enhanced speech spectrogram by adopting the FC neural network as a regression model as shown in Fig. 1-5.

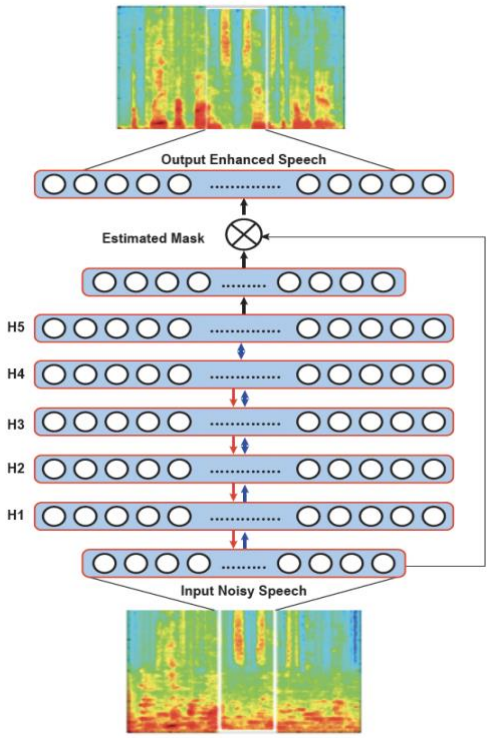


Figure 1-4: A FC model for masking [5]

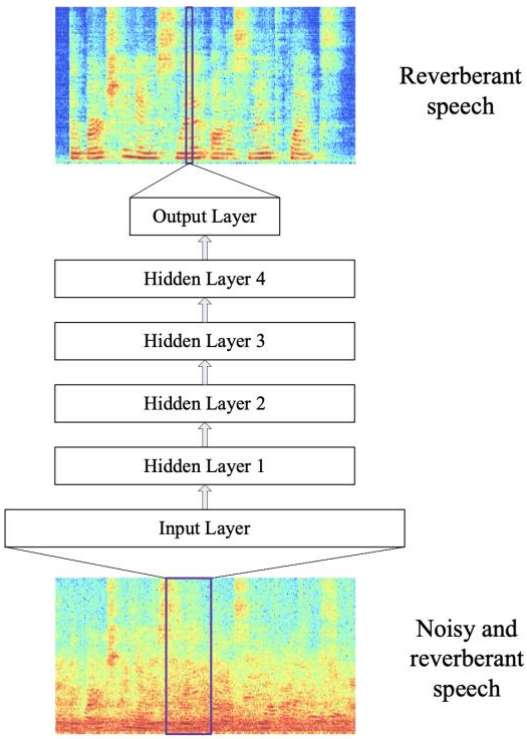


Figure 1-5: A FC model for spectra mapping [6]

Convolutional neural network

Different from FC-NNs involving redundant trainable parameters, CNNs can avoid the overfitting problems since they perform local operations and share the weights. The CNNs are originally used

to extract the spatial information of images, which has presented an impressive achievement in image processing tasks like image classification. Typically, as shown in Fig. 1-6, the CNN takes the 2-D image as input signal and employs several cascaded pairs of convolutional layer and pooling layer to extract the spatial information of image, where the convolutional layer extracts the local features within a limited window and the pooling layer aggregates the features to decrease the resolution of feature maps [20].

Generally, the beginning CNN layers can learn some simple features including textures, colors and edges. As the CNN becomes deeper, more complex and contextual features are extracted for the following processing in the classifier. The feature maps from CNN layers are flatten so as to be processed by the FC layers for classifying the input image.

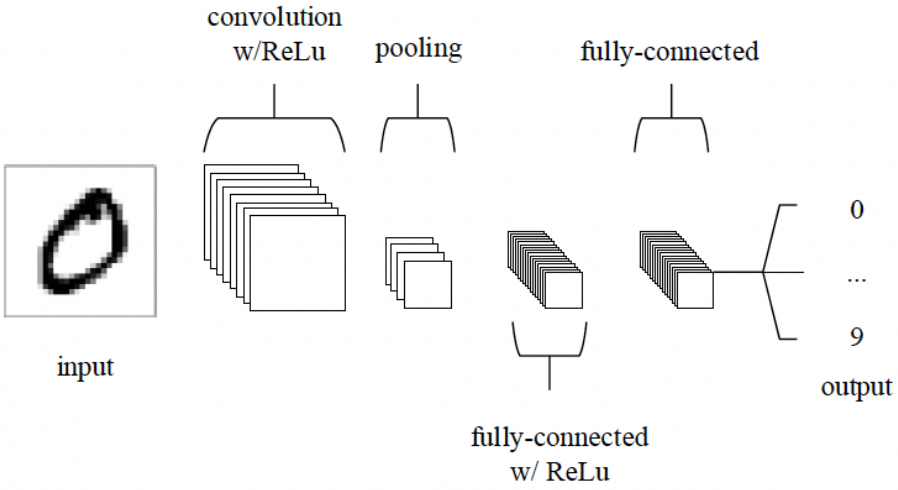


Figure 1-6: An illustration of a CNN for image classification [20]

In CNNs, the convolutional operation is achieved by employing multiple kernels applied in a sliding window across the entire input image via the weight sharing. Fig. 1-7 indicates the specific procedures of convolutional operation [21]. For the 5x5 feature map of input as represented in the blue grids, a 3x3 convolutional kernel is first assigned to the left-top region, whose values are shown in smaller fonts in blue grids. Then, the elements of kernel are multiplied with corresponding ones of feature maps, whose results are summed to produce the outputs of one location. By sliding the kernel window across the entire input feature map through weight sharing, the local correlations among all locations are learnt, generating the final result of one-kernel operation as shown in the green grids.

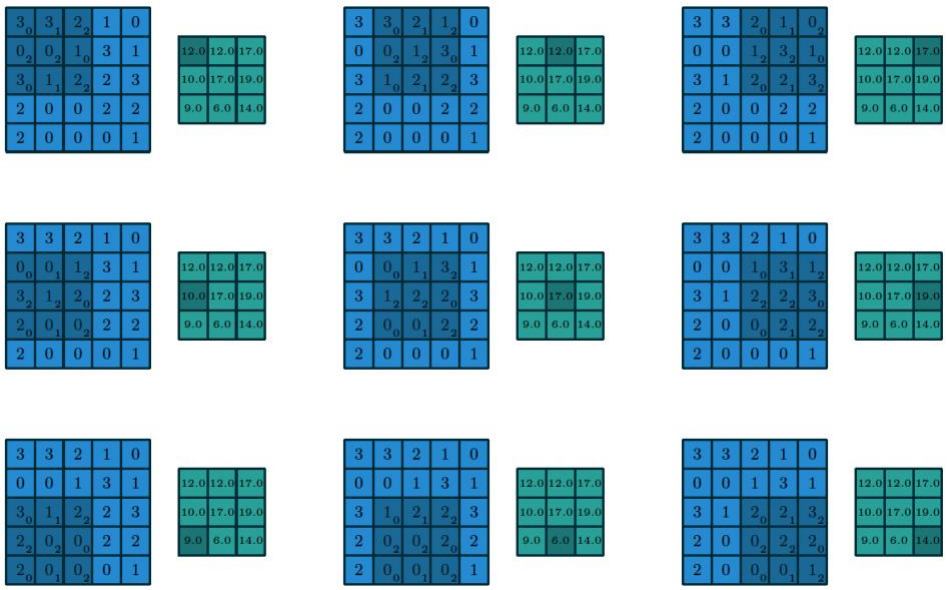


Figure 1-7: An example of the convolutional operation [21]

The CNN normally adopts multi-kernel operation to parallelly capture the features from different aspects, where outputs of each kernel operation are concatenated along the channel dimension to construct the generated feature maps. As shown in Fig. 1-8, three kernels are used to extract features in parallel, resulting in the 3-channel feature map.

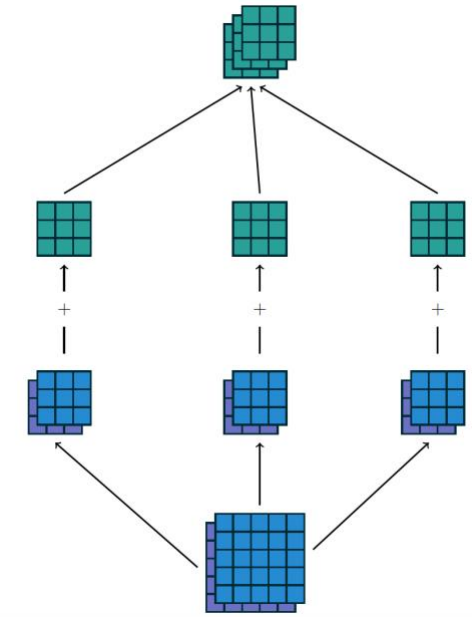


Figure 1-8: A mapping from two to three channels of convolution [21]

Apart from the basic convolutional computation itself, the pooling layer is usually associated with the convolution to perform the down-sampling operation, including maxing pooling and average pooling. The maxing pooling is used to select the maximum element representing the output feature of the kernel window. Compared to maxing pooling, the average pooling computes the average value among the kernel window to generate the output feature. The whole pooling operation performs as the convolution to consider all the locations of feature maps while no trainable parameters are involved.

Inspired by effectiveness of the CNNs on extracting features of 2-dimensional data, many SE systems widely adopt the CNNs to process the image-like spectrogram of speech signal. The authors of [7] proposed a CNN-based model to estimate the complex spectrogram as shown in Fig. 1-9. First, the real and imaginary spectrogram are concatenated along the channel dimension to form the 2-channel image-like data, which are then passed through a series of CNN layers to generate the contextual information. Finally, two FC layers are employed to separately estimate the enhanced real and imaginary spectrogram.

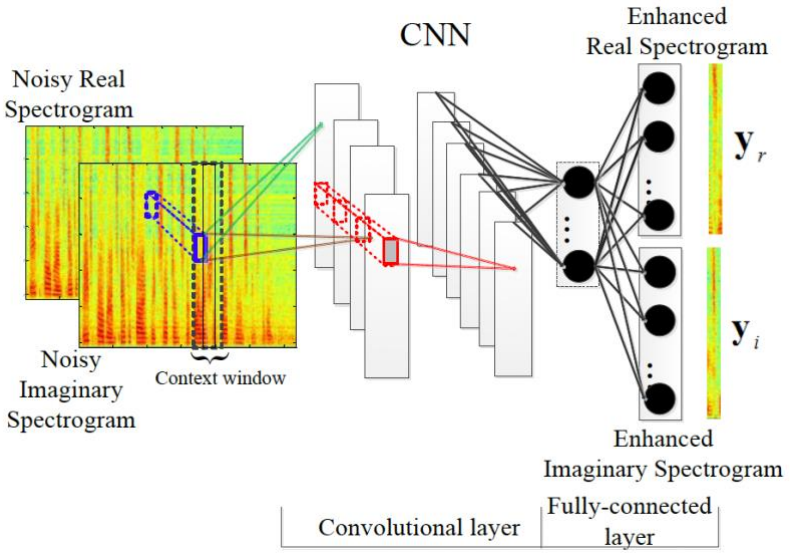


Figure 1-9: CNN on complex spectrogram estimation [7]

Apart from operating on the T-F domain to process the image-like spectrogram, the CNN-based models can be also directly implemented on the speech waveform by using one-dimensional convolutional layers. For long-range speech waveform, the stacked conventional 1-D convolution will only make receptive field grow linearly, which limits extracting global dependency of speech

sequences. To solve this problem, the authors of [8] proposed a WaveNet for time-domain speech enhancement with 1-D dilated convolutional layers, leading to an exponential growth of receptive field. As shown in Fig. 1-10, some stacked 1-D convolutional layers with exponential dilation values are adopted into noisy speech waveform for generating multi-scale features, which will be connected to be pass through 2-D convolutional layers for outputting the target fields.

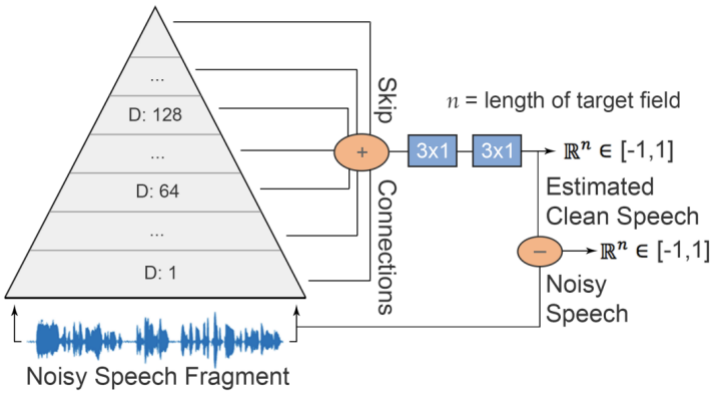


Figure 1-10: WaveNet for time-domain speech enhancement [8]

Recurrent neural network

Due to the intrinsic local operation of convolution, the CNNs require to be deep enough to extract the contextual information of long-range speech sequences, which constrains the performance of speech enhancement. To tackle this issue, the RNNs are proposed to model the temporal information of long-range sequences via sequential processing. In the vanilla RNN, the output of each time step will be saved in memory cell and then adopted by next time step, which means the information of previous time steps will affect the one of current time step. Benefited from the memory mechanism, the RNN can naturally process the tasks related to sequential signals like language processing and speech processing. As shown in the left part of Fig. 1-11, the vanilla RNN adopts the RNN cell to process the input of the current time step and the output of the last time step, whose procedure is recurrently conducted along the various time steps. By unfolding its recurrent structure as indicated in the right part of Fig. 1-11, the vanilla RNN can be treated as a DNN with many layers based on the length of input sequence. In addition, the output of the current time step will be used to update the memory cell which will be used again by the next time step. The procedure of each time step can be defined as follows:

$$h_t = \sigma(x_t \times \omega_{xt} + h_{t-1} \times \omega_{ht} + b) \tag{1-13}$$

$$\hat{y}_t = \sigma(h_t \times \omega_t + b) \tag{1-14}$$

where x_t and h_t are the input and the hidden state at time step t, ω_{xt} and ω_{ht} are transformation matrices for the input at time step t and the hidden state of the last time step, respectively, ω_t denotes the transformation matrix for generating the output of RNN cell, $\sigma(\cdot)$ means the sigmoid activation function, b denotes the bias, and $\hat{y}_t$ is the output of RNN cell at time step t.

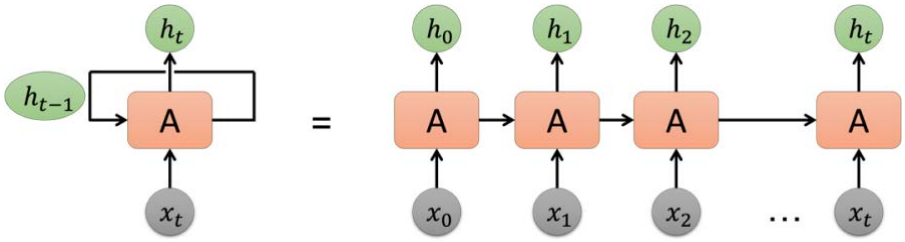


Figure 1-11: A basic structure of RNN

However, in the vanilla RNN, the hidden states of current time step only have an impact on the neighboring time steps due to its mechanism of memory cell, leading to a long-term dependency problem. More specifically, a certain time step is difficult to affect the long-distance positions since the information of memory cell is refreshened for each time step, in dealing with long-range input sequences. Therefore, the long-short term memory (LSTM) [22] is proposed to tackle the problems preventing the vanilla RNN from learning the long-range dependency.

Different from vanilla RNN, the LSTM is comprised of one memory cell and three gates to control the information flow shown in Fig.1-12, including the input gate, forget gate and output gate, respectively. The input gate controls how much input information should be used by the memory cell, the forget gate decides how much previous information from memory cell should be eliminated, and the output gate dominates how much information of memory cell should be adopted to generate the output.

By the gate mechanism, the LSTM can flexibly remember the important information and erase the inconsequential information from previous states, which dynamically learns the long-term dependency of input sequence. The three gate operations are given by:

$$z_o = \sigma(w_o[x_t; h_{t-1}]) \tag{1-15}$$

$$z_f = \sigma(w_f[x_t; h_{t-1}]) \tag{1-16}$$

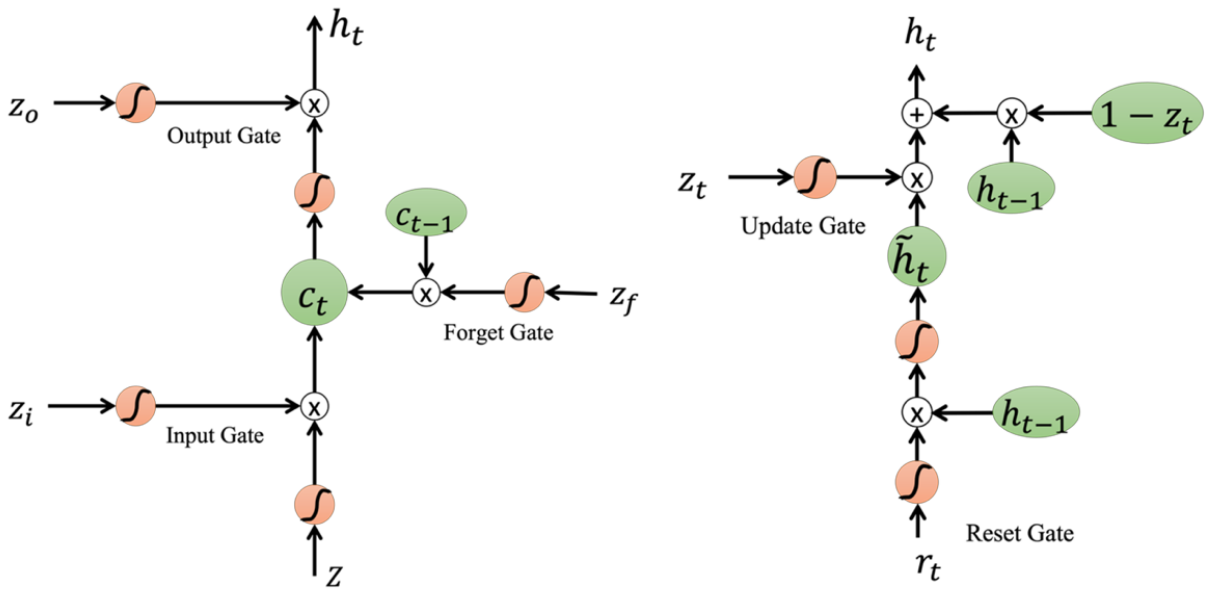


Figure 1-12: LSTM & GRU

$$z_i = \sigma(w_i[x_t; h_{t-1}]) \tag{1-17}$$

where x_t and h_{t-1} denote the input at time step t and hidden states at time step $t-1$, w_i, w_f and w_o mean the transformation for input gate, forget gate and output gate, respectively, $\sigma(\cdot)$ means the sigmoid activation function, z_i, z_f and z_o denote the outputs of three gate operations. All the gates have a value between zero and one to control how much the corresponding information should be remained or eliminated, whose procedures can be formulated as:

$$z = tanh(w[x_t; h_{t-1}]) \tag{1-18}$$

$$c_t = z_f \odot c_{t-1} + z_i \odot z \tag{1-19}$$

$$h_t = z_o \odot tanh(c_t) \tag{1-20}$$

$$y_t = \sigma(w_t h_t) \tag{1-21}$$

where w denotes the transformation matrix for input sequence, z is the feature representation of transformed input data, c_t denotes the hidden state in memory cell, w_t is the transformation matrix for obtaining the final outputs y_t at time step t, $tanh(\cdot)$ is the hyperbolic tangent function to curve the inputs between -1 and 1, and $\odot$ denotes the element-wise dot product operation.

Another efficient variant of RNN architecture is the gated recurrent unit (GRU) [23], which is modified based on LSTM by combining the forget gate and input gate into one update gate.

Compared to LSTM, the GRU involves less parameters and computational complexity while maintaining a similar performance in sequence modeling. In the GRU as shown in Fig. 1-12, the reset gate is introduced to control whether the hidden states at the previous time step should be used as the input. Furthermore, the update gate simultaneously controls how much the information of the previous time step should be employed and how much hidden states at current time step are outputted. The whole operations are defined as:

$$z_t = \sigma(w_z[h_{t-1}; x_t]) \tag{1-22}$$

$$r_t = \sigma(w_r[h_{t-1}; x_t]) \tag{1-23}$$

$$\tilde{h}_t = tanh(w[r_t \odot h_{t-1}; x_t]) \tag{1-24}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{1-25}$$

where w_z and w_r are the transformation matrices for update gate z_t and reset gate r_t, respectively, w denotes the linear transformation matrix for generating the transitional hidden state $\tilde{h}_t$ at time step t, and h_t means the final output at time step t.

Considering the effectiveness of RNNs in extracting the temporal information of sequential sequences, many works adopted the RNN-based layers as a part of neural network for speech enhancement. It is worth mentioning that the RNN-based layers can be unidirectional or bi-directional, where the former strictly captures the long-range dependency in temporal sequence to build a causal SE system, and the latter decides the output of certain time step by considering its previous and future positions.

In addition, the RNN-based layers can be flexibly incorporated into CNNs to construct a hybrid structure named convolutional recurrent network (CRN). Authors of [9] proposed a composite CNN-RNN based models for speech enhancement as shown in Fig. 1-13, where the RNN layers are used to explore the contextual information from the feature maps of CNN layers.

More specifically, the stacked convolutional layers are used to extract the abundant spatial features, generating the multi-channel feature maps with temporal and frequency information. Next, the features are concatenated along the frequency dimension to be passed through the bi-directional RNN layer, where the long-range dependency of speech sequence is explored to obtain the contextual features. Finally, the FC neural network is used to transform the features into enhanced speech spectrogram.

Authors of [10] proposed a fully RNN-based model for waveform-based speech enhancement, where the GRU layers are used to construct the hourglass architecture to process the high-

resolution sequences as shown in Fig. 1-14. In the proposed model, the GRU layers of the lower pyramid aim to decrease the number of time steps and increase the feature dimension, while the ones of the upper pyramid are used to perform the inverse operation. In addition, the residual connection is inserted into corresponding layers from the lower pyramid to the upper one to avoid the gradient vanishing problem.

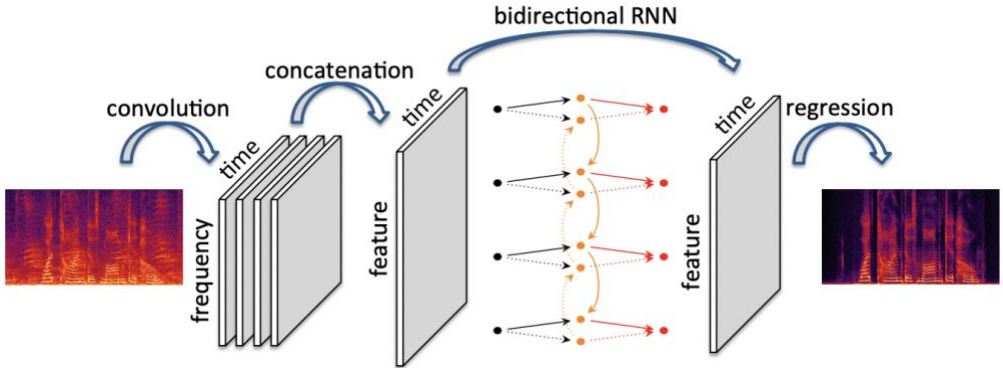


Figure 1-13: A convolutional RNN for speech enhancement [17]

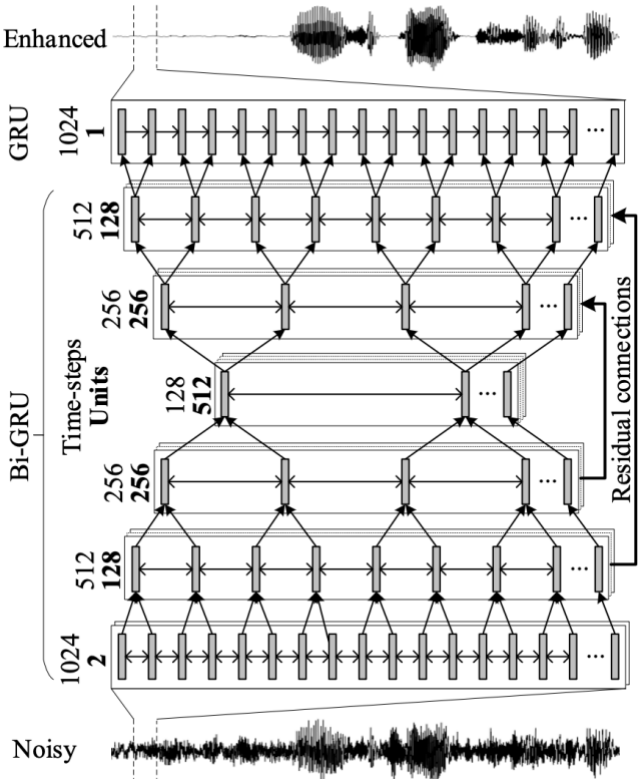


Figure 1-14: A fully RNN-based model for waveform-based speech enhancement [18]

1.1.3 Attention mechanism for sequence processing

Although the RNN-based models have achieved an impressive performance in sequence modeling, the sequential operation of RNNs brings high computational burdens for processing long-term speech sequences. In addition, the long-term memory is still difficult to be saved by RNNs, making the long-range dependency problem exists in the RNN-based SE systems. Recently, self-attention mechanism is proposed to efficiently capture long-term dependency of sequences with parallel operation, which has promoted the advancement on NLP and computer vision tasks. The attention mechanism comes from the observation behavior of creatures. For example, when looking at a picture, people may catch a specific area where more attention is paid, as opposed to a global view on the picture. This allows us to focus on the important part to learn more details about it and suppress less important information at the meantime.

The self-attention, as one of the attention mechanisms, is operated on the input sequence itself, where each sample position in the sequence will be affected by remaining ones with various attention weight. For example, when the state of certain sample point depends on the neighboring positions, more attention weights will be assigned to regional scope around the sample to perform the local attention. In contrast, assigning most attention weights to the distant sample regions will build a global attention, which extracts the contextual information of long-range speech sequences.

In the self-attention mechanism as shown in Fig. 1-15, the input sequences are first transformed into keys, queries and values representation by using three different linear layers. Next, the attention weight matrix is obtained by calculating the similarity between each query and all keys. Then, the softmax function is adopted to transform the similarity values into probability distribution between 0 and 1. Finally, the output of self-attention is obtained by summing the multiplying results between each value and the corresponding attention weight.

To calculate the attention weights on the queries and keys representation, some practical approaches have been proposed:

Dot-product attention: Dot-product attention is a simple method to compute the similarity between query and key matrices, which requires the two matrices to have the same dimension. Due to the capacity of efficient computation, the dot product method is commonly used for attention calculation while not involving any trainable parameters, whose formula is indicated as:

$$A(q,k) = q^T k \tag{1-26}$$

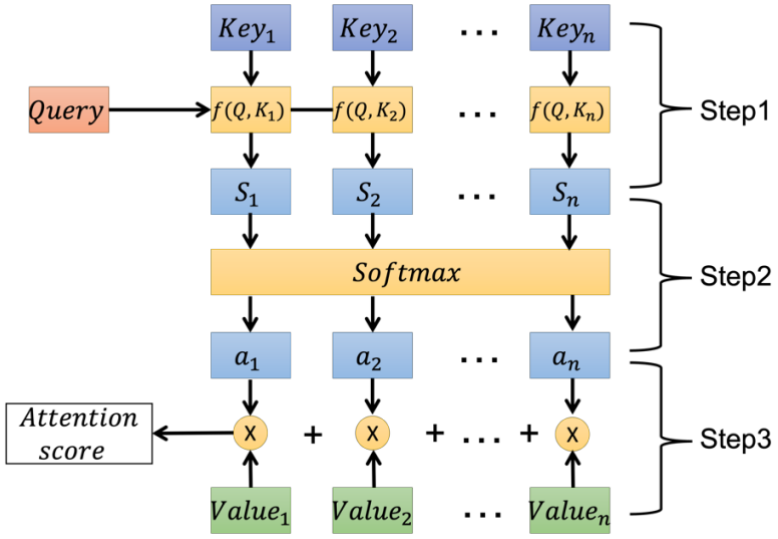


Figure 1-15: A common framework of attention mechanism

where q and k mean the query and key representation, respectively.

Scaled dot product attention: The scaled dot-product attention is an attention mechanism where the dot products are scaled down by a factor. Assuming that the query and key are d-dimensional vectors whose elements are independent random variables with mean 0 and variance 1. After applying the dot-product attention, the generated output will have mean 0 and variance d, which will have a potentially bad influence on the model training. Thus, the result of dot product attention is divided by a factor $\sqrt{d}$, producing the values having a variance 1. The scaled dot-product attention can be defined as:

$$A(q,k) = \frac{q^T k}{\sqrt{|k|}} \tag{1-27}$$

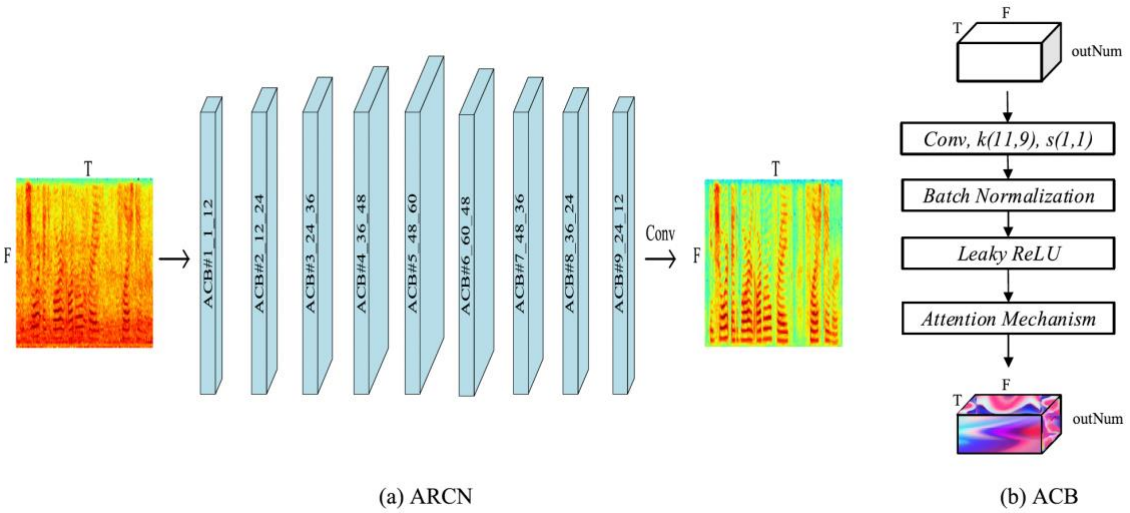
Neural network based attention: Attention can also be realized by a trainable neural network such as FC neural network. More specifically, the query and key vectors are concatenated together to form the input, which are passed through a two-layer FC neural network with tanh activation function to generate the attention. This method can make FC layers dynamically generate the efficient attention weight by training on the large-scale datasets, whose definition can be found as:

$$A(q,k) = W_2 \tanh\left(W_1[q;k]\right) \tag{1-28}$$

where W_1 and W_2 denote the transformation matrixes of two FC layers.

Authors of [11] proposed an attention-based architecture for speech enhancement as shown in Fig. 1-16, where the attention mechanism is incorporated into CNN layers to extract contextual

information. The proposed model takes the speech spectrogram as input and adopts stacked attention-based convolutional blocks (ACBs) to capture the abundant speech features. Each ACB consists of a 2-D convolutional block and attention block, where the convolution block is used to extract spatial features of spectrogram through a convolution with kernel size of (11, 9) with stride 1, which is then followed by batch normalization and a Leaky ReLU activation.



(a) ARCN (b) ACB

Figure 1-16: CNN with attention mechanism. (a) The whole framework of the model called attention-based redundant convolutional network (ARCN), where the parameter on each block means the order number of the blocks and the time-frequency shape of input-output feature maps for each block, respectively. (b) The attention-based convolutional block (ACB) that builds the ARCN [11].

The feature maps from convolutional operation are passed through attention block comprised of channel-wise attention and spatial attention, where the former captures which features are important, and the latter focuses on where is informative. In the channel-wise attention as shown in Fig. 1-17, the input features are first processed by the global average pooling along the time and frequency axis, generating a vector descriptor of channels where each component is the average value among each T-F feature space.

Next, the bottleneck FC neural network is adopted to generate the channel-wise attention weights, which are multiplied with each spatial pixel along the channel dimension to produce the final outputs of channel-wise attention. Different from the channel attention squeezing the T-F dimensional size while keeping the channel size, the spatial attention first squeezes the channel dimension but maintains the spatial T-F dimension by using two pooling layers and three dilated

convolutions, creating the five-channel T-F feature which will be processed by a convolution with one kernel to create the spatial attention weights. In the following, the attention weights are assigned to the spatial pixel along the T-F axis to capture the crucial spatial features. Note that the work explores different combination methods to connect the channel-wise attention with spatial attention, yielding parallel and cascaded connections.

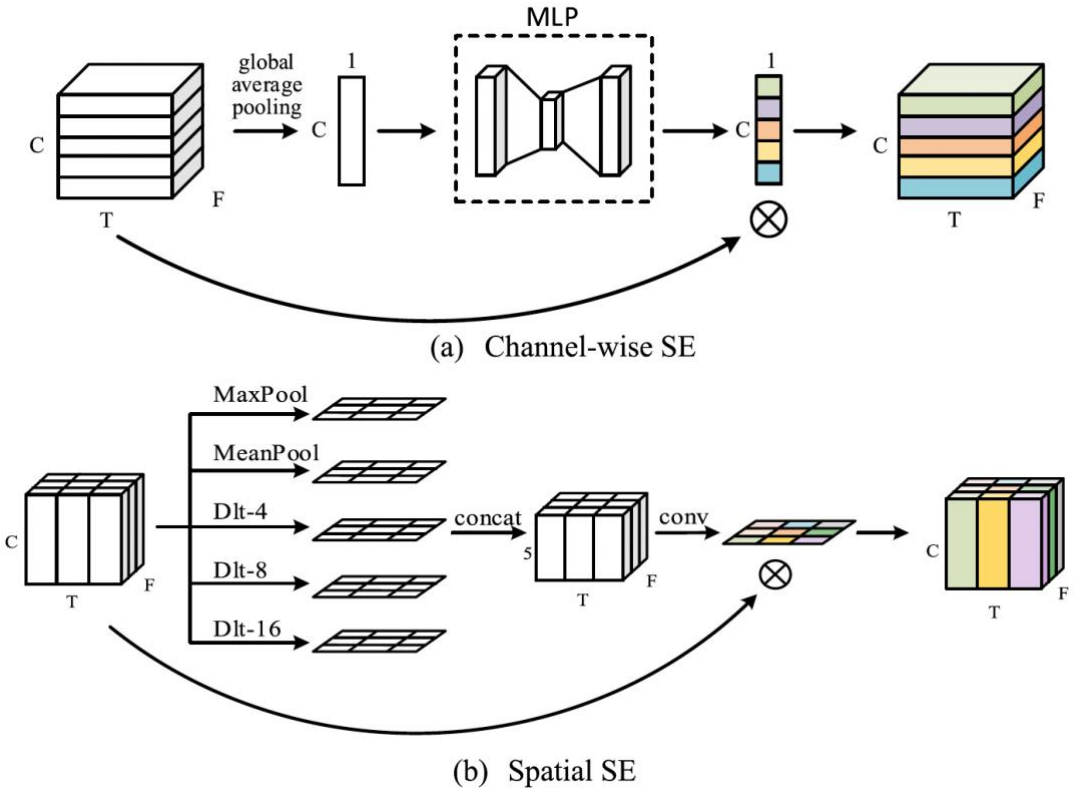


Figure 1-17: Attention mechanism in ARCN model [12]

Authors of [12] proposed a dual-path attention block to extract the contextual information of speech sequences. As shown in Fig. 1-18, the features are first fed into two residual blocks to learn the local correlations by using several CNN layers. In the subsequent processing, the self-attention mechanism is separately applied into temporal and frequency dimension of speech features to simultaneously extract global contextual information of temporal and frequency features, leading to the frequency-wise self-attention and temporal self-attention flow, respectively. Finally, the outputs of residual blocks and two attention blocks are concatenated to pass through a 2-D convolution for obtaining the feature representations of proposed attention method.

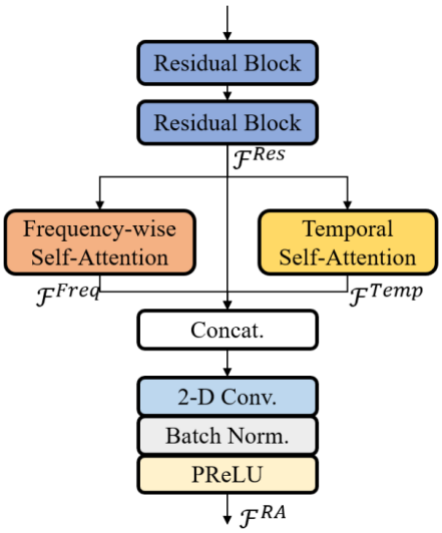


Figure 1-18: A dual-path attention block for speech modeling [12]

1.2 Datasets and evaluation criteria for SE neural networks

The neural networks are trained based on a number of audio segments to distinguish features of clean and noisy speech, where a collection of audio pieces is called speech dataset. Here are some speech datasets often adopted for speech enhancement.

1) TIMIT corpus [24]: The dataset is designed by Texas instruments and Massachusetts institute of technology (TIMIT), providing 8 types of dialects of American English from 630 speakers of 10 spoken sentences for each one.
2) LJ speech dataset [25]: The dataset is recorded by the LibriVox project, including 13,100 utterances with a length between 1 and 10 seconds, which is totally about 24 hours recorded from a female speaker reading 7 non-fiction books.
3) WSJ0 SI-84 dataset [26]: It consists of reading speech of texts from Wall Street Journal news, which is a machine-readable corpus containing 7138 utterances from 83 speakers including 41 females and 42 males.
4) VCTK dataset [27]: A public dataset provided by Valentini et.al., which contains 30 speakers chosen from Voice Bank corpus [28]. It provides well-prepared training set and testing set of 11572 audios from 28 speakers and 824 audios from 2 speakers, respectively. The noisy speech for training involves 10 noises at SNRs of 0, 5, 10 and 15 dB, where 2

types of noises are artificially generated and 8 real noises. There are 5 unseen noises with SNR levels of 2.5, 7.5, 12.5 and 17.5 dB for testing. The noises are selected from the DEMAND [29].

Except for the VCTK dataset above, the other three datasets cannot be applied for training directly due to the lack of noises. Researchers require to mix clean audios with other noise sources to produce their own training set and testing set. All experiments in this thesis are conducted on the VCTK dataset. VCTK dataset is open to the public and has been adopted by many researchers to evaluate their models. Therefore, a direct comparison of model performance on the same dataset will be given in the later chapters.

Table 1-1: Evaluation criteria for speech enhancement [30]

S. No	Evaluation measure	Mathematical expression	Purpose
1	PESQ (Rix et al. 2001) ITU-T P.863 recommendation	$\text{PESQ} = \alpha_0 - \alpha_1 \cdot \text{A}_{\text{ins}} - \alpha_2 \text{B}_{\text{ins}}$	Speech quality
2	BSD (Loizou 2011) Bark distortion measure	$\text{BSD}(\vec{b}_x, \vec{b}_{\hat{x}}) = \sum_{k=1}^{N} [\vec{b}_x(k) - \vec{b}_{\hat{x}}(k)]^2$	Speech quality/distortion
3	LLR (Quackenbush 1995) log likelihood ratio distance	$\text{d}_{LLR}(\vec{b}_x, \vec{b}_{\hat{x}}) = \log\left(\frac{\vec{b}_x R_x \vec{b}_x^T}{\vec{b}_{\hat{x}} R_x \vec{b}_{\hat{x}}^T}\right)$	Speech quality and spectral distance
4	IS (Quackenbush 1995) Itakura-Saito distance	$\text{d}_{IS}(\vec{b}_x, \vec{b}_{\hat{x}}) = \frac{\sigma_x^2}{\sigma_{\hat{x}}^2}\left(\frac{\vec{b}_x R_x \vec{b}_x^T}{\vec{b}_{\hat{x}} R_x \vec{b}_{\hat{x}}^T}\right) + \log\left(\frac{\sigma_x^2}{\sigma_{\hat{x}}^2}\right) - 1$	Speech quality and spectral distance
5	CEP (Loizou 2011) Cepstral Distance	$\text{d}_{CEP}(\vec{b}_x, \vec{b}_{\hat{x}}) = \frac{10}{\log_e 10}\sqrt{a\sum_{k=1}^{p} [\vec{b}_x(k) - \vec{b}_{\hat{x}}(k)]^2}$	Speech quality and spectral distance
6	SNRSeg (Loizou 2011) Segmental SNR	$\text{SNR}_{\text{SEG}}(\text{m},\omega_m) = \frac{10}{\text{M}} \sum_{\text{m}=0}^{\text{M}-1} \log_{10}\left(\frac{\lvert\text{S}(\text{m},\omega_m)\rvert^2}{\lvert\lvert\text{S}(\text{m},\omega_m)\rvert - \lvert\text{S}_{\text{EST}}(\text{m},\omega_m)\rvert\rvert^2}\right)$	Speech quality and noise suppression
7	FWSNRSeg (Loizou 2011) Frequency weighted segmental SNR	$\text{FWSNR}_{\text{SEG}}(\text{m},\omega_m) = \frac{10}{\text{M}} \sum_{\text{m}=0}^{\text{M}-1} \frac{\sum_{j=1}^{k} B_j \log_{10}\left[\frac{F^2(m,j)}{F(m,j)-\hat{F}(m,j)}\right]}{\sum_{j=1}^{k} B_j}$	Speech intelligibility/speech quality
8	Composite measures (Loizou 2007)	$\text{Csig} = 3.093 - 1.029\text{S}_{\text{LLR}} + 0.603\text{S}_{\text{PESQ}} - 0.009\text{S}_{\text{WSS}}$ $\text{Cbak} = 1.634 + 0.478\text{S}_{\text{PESQ}} - 0.007\text{S}_{\text{WSS}} + 0.063\text{S}_{\text{SNR}_{\text{SEG}}}$	Speech distortion and residual noise
9	STOI (Taal et al. 2010)	$f(\text{STOI}) = \frac{100}{1+\exp(\sigma\text{STOI}+\delta)}$	Speech intelligibility
10	SNR_{LOSS} (Ma and Loizou 2011)	$\text{SNR}_{\text{Loss}} = 10\log_{10}\frac{\text{X}(k,m)^2}{\hat{\text{X}}(k,m)^2}$ $\text{Loss} = \text{SNR}_{\text{X}}(k,m)^2 - \text{SNR}_{\hat{\text{X}}}(k,m)^2$	Speech intelligibility
11	fAI [112] Fractional Articulation Index	$f\text{AI} = \frac{1}{\sum_{k=1}^{m} \text{W}_k}\sum_{k=1}^{m} \text{W}_k . f\text{SNR}_k$	Speech intelligibility

A number of evaluation criteria have been proposed to measure speech quality as shown in Table. 1-1. There are six criteria used in my research work. I adopt these commonly used criteria to compare my research work with existing methods for demonstrating the performance:

- Perceptual evaluation of speech quality (PESQ) [31]: a very popular metric computed by

comparing clean speech and denoised speech with a score between -0.5 and 4.5.

- Short-time objective intelligibility (STOI) [32]: a metric calculated on correlation coefficient of clean speech and enhanced speech in temporal envelopes on short-time and overlapped pieces, which has a value ranging from 0 to 1.
- Segmental signal-to-noise ratio (SSNR) [33]: It computes the SNRs and energies of segments from the estimated and clean speech, giving a score usually between -10 and 35.
- Three composite objective metrics: CBAK, CSIG and COVL are for noise distortion, signal distortion and overall speech quality, respectively. Their score varies between 0 and 1 [34].

1.3 Training strategies

A deep-learning model is trained via a large dataset based on the loss function to approximate the target output, where the loss function aims to provide the gradients for updating the weights and biases by back-propagation algorithm. By iteratively updating the model parameters, the distance between model outputs and ground truth is decreased. However, sometimes the loss in the testing data is still unexpectedly large even though the one in the training data has been very low, which is potentially caused by the overfitting problem. To address this issue, the normalization technologies are introduced to promote the model training and generalization capacity. Here I briefly introduce some commonly used loss functions and efficient normalization methods for efficiently training neural networks.

1.3.1 Loss function

The loss function is a measurement of the distance between the estimate and the expected result for each output, which is used to transit the gradients for updating model parameters by back-propagation algorithm. Here are some loss categories commonly used for training models with deterministic target.

1) Mean average error (MAE): a loss in L1 norm, computes the average of absolute difference between the ground truth y_i and the prediction $\hat{y}_i$:

$$loss_{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{1-29}$$

2) Mean square error (MSE): a loss in L2 norm, computes the average of square difference between estimate and target:

$$loss_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \tag{1-30}$$

Generally, it converges faster than the MAE loss during training while the robustness of the MSE becomes a little worse due to it sensitive to outliers.

3) Loss regularization: It is to add a penalty term of L1/L2 norm into loss function to restrain the considerable weight, which is often used for avoiding overfitting problem. The regularized loss functions of L1 and L2 norm are defined below:

$$L^{L1} = L(\hat{y}, y) + \alpha\sum_{i=1}^{M}|w| \tag{1-31}$$

$$L^{L2} = L(\hat{y}, y) + \alpha\sum_{i=1}^{M}(w)^2 \tag{1-32}$$

where M is the number of the weights w and α is a hyperparameter to control the scale of regularization.

4) Signal distortion rate (SDR) loss:

$$loss_{SDR} = 10log_{10}\frac{||y||^2}{||E||^2} \tag{1-33}$$

where y and $E = y - \hat{y}$ denote the ground truth and the error between y and estimate $\hat{y}$.

1.3.2 Normalization

Normalization techniques are commonly used in each network layer to make outputs of each layer have similar mean and variance, which could solve the challenges of training models and help network converge faster by gradient descent. Here are several types of normalization methods that have been proven effective and widely used in deep neural networks as shown in Fig. 1-19.

1) Batch normalization (BN) [35]: It computes the mean and deviation of input features based on the batch scale, thus is called the batch normalization. Suitable utilization of the BN can usually assist model to obtain expected results based on the considerably large batch size, where the hardware with abundant memory is required.
2) Layer normalization (LN) [36]: Different from BN, LN operates on the individual samples and normalize the input of each layer, which can be used in mini-batch training without interference from data distribution. LN does not have to store the mean and deviation of

the batch thus it can save memory during training.

3) Instance normalization (IN) [37]: It operates on the channels to obtain the mean and deviation of feature maps within each sample. The processing along channels is found affecting the style of generated images thus the IN is initially applied to generative models used for the image style transfer. Like the LN, IN can also be used for mini-batch training to improve performance with little memory occupied.
4) Group normalization (GN) [38]: It is a trade-off between LN and IN, which first divides the channels of each sample feature maps into several groups, then computes the mean and deviation independently in each group of channels. GN is often recommended for the task occupying large memory such as image segmentation and complicated attention-based models.

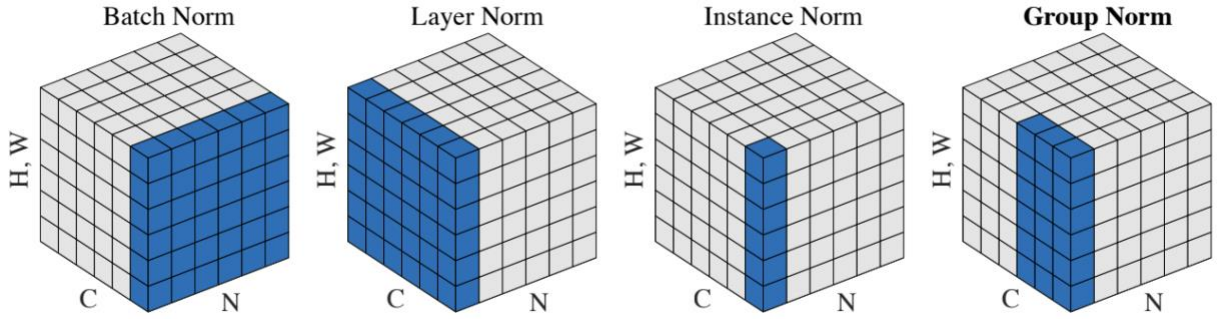


Figure 1-19: Normalization methods [38]

1.4 Objective and organization of the thesis

The objective of this thesis is to propose efficient attention-based U-Net models for single-channel speech enhancement in the time domain. In the first contribution, we propose a self-attention based U-Net with dual-branch attention, where the spatial attention is used to capture the spatial information and the channel-wise attention is employed to extract channel information of speech features. By incorporating the dual-branch attention between encoder and decoder, the contextual information of long-range speech sequences is learnt, which indicates a comparable performance to existing U-Net based structures.

In the second contribution, we propose two multi-head attention based U-Nets, where the attention mechanism can learn the correlations of speech sequences from different aspects. In the first U-Net model, we propose sample-wise and frame-wise attentions with cascaded structure, which are inserted between encoder and decoder, leading to a context-aware U-Net. More specifically, the sample-wise attention is adopted to extract samples features of each individual frame, whose outputs are then processed by the frame-wise attention to capture the relationship between different frames. By using multiple stacked cascaded attentions, the contextual information of long-term sequences is efficiently explored. Most U-Net based SE systems adopt the convolutional encoder-decoder structure, which requires deep enough encoder and decoder to extract the large receptive fields while bringing the overfitting problem and parametric overhead. To tackle this issue, we further propose multi-head attention based encoder and decoder layers, where both of them are based one sample and frame attention except that the encoder uses the down-sampling block before attention while the decoder adopts the up-sampling block. By building attention-based encoder and decoder layers, the extraction of contextual information can perform on the encoder and decoder, which can solve the intrinsic locality of convolution and optionally remove the inserted block between encoder and decoder. We have demonstrated that our proposed multi-head attention U-Nets achieve an impressive performance of speech enhancement among current SE model while involving fewer trainable parameters.

The rest of this thesis is organized as follows:

Chapter 2: Previous U-Nets for speech enhancement are first introduced, based on which we propose a novel U-Net with dilated-dense convolutional layers as encoder and decoder and a dual-branch attention block before decoder. Experiments on a benchmark dataset indicates that the proposed models provided competitive performance among existing methods.

Chapter 3: This chapter first introduces multi-head attention mechanisms used in previous SE works. It then presents two efficient U-Nets incorporating the proposed multi-head attention block at the center of U-Net structure and inside each encoder-decoder layer, respectively. To demonstrate the performance of our proposed methods, we carry out simulation-based experiments on a standard dataset and show that most evaluation results of our models are better than that of the comparison models.

Chapter 4: It gives the conclusion of this thesis and suggestions for future work.

Chapter 2
Proposed Dual-branch Attention U-Net for speech enhancement

In this chapter, we propose a dual-branch attention U-Net (DBAUNet) for speech enhancement. This new architecture is comprised of encoder, decoder and dual-branch attention block in between. In the dual-branch attention block, the channel attention and spatial attention are proposed to extract spatial and channel information, which are then fused to obtain the contextual information.

In the first section, some previous works about initial U-Net structure are first introduced, including a couple of existing U-Net based methods for speech enhancement and some efficient components related to our proposed model. In the second section, we elaborated each component of our proposed model in details. In the experimental result section, we demonstrate the efficiency of our proposed models by setting various groups of comparison models using different configurations and parameters. It is shown that our proposed model achieve a comparable performance in speech enhancement.

2.1 Previous work

2.1.1 Introduction of U-Net neural network

The U-Net structure was first introduced in biomedical image segmentation, which is generally comprised of a down-sampling part in the left side and an up-sampling part in the right side of the U-shape architecture [39]. As the convolutional neural network achieves an impressive performance in image recognition, convolutional blocks are usually adopted in U-Net to perform down-sampling and up-sampling operations.

As shown in Fig. 2-1, the input images are processed by multiple down-sampling convolutional blocks to obtain the compressed feature representations in which some redundant information is removed. The up-sampling convolutional blocks are employed to reconstruct the compressed

features to output segmentation map which has the same size as the input image. The feature size is halved while the number of feature channels is doubled by each down-sampling convolutional block, which involves two 2D-convolutional layers using a filter of size (3, 3) and a max pooling layer using kernel of size (2, 2) with 2 strides. The ReLU nonlinearity is inserted between two convolutional layers to avoid overfitting. On the contrary, each up-sampling convolutional block doubles feature size and halves feature channels by using two up-convolution layers with ReLU nonlinearity in between.

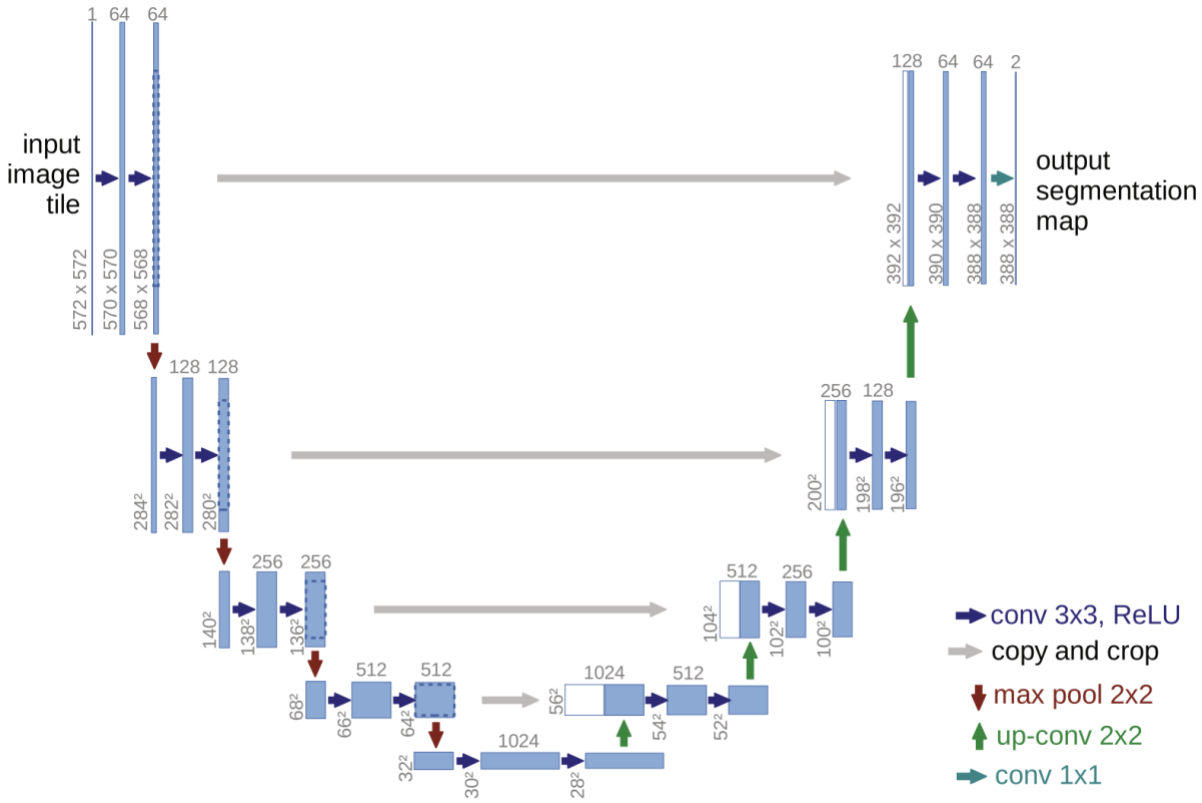


Figure 2-1: Original U-Net for image segmentation [39]

Moreover, the output of each down-sampling layer is concatenated to the corresponding up-sampling layer, called skip connections, to reuse the useful information from encoder layers to help information recovery at decoder. A common practice of up-sampling operation is the transposed convolution, which is an approach to recover information in signal processing through convolving padded feature maps to obtain output signal with the same size as input.

2.1.2 U-Net for speech enhancement

Inspired by the effectiveness of U-Net architectures in medical image processing, some U-Net based methods are adopted in speech enhancement and separation tasks. Wave-U-Net is a classic U-Net application solving speech separation, exclusively implemented on the waveform in the time domain at different time scales [40]. The goal is to obtain K sources of audio waveform separate W^1, W^2, …, W^K from the input mixture of waveform $I \in [-1, 1]^{S \times C}$, where S is the number of speech samples and C denotes the number of speech channels. Based on this method, the authors of [41] applied the Wave-U-Net on the speech enhancement (Fig. 2-2) by replacing the multi-source audio of input with a noisy mixture and separating it into the denoised target speech and the noise. In terms of single-channel speech enhancement task, here $C = 1$ and $K = 2$.

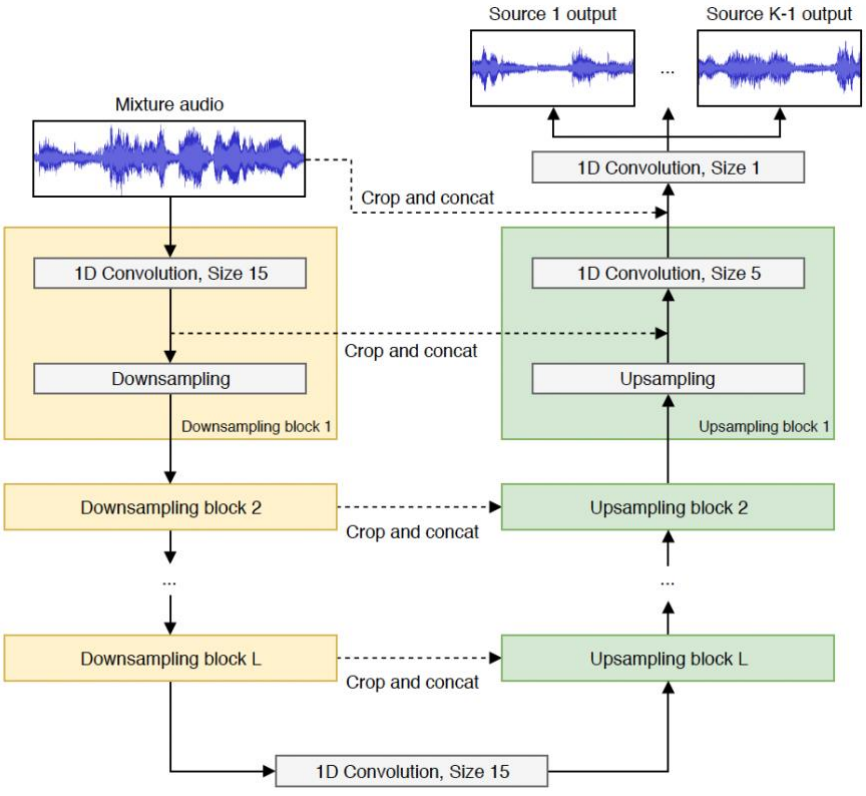


Figure 2-2: Wave-U-Net for speech segmentation [40]

In speech enhancement, the U-Net based structure usually employs a convolutional encoder-decoder with skip connection to estimate the clean waveform from noisy speech mixture. However,

a CNN based model requires very deep convolutional layers to enlarge receptive field since the convolution operation only focuses on the information within local region. Capturing long-range and global contextual information is really critical for speech sequence modeling. To boost the effectiveness of the U-Net, another popular practice is to embed other types of neural networks as partial components between the encoder and the decoder of U-Net to further extract the features after down-sampling operation. The authors of [42] inserted the Bi-directional LSTM between the encoder and decoder of U-Net to further extract the long-term information of speech sequences, which is modified from a U-Net based model for music source separation [43].

As shown in Fig. 2-3, the DEMUCS model is composed of encoder, decoder and two LSTM layers in between. The input noisy waveform is first down-sampled by several encoder layers to obtain the compressed features, which will be further extracted by two LSTM layers to learn the long-range dependency from the past. After that, the decoder layers convert the output features of LSTM to the enhanced speech waveform. More specifically, for the encoder, each layer comprises a 1D-convolution with $2^{i-1}H$ kernels of size K and stride S and a one-by-one convolution with $2^{i}H$ kernels of size 1 and stride 1, where i denotes the index of encoder layer. Note that, the two convolutions are followed by ReLU and GLU [44] function, respectively.

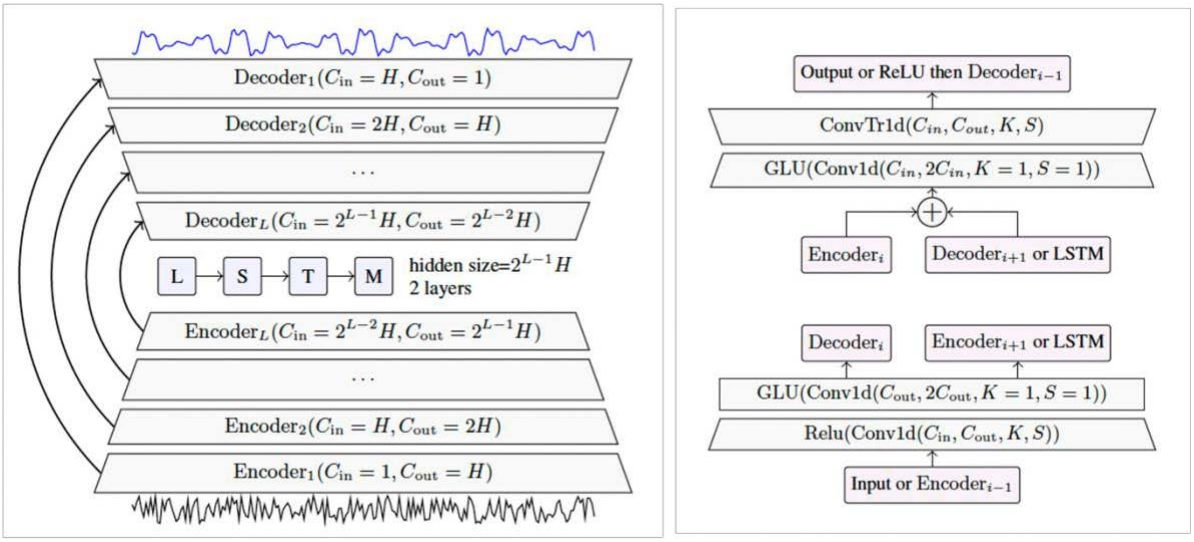


Figure 2-3: The DEMUCS and an illustration of layer connections of the model [42]

To extract global contextual information, the sequence mapping model then processes the outputs from the encoder E_o and outputs $L_o = LSTM(E_o) + E_o$, where LSTM has two layers and

the number of hidden units is 2^{L-1}. Next, the decoder takes L_o as input and operates reversely to the encoder by using 1D transposed convolution, thus generating the enhanced speech waveform. Note that, C_{in} and C_{out} respectively denote the number of input and output channels of each layer. H is the number of initial hidden channels of encoder layers and the number of the final hidden channels of decoder layers due to the symmetrical structure of the U-Net and L means the layer number of both the encoder and the decoder.

Although the LSTM achieves impressive performance in sequence modeling tasks, it cannot perform parallel processing, thus leading to low-speed processing especially for long speech sequences. The temporal convolutional network (TCN) based on the dilated convolution, as an alternative of RNN layer, has been proposed for the sequence model [45] as shown in Fig. 2-4.

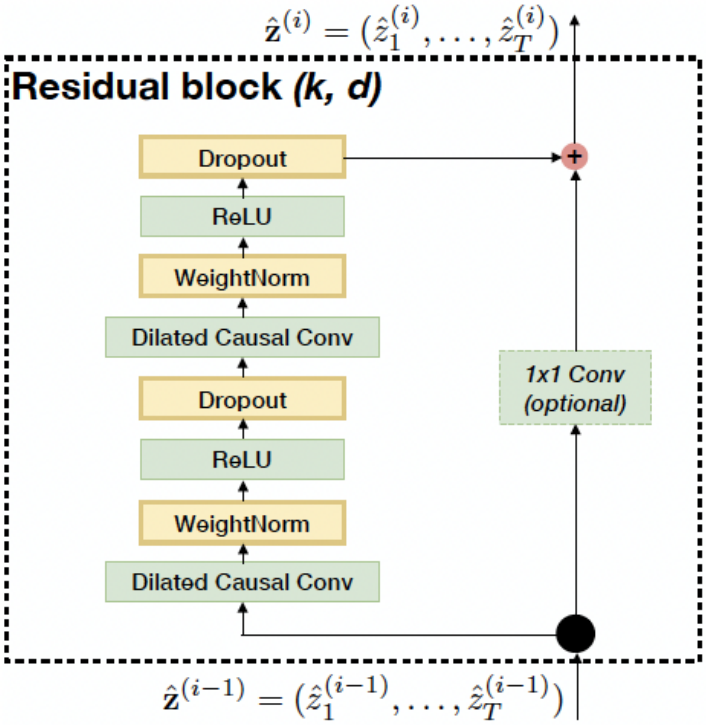


Figure 2-4: TCN block [45]

Each TCN block applies a residual connection from the input to the output which is experimentally proven to benefit for networks with very deep layers, where a 1-D convolution is optionally used in between to match the shape of input to the output. Within each TCN block, two pairs of a dilated casual convolution and a ReLU nonlinearity are used, and weight normalization and dropout are added for normalization and regularization, respectively. Additionally, zero-

padding is always applied to ensure each hidden layer has the same length as input layer.

There are several advantages using the TCN: The structure incorporating casual convolutions can avoid information leakage from the past of the sequence. Second, TCN can output sequence of any length identical to that of the input as the RNN does, but TCN processes all input sequences in parallel unlike sequential processing in RNN, which means the TCN is more efficient compared with basic RNN structure. On the other hand, this parallelism enables TCN have lower requirement of training memory because it does not have to save partial results while training.

Dilated convolution is an efficient alternative of conventional convolution to obtain the output that is able to extract broader range of feature representations from the input. It utilizes inflated kernels with spaces among elements, expanding the receptive field to extract more features with fewer convolutional layers. The rate of dilation is commonly set as an exponential increasing trend. Fig. 2-5 shows the process of the general convolution and the dilated convolution with one-dimensional sequence input. Compared with the conventional convolution, a dilated convolution enlarges the receptive field from K to $(K-1)\times(R-1)+K$, where K is kernel size and R is dilation rate [46].

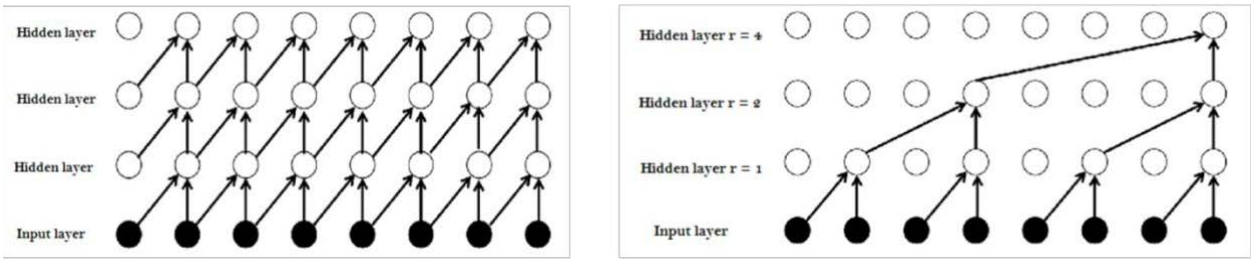


Figure 2-5: An illustration of conventional convolution and dilated convolutions with exponentially increasing rate [43]

Inspired by TCN and dilated convolution described above, the authors of [47] proposed a temporal convolutional neural network (TCNN) for speech enhancement, which combines the U-Net structure and TCN block. As shown in Fig. 2-6, this TCNN includes an encoder, an decoder and a TCM, where the TCM consists of multiple TCN blocks. In TCNN, the encoder takes the noisy speech waveform as input and decreases the input frame size from 320 to 4 while increases channel dimension from 1 to 64 by using 2D-convolutional layers, thus generating the two-dimensional signal of size (4, 64). The output of encoder is then reshaped to a one-dimensional signal, which will pass through the TCM to learn the long-range dependency. Different from [45],

authors in [47] constructed TCN blocks using a 1×1 input convolution, 1D-dilated convolution, output 1×1 convolution and residual connection from inputs. And also, the parametric ReLU (PReLU) and batch normalization are utilized to replace the ReLU and weight norm, respectively, for better performance.

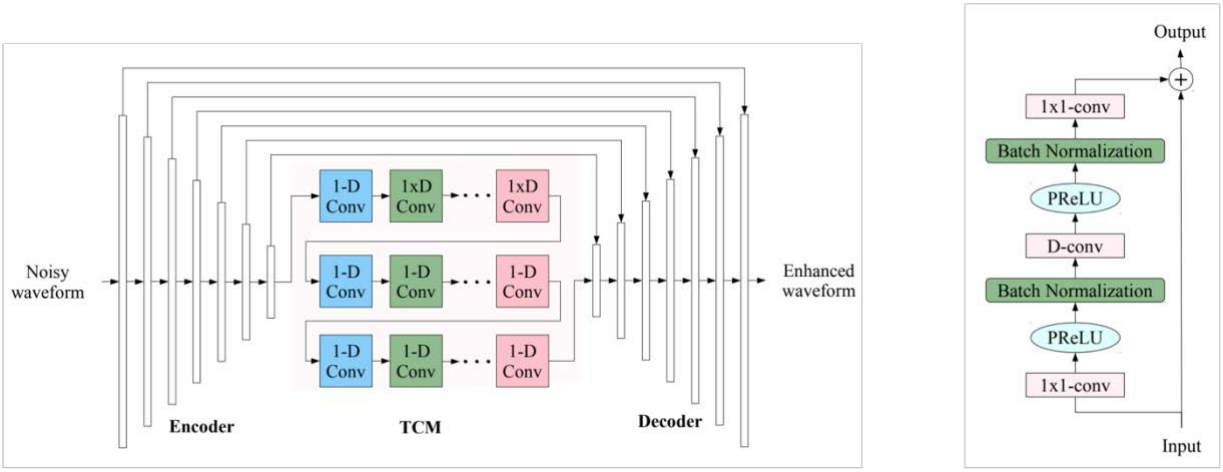


Figure 2-6: Temporal convolutional neural network [47]

2.1.3 U-Net with attention mechanism for speech enhancement

Since attention mechanism has been proven highly efficient for modeling sequences, recent methods in a variety of fields including the speech enhancement incorporates an attention component in previous neural networks. For the U-Net, generally, there are three ways of employing attention: combining attention with skip connections, inserting attention between the encoder and the decoder, and incorporating attention to encoder-decoder layers of the U-Net.

The attention Wave-U-Net (Fig. 2-7) is modified from the Wave-U-Net by implementing an attention mechanism on the skip connections, which help decoder layers efficiently reuse the important information from corresponding encoder layers [48]. In other words, the attention mechanism incorporated into skip connection can help decoder to learn which features from encoder are useful instead of using all of them.

As shown in Fig. 2-8, the attention block in attention Wave-U-Net aims to generate an attention mask to filter the redundant information from encoder layers, which produces more important information for decoder layers. Two steps of computations are involved to obtain an attention mask.

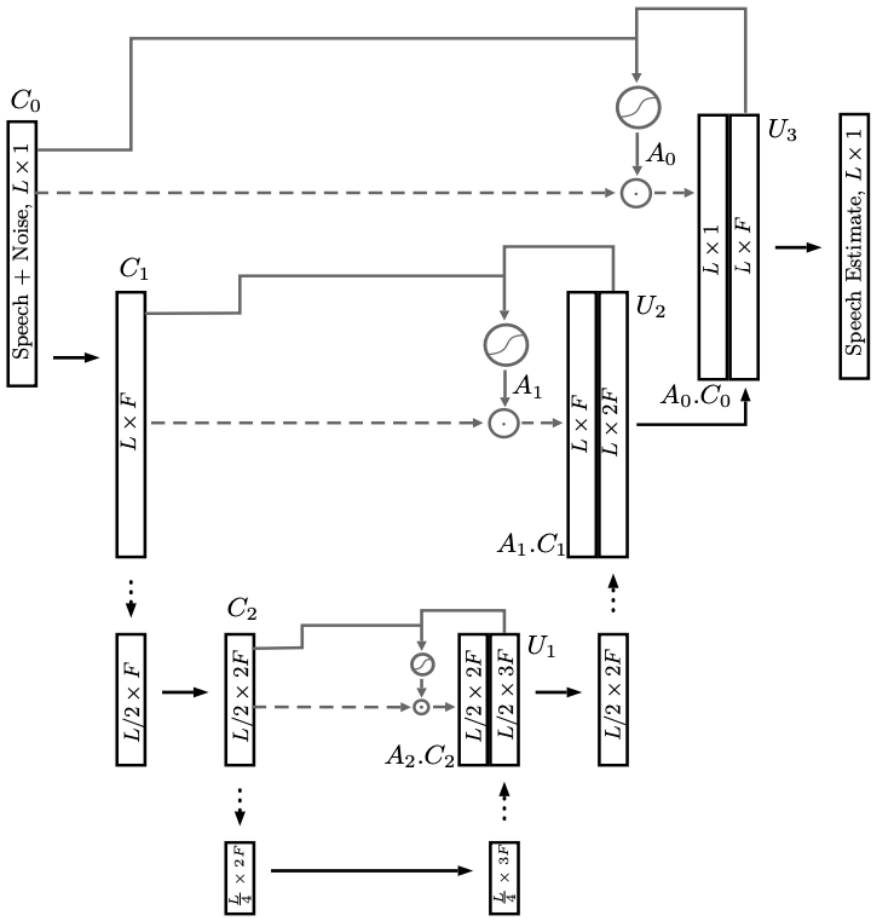


Figure 2-7: Attention Wave-U-Net [45]

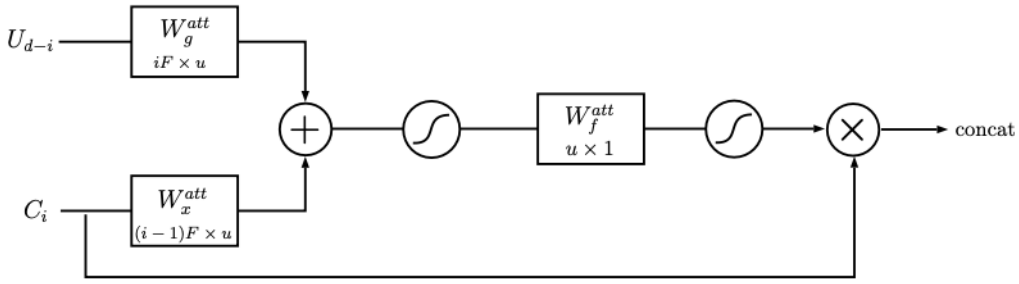


Figure 2-8: Attention mechanism in attention Wave-U-Net [48]

The first step is to sum up information from down-sampling layer and corresponding up-sampling layer, and the second step is to conduct a convolution with a kernel of size 1 on the outputs of the first step. The two steps are followed by an activation function. The whole attention method can be expressed as:

$$S_1 = \sigma(W_1 C + W_2 U + b_1) \tag{2-1}$$

$$S_2 = \sigma(W_3 S_1 + b_2) \tag{2-2}$$

$$O = C \odot S_2 \tag{2-3}$$

where S_1, S_2 and O denote the outputs of two steps and the concatenated outputs, respectively. C and U are the down-sampling layer and the corresponding up-sampling layer.

The authors of [49] proposed a nested U-Net with self-attention for speech enhancement, which introduces the self-attention mechanism into encoder and decoder of U-Net to further extract contextual information as shown in Fig. 2-9. The detailed illustration of the attention mechanism of this model is shown in the left side of the Fig. 2-9.

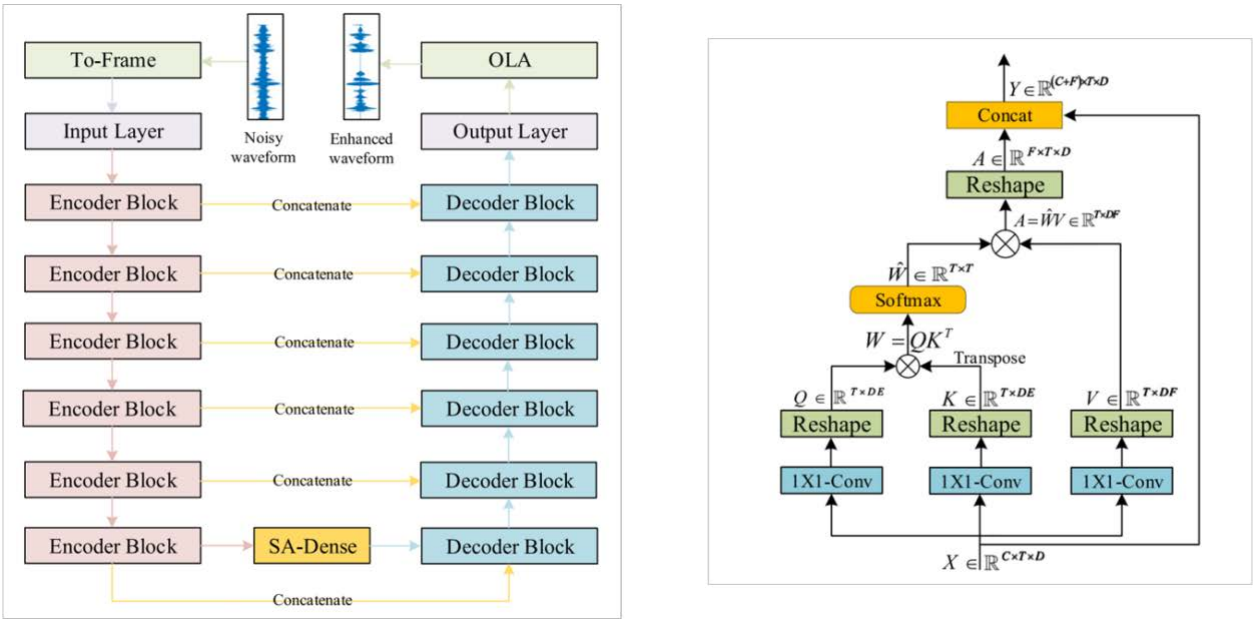


Figure 2-9: Nested U-Net with attention for speech enhancement [49]

The self-attention here is characterized by the input X and three matrices Q, K, V transformed from X by using three one-by-one convolutions. Then, the matrix Q is multiplied with the transposed version of matrix K, yielding the attention matrix which passes through the softmax function for generating the attention scores ranging from 0 to 1. Finally, the outputs of self-attention are obtained by the concatenation between inputs of the self-attention and multiplied results of attention scores and matrix V. The procedures can be formulated as follows:

$$A = Softmax(QK^T)V \tag{2-4}$$

$$Y = Concatenate(X, Reshape(A)) \tag{2-5}$$

where A and Y denote the self-attention score and final output of the attention block.

Different from two mentioned attention-based SE models, authors of [50] proposed a dense convolutional network (DCN) with self-attention for speech enhancement shown in Fig. 2-10, where the self-attention mechanism is introduced in encoder and decoder layers of U-Net. Each layer in the encoder and decoder is composed of a self-attention block and dense block, where the self-attention is adopted for extract global information and dense block is used to aggregate features of previous layers.

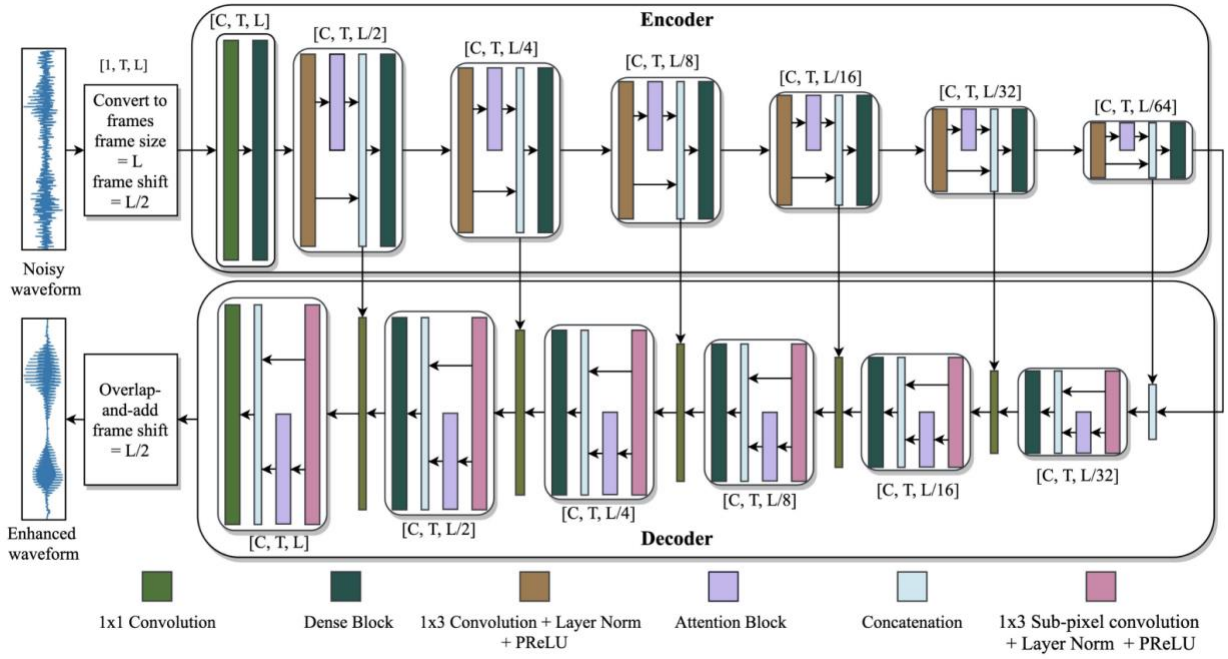


Figure 2-10: Dense convolutional network with attention for speech enhancement [50]

In addition, the dense block is built based on densely connected convolutional networks recently proposed in [51]. The densely connected convolutional network is designed to reuse the features at the current layer multiple times for the subsequent layers. In other words, a layer of dense block takes the input of dense block and the outputs from the previous layers. This dense connection can strengthen and reuse feature propagation. Meanwhile, it alleviates the vanishing gradient problem as CNNs become increasingly deep. An illustrative diagram of the dense block is shown in Fig. 2-11.

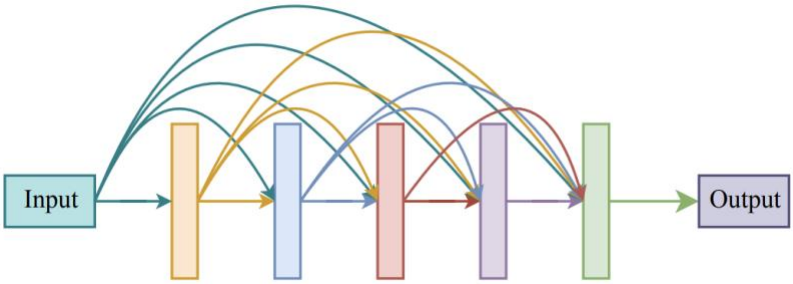


Figure 2-11: Dense block [51]

2.2 Proposed dual-branch attention U-Net

In this section, we propose a dual-branch attention U-Net (DBAUNet) for speech enhancement. As shown in Fig. 2-12, the proposed model consists of an encoder, a dual-branch attention block and a decoder. Moreover, a preprocessing unit is set before input feeding into the encoder and a postprocessing operation is set after decoder to reconstruct the enhanced speech as shown in Fig. 2-13.

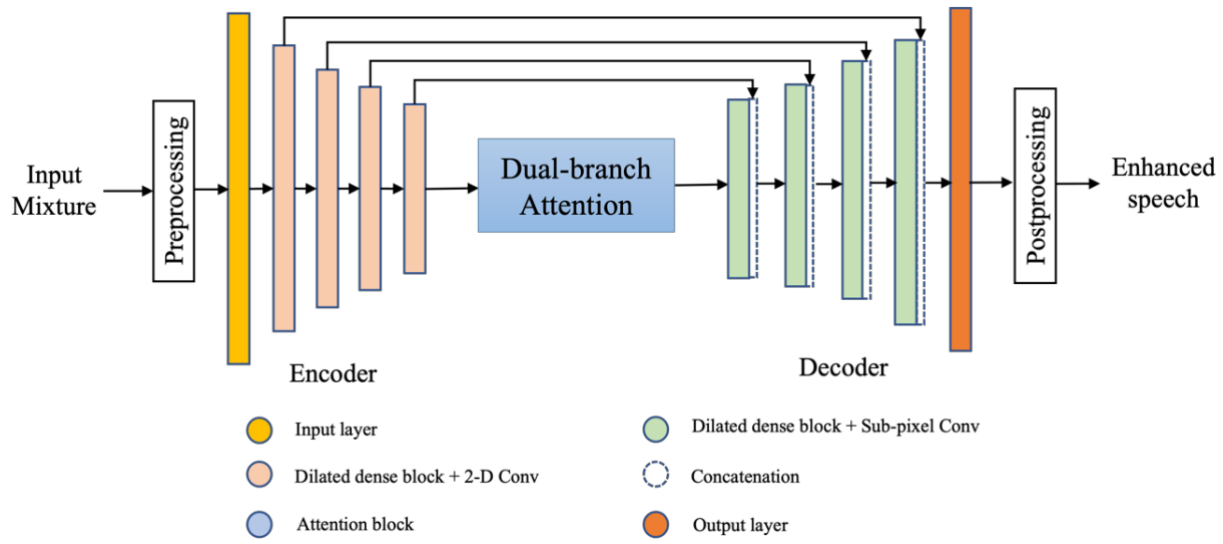


Figure 2-12: Proposed DBAUNet

2.2.1 Preprocessing and postprocessing

The preprocessing stage is to process the input signal before it is fed into the neural network. The input signal is a mixture of noisy utterances, which is cut into small frames with appropriate overlap of each two adjacent frames. The postprocessing before reconstruction of speech is an inverse process of the preprocessing to recover frames back to utterances with their original lengths.

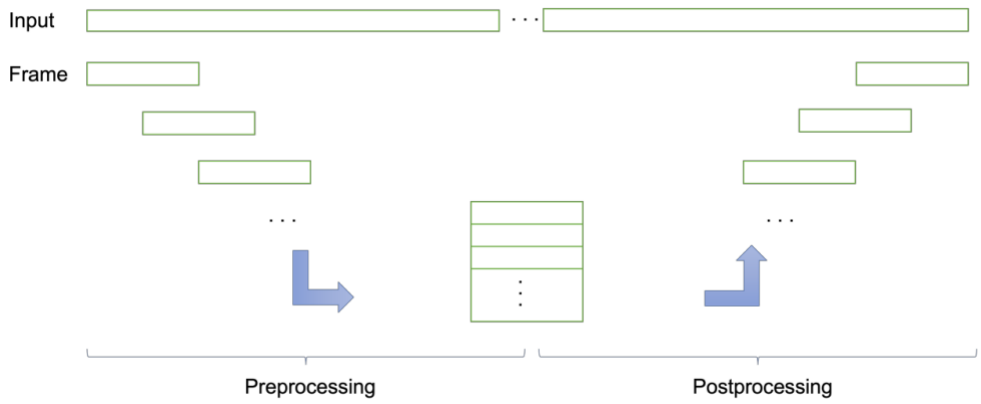


Figure 2-13: Preprocessing and postprocessing stage

More specifically, preprocessing stage splits an original speech mixture $M^{1\times L}$ into overlapped K frames as shown in Fig. 2-13, where each frame has the length of S and adjacent frames have a hop size of H. Then all the frames are packed together to form a 3D tensor $I \in \mathbb{R}^{1\times K\times S}$ if considering channel dimension. Here K is defined as:

$$K = \lceil (L-S)/(S-H) + 1 \rceil \tag{2-6}$$

where L is the length of original speech mixture, the operator $\lceil \cdot \rceil$ means rounding up to the nearest integer. Frames with smaller size than S will be padded with zero to match the equal frame size. Inversely, the postprocessing is adopted to recover the enhanced waveform from 3D tensor.

2.2.2 Encoder

The encoder is comprised of 5 layers, containing an input layer and 4 down-sampling layers. each down-sampling layer consists of a dilated-dense block and a 2-D convolution. The dilated-dense block is the combination of dense connection and dilated 2-D convolution. Different from the

densely connected convolutional network. All the conventional convolutions in dense block are replaced by dilated convolutions.

Dilated convolution is used to increase the receptive field of CNN, which has been proved to be an efficient alternative to recurrent neural networks (RNNs) for modeling long-range sequences. An illustrative diagram of the dilated dense block is shown in Fig. 2-14.

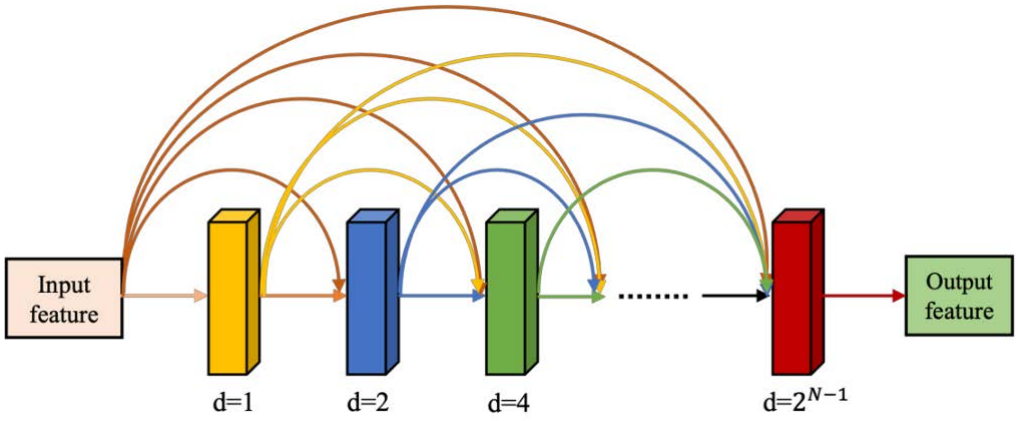


Figure 2-14: Dilated dense block

Each dilated-dense block includes N layers of 2-dimensional depthwise separable convolutions with exponentially increasing dilation rates 1, 2, 4, …2^N as shown in Fig. 2-14. Each convolution layer is followed by the layer normalization [37] and parametric ReLU (PReLU) nonlinearity. For the encoder processing in the model, the first layer uses 2-D convolution with 64 filters of size (1, 1) to increase the number of channels of input mixture from 1 to 64.

In each down-sampling layer, the dilated-dense block has 4 dilated convolution layers where each one uses filter of size (2, 3) with 64 output channels to keep its output in the same shape as the input. Then, the dilated-dense block is followed by a 2-D convolution with filters of size (1, 3) and a stride of (1, 2) which halves last feature dimension. All convolutions are followed by the layer normalization and PReLU non-linearity. After 4 down-sampling layers, the last dimension of feature representation will have 16 times reduction compared with the input of down-sampling layers.

To further increase the receptive field as well as decrease parameters of convolutional neural network, the depthwise separable convolution is applied in the dilated-dense block. It generally includes two parts: depthwise (DW) and pointwise (PW) to extract feature maps as shown in Fig.

2-15 for example. For DW convolution, each channel input is convolved with one kernel separately and the number of feature maps is the same as channels. Here the number of parameters in DW step is computed by multiplying the number of kernels and the kernel size, which is 27 in this example. The PW convolution uses kernels with size of (1, 1) to convolve along the depth of feature maps from DW convolution to produce new feature maps, which is the final output of the depthwise separable convolution.

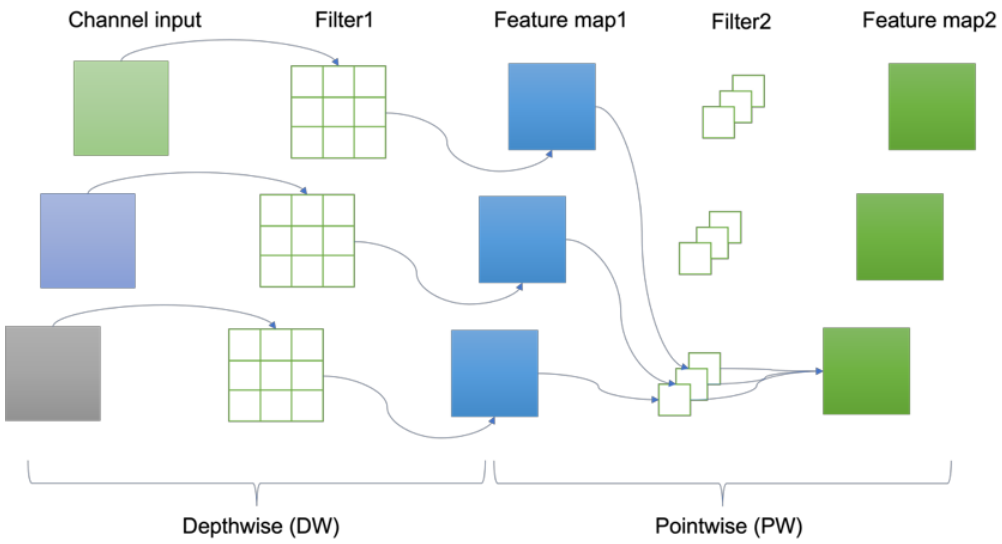


Figure 2-15: Depthwise separable convolution

The number of parameters generated by the DW step in Fig. 2-15 is 9. The total number of the parameters in the depthwise separable convolution is the parameters from PW and DW steps, which is 36 in this example. However, conventional convolution needs to use a group of three-kernel filters and create 81 parameters, which is much more than that of the depthwise separable convolution.

2.2.3 Dual-branch attention block

The dual-branch attention block includes two parts: spatial attention block and channel attention block, as shown in Fig. 2-16. Input features are simultaneously processed by these two attention blocks and the final output of the self-attention module will be obtained by fusing the two-stage outputs from two attention blocks.

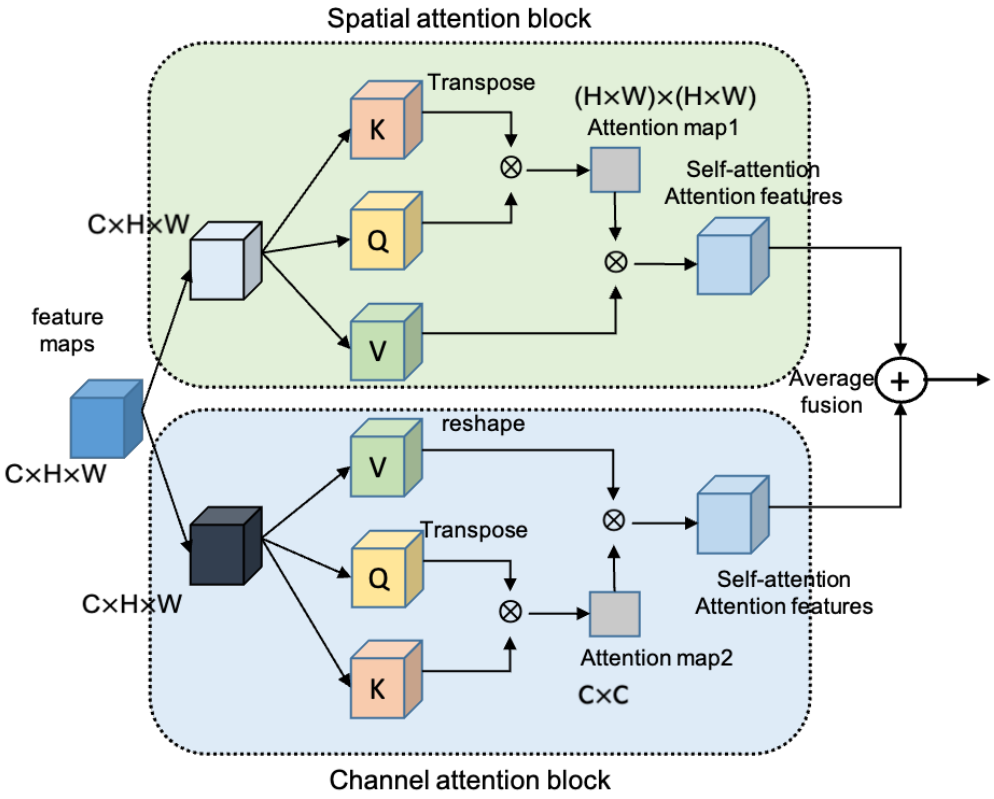


Figure 2-16: Proposed DBAB

The spatial attention block is proposed to extract spatial information of speech features, which focuses on informative spatial region. In the block, input feature $X \in \mathbb{R}^{C\times H\times W}$ is transferred to three same-shaped matrices K, Q, V, where C, H, and W are channel, height, and width respectively. The attention map 1 is the multiplication of Q and transposed K, with a shape of $[H \times W, H \times W]$, where the original channel dimension of input features is removed and information along height and width is incorporated, which is so-called spatial attention block. The calculation procedures can be expressed as:

$$Q = Conv^{q}_{1\times1}(X), \quad K = Conv^{k}_{1\times1}(X), \quad V = Conv^{v}_{1\times1}(X) \tag{2-7}$$

$$SpatialAttention = Softmax(QK^{T})V \tag{2-8}$$

where $Conv^{q}_{1\times1}(\cdot), Conv^{k}_{1\times1}(\cdot), Conv^{v}_{1\times1}(\cdot)$ denote three different one-by-one convolutions for obtaining three matrices $Q, K\ and\ V$ of spatial branch respectively.

The channel attention block is proposed to extract channel information of speech features, which focuses on what is important among channel features. In channel attention block, Q, K, and V are created in the same way as that of spatial attention block, generating the attention map 2 with

a shape of $[C,\ C]$ which is computed on the transposed Q and K, where the channel information of input features is exclusively collected after passing through the channel attention block. The details are shown as follows:

$$Q = Conv_{1\times1}^{q}(X), \quad K = Conv_{1\times1}^{k}(X), \quad V = Conv_{1\times1}^{v}(X) \tag{2-9}$$

$$ChannelAttention = Softmax(KQ^{T})V \tag{2-10}$$

where $Conv_{1\times1}^{q}(\cdot), Conv_{1\times1}^{k}(\cdot), Conv_{1\times1}^{v}(\cdot)$ denote three different one-by-one convolution for obtaining three matrices $Q, K\ and\ V$ of channel branch respectively. To obtain the final output of dual-branch attention block, the spatial-wise features from spatial branch and channel-wise features from channel branch are fused by the average operation along with the residual connection from inputs, which is shown as follows:

$$Output = (SpatialAttention + ChannelAttention)\ /\ 2\ + X \tag{2-11}$$

2.2.4 Decoder

The decoder has a symmetric representation to the encoder, which processes the feature representation from attention block to reconstruct the enhanced speech waveform. There are 4 up-sampling layers where each one includes dilated-dense block, a sub-pixel convolution and the concatenation of output from corresponding symmetric encoder layer. The dilated-dense block is the same as the corresponding one in the encoder layer. Finally, the output layer uses convolution with filter of size (1, 1) to decrease the number of the channels of feature from 64 to 1, in order to obtain the enhanced speech waveform through postprocessing.

The sub-pixel convolution is used as an efficient alternative of the transposed convolution to double the last dimension of feature representation without introducing checkerboard artifacts [52]. It is comprised of a general convolution to increase the number of output channels and a periodic shuffling. It is initially used for image processing to generate high-resolution images from low-resolution ones as shown in Fig. 2-17.

Speech enhancement on spectrograms using sub-pixel convolution for up-sampling is similar to the operation of increasing image resolution. With respect to the one-dimensional waveform implementation, as Fig. 2-18 depicts, the input first passes through a standard convolution to increase the channels with the upscale rate of up-sampling, which will be shuffled to form features with the number and the size of channels being the same as those of the input channels. There is

no extra information introduced during sub-pixel convolution, thus leading to a better reconstruction of the enhanced speech.

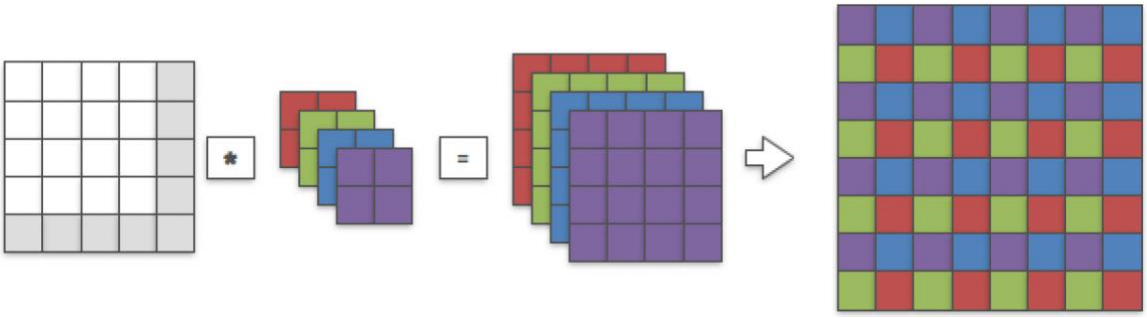


Figure 2-17: An illustration of sub-pixel convolution to generate high-resolution output [52]

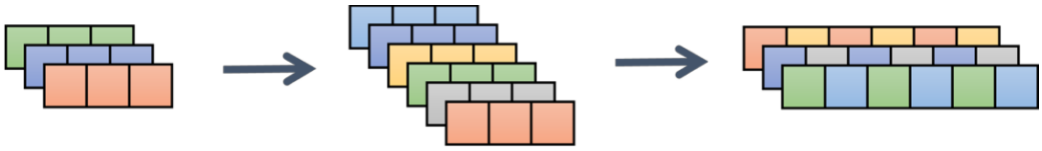


Figure 2-18: An illustrative of 1-D sub-pixel convolution

2.3 Experiments

2.3.1 Experiment setup

The experiments are conducted on the VCTK dataset proposed by Valentini et al. [27], which is selected from Voice Bank corpus [28]. This dataset has been widely adopted to verify deep learning-based speech enhancement methods in recent years, since both training set and testing set are provided under given noise conditions, which very much facilitates users' experimental studies with comparison to state-of-the-art methods. The training set includes 11572 utterances of 28 speakers (14 female and 14 male) with SNR levels of 15 dB, 10 dB, 5 dB and 0 dB, where 8 types of noises come from DEMAND dataset [29] and 2 noises are artificially generated. The testing set has 824 utterances of 2 speakers (one male and one female) with 5 types of unseen noises at SNRs of 17.5 dB, 12.5 dB, 7.5 dB and 2.5 dB.

In order for the neural network to suitably deal with the dataset, all utterances are resampled at 16kHz and cut into 4-second pieces. For those small sequences shorter than 4 seconds, zero-padding is applied to match the same length. The window size of 32 ms (512 samples) is used at preprocessing stage to obtain chunks of 50% overlap. We use the Adam optimizer to train our model for 100 epochs. The weight decay is used during the optimization to avoid gradient vanishing, which is set as $1e^{-7}$ in our experiments.

The loss function used here is a combination of time domain loss and frequency domain loss. The frequency-domain loss can supervise the model to learn more information, leading to higher speech intelligibility and perceptual quality [52], which is defined as:

$$L_F = \frac{1}{TF}\sum_{t=0}^{T-1}\sum_{f=0}^{F-1}\left[\left(|S_r(t,f)|+|S_i(t,f)|\right) - \left(\left|\hat{S}_r(t,f)\right|+\left|\hat{S}_i(t,f)\right|\right)\right] \tag{2-12}$$

where S and $\hat{S}$ denote the clean spectrogram and the enhanced estimate of spectrogram. r and i are the real and imaginary parts of the complex variable, T is the number of frames and F is the number of frequency bins. For computing the loss in time domain, we use mean square error (MSE) between the enhanced speech waveform and clean speech waveform, which is defined as:

$$L_T = \frac{1}{N}\sum_{i=0}^{N-1}\left(s_i - \hat{s}_i\right)^2 \tag{2-13}$$

where s_i and $\hat{s}_i$ are the $i\text{-}th$ sample of clean speech waveform and the denoised speech waveform respectively, and N denotes the number of samples in an utterance. The final loss function combines these two types of losses mentioned above, giving:

$$L = \lambda * L_F + (1-\lambda)L_T \tag{2-14}$$

where λ is a tunable parameter, which is set as 0.2 in the experiments.

2.3.2 Comparison with existing methods

Table 2-1 shows the comparison results of the proposed DBAUNet and some existing methods evaluated in the same dataset with same evaluation criteria, where all scores of baseline models in the table are given by their original papers and values of the proposed model is from our conducted experiment. First, the proposed DBAUNet has an obvious improvement in PESQ value from 1.97 to 2.84. which has a better performance than most baseline models while having the lowest model parameter with only 0.66 million. In view of speech intelligibility performance, the DBAUNet achieves the best STOI score (94%) compared with all other baseline models. Although DBAUNet

obtains a slightly inferior PESQ (2.84) than DCUNet (2.93) and MetricGAN (2.86), it involves only 2.5 times fewer trainable parameters than DCUNet (2.30 million parameters).

Besides, our proposed DBAUNet achieves comparable performance in three MOS evaluation metrics with existing baselines, where DCUNet using complex neural network has best CBAK and COVL scores but worse CSIG than DBAUNet. In terms of SSNR performance, the DCUNet gets the best score. A possible reason is that it utilizes the complex inputs (magnitude and phase) and complex CNN to simultaneously estimate the magnitude and phase information of clean speech. Meanwhile, DCUNet involves much more trainable parameters.

Table 2-1: Evaluation scores of the proposed model and some existing models

Model	PESQ	STOI	CSIG	CBAK	COVL	SSNR	Para. (Million)
Unprocessed	1.97	0.91	3.34	2.44	2.63	1.73	-
Wiener	2.22	-	3.23	2.68	2.67	5.07	-
SEGAN [53]	2.16	0.93	3.48	2.94	2.80	7.73	97.47
WaveNet [8]	-	-	3.62	3.23	2.98	-	-
CNN-GAN [54]	2.34	0.93	3.55	2.95	2.92	-	-
Wave U-Net [41]	2.40	-	3.52	3.24	2.96	9.97	10.00
Att. WU-Net [48]	2.62	-	3.91	3.35	3.27	-	-
MMSE-GAN [55]	2.53	0.93	3.80	3.12	3.14	-	-
MetricGAN [56]	2.86	-	3.99	3.18	3.42	-	-
DCUNet [57]	**2.93**	-	4.10	**3.77**	**3.52**	**14.44**	2.30
SE-Flow [58]	2.43	-	3.77	3.12	3.09	8.07	-
CP-GAN [59]	2.64	0.94	3.93	3.29	3.28	18.10	-
SASE [60]	2.76	0.94	4.09	3.32	3.43	-	10.35
SADNUNet [49]	2.82	-	**4.18**	3.47	3.51	-	2.63
DBAUNet	2.84	**0.94**	4.14	3.47	3.50	9.25	**0.66**

Second, we further compare the proposed DBAUNet with some U-Net based models with or without attention mechanism to demonstrate the effectiveness of our proposed dual-branch attention in speech enhancement, including Wave U-Net, Attention Wave U-Net, SASE and SADNUNet. As introduced before, the Wave U-Net is 1-D convolutional encoder-decoder

structure to estimate the enhanced waveform from noisy speech waveform. Attention Wave U-Net applies the attention mechanism in skip connection of U-Net to make important information from encoder to be used by corresponding decoder for better speech reconstruction. SADNUNet inserts the dense block and self-attention block between encoder and decoder of U-Net to further capture long-range dependency of speech sequences. Similar to SADNUNet, SASE incorporates the LSTM layer and self-attention block into U-Net encoder and decoder to extract high-level speech features.

From Table 2-1, we can observe that our proposed DBAUNet achieves superior performance in terms of PESQ and STOI to U-Net based models while having fewest trainable parameters. Specifically, DBAUNet has best PESQ improvement compared with Wave U-Net (2.40), Attention Wave U-Net (2.62), SASE (2.76) and SADNUNet (2.82). Wave U-Net without attention mechanism obtains the worst PESQ score among other U-Net based models, which indicates that introducing attention mechanism is relatively helpful for improving the performance of U-Net based denoising models. SADNUNet performs slightly better than proposed DBAUNet in terms of CBAK, however, it includes much more parameters (2.63 million). For other MOS metrics, DBAUNet shows an impressive improvement compared with other models.

By comparing with existing speech enhancement models, it is found in general that our proposed DBAUNet achieves a comparable performance in modeling long-range speech sequences while having lowest model complexity. The main reason is that our DBAUNet adopts two-branch attention block to parallelly extract spatial and channel information, which are finally packed together to obtain the efficient feature representations. We will carry out a further study below on the efficiency of the proposed two-branch attention in our model.

2.3.3 Ablation study

In our model, the dual-branch attention block (DBAB) is proposed to extract spatial information and channel information at the same time, leading to spatial-wise and channel-wise feature representation. To further explore the influence of proposed DBAB on speech enhancement, we design two reference models for comparison by removing spatial branch attention or channel branch attention or both of them, yielding Spatial-UNet, Channel-UNet and Pure U-Net as shown in Table 2-2.

Table 2-2: Experimental results of different configuration in proposed model

Model	PESQ	STOI	CSIG	CBAK	COVL	SSNR	Para. (Million)
Unprocessed	1.97	0.91	3.34	2.44	2.63	1.73	-
Pure U-Net	2.54	0.94	3.85	3.26	3.20	8.91	**0.63**
Channel-UNet	2.73	0.94	4.11	3.37	3.43	9.16	0.65
Spatial-UNet	2.72	0.94	4.07	3.38	3.40	**9.30**	0.64
DBAUNet	**2.84**	**0.94**	**4.14**	**3.43**	**3.50**	9.25	0.66

As indicated in Table 2-2, omitting spatial branch attention or channel branch attention would result in the performance reduction compared with DBAUNet containing both spatial and channel branch attention, showing that both spatial and channel attention are important for extracting abundant features of long-range speech sequences. Moreover, the Pure U-Net without any attention mechanism obtains the worst performance of speech enhancement compared with other comparison models, confirming that introducing attention mechanism is indispensable for improving the performance of U-Net on long-range sequence modeling.

Additionally, in order to demonstrate the efficiency of the proposed DBAB, we explore two different blocks inserted between encoder and decoder of U-Net, including LSTM and temporal convolutional network (TCN) blocks. The LSTM and TCN are commonly used in sequence modeling, which has been introduced in previous chapter. To simplify the experiment, we use 2 LSTM layers and 2 groups of TCN blocks where each group has 6 dilated convolutional layers as the module between encoder and decoder. These two reference models are named as LSTM-UNet and TCN-UNet, respectively, as shown in Table 2-3.

Table 2-3: Performance of different comparison model

Model	PESQ	STOI	CSIG	CBAK	COVL	SSNR	Para. (Million)
Unprocessed	1.97	0.91	3.34	2.44	2.63	1.73	-
Pure U-Net	2.54	0.94	3.85	3.26	3.20	8.91	**0.63**
LSTM-UNet	2.67	0.94	3.92	3.33	3.30	9.21	0.64
TCN-UNet	2.55	0.94	3.81	3.27	3.17	9.20	2.02
DBAUNet	**2.84**	**0.94**	**4.14**	**3.43**	**3.50**	9.25	0.66

From Table 2-3, we found that inserting LSTM, TCN or DUAB would have different degrees of improvement on U-Net based speech enhancement. Especially, the proposed DBAUNet achieves a superior performance in all evaluation metrics compared with reference models using LSTM and TCN, indicating that the proposed DBAB not only works well on long-range speech sequences but also extracts more efficient contextual feature by learning both spatial and channel information with dual-branch structure.

2.4 Summary

A U-Net with a dual-branch attention block incorporated, named DBAUNet, has been proposed in this chapter. The encoder layer of DBAUNet is comprised of a dilated dense block and a two-dimensional convolution performing the down-sampling operation. The dilated dense block consists of multiple densely connected dilated convolutional layers with exponentially increasing dilated rates and the convolution here adopts the depthwise separable convolution. The decoder layer of DBAUNet comprises of a dilated dense block and a sub-pixel convolution as the up-sampling. The dilated convolution is used to expand receptive fields and the depthwise separable convolution is a light-complexity alternative of the conventional convolution. The sub-pixel used as up-sampling approach can avoid checkboard artifacts which is introduced by common convolutions for up-sampling operation.

The dual-branch attention block in the DBAUNet contains a spatial-branch attention along the height-width dimension and a channel-branch attention on channel dimension respectively, and then the features from the two-branch attentions are fused through average computation.

Experimental results on a benchmark dataset have shown that the DBAUNet achieves comparable performance to most of the comparison models in most evaluation metrics and holds the lightest model complexity. Additionally, U-Nets with LSTM, TCN and one-path attention are also investigated to demonstrate the efficiency of the proposed two-branch attention block, demonstrating that the proposed U-Net structure combined with proposed two-branch attention outperforms other U-Nets with different components and still keeps a competitively low model complexity.

Chapter 3
Proposed Multi-head Attention U-Nets for Speech Enhancement

In this chapter, we propose two efficient U-Net architectures with multi-head attention mechanism for speech enhancement, including multi-head attention U-Net-1 (MHAUNet-1) and multi-head attention U-Net-2 (MHAUNet-2). The MHAUNet-1 incorporates proposed attention blocks between encoder and decoder to further extract the contextual information of the encoder outputs, and MHAUNet-2 applies attention mechanism in each encoder and decoder layer to learn the global information of speech features. This chapter is organized as follows, the multi-head attention mechanism is first introduced, and some existing works using multi-head attention mechanism for speech enhancement are reviewed in section 3.1. Then, we present the proposed MHAUNet-1 and MHAUNet-2 for time-domain speech enhancement in section 3.2. Finally, section 3.3 provides the experimental results of the proposed MHAUNet-1 and MHAUNet-2.

3.1 Previous work

3.1.1 The introduction of multi-head attention

Attention mechanism is proposed to focus on extracting the information or element that is most important for performance. Different from the self-attention mechanism mentioned above only considering the interested information in one aspect, the multi-head attention provides multiple attention operations simultaneously for the neural network to extract features in multiple representation subspaces [61]. Then, the extracted features of each representation subspace are packed together to constitute the final feature representation of multi-head attention. In other words, the multi-head attention is able to learn the information in different aspects, which performs a similar procedure to the convolutional neural network using different convolutional kernels to learn the features from different aspects.

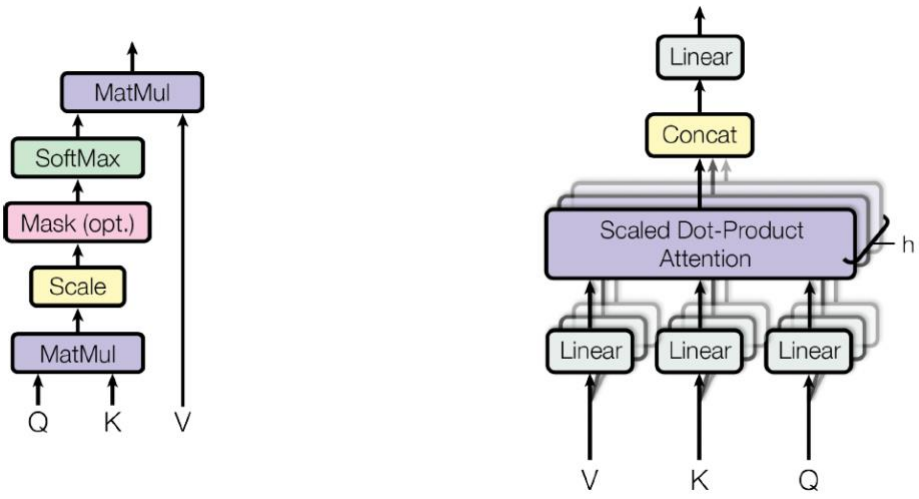


Figure 3-1: Scaled dot-product attention and multi-head attention [61]

In the multi-head attention shown in Fig. 3-1, the input features are first passed through three different learnable linear transformations with h times to obtain h representation subspaces, where each one has three different feature representation queries and keys with feature dimension d_q, and values with feature dimension d_q. Then, the query, key and value in each feature subspace performs the scaled dot-product attention in parallel, which is one-head attention. In terms of the scaled dot-product attention described in Fig. 3-1, the query matrix is multiplied with the transposed version of key matrix, whose result is divided by the scaled factor $\sqrt{d_q}$ for the sake of normalization. Next, the softmax operation is employed to the scaled dot product to obtain the attention with values ranging from 0 to 1. Finally, the one-head attention is computed by multiplying the attention matrix with value matrix. Multiple one-head attentions are computed with a parallel procedure, which are eventually concatenated and projected again with a learnable linear layer to generate the outputs of multi-head attention. The multi-head attention forces the model to learn the feature representation from a different perspective. The whole procedures can be formulated as follows:

$$Q_i = XW_i^Q, K_i = XW_i^K, V_i = XW_i^V \tag{3-1}$$

$$head_i = Attention(Q_i, K_i, V_i) = softmax(\frac{Q_i {K_i}^T}{\sqrt{d_q}})V_i \tag{3-2}$$

$$MultiHead(Q, K, V) = Concact(head_1, \dots, head_i, \dots)W^o \tag{3-3}$$

where X denotes the input features with dimension d_{model}, Q, K, V stand for the queries, keys

and values, W_i^Q, W_i^K, W_i^V denote the ith linear transformation matrix for queries, keys and values, respectively, and $i = 1, 2, \ldots, h$, and h is the number of heads, W^O is the linear transformation matrix for obtaining the outputs of multi-head attention.

3.1.2 Multi-head attention based models

Inspired by the effectiveness of multi-head attention mechanism in modeling the long-range dependency of speech sequences, some multi-head attention based models have been proposed to improve the performance of speech enhancement. The authors of [62] proposed a LSTM-augmented multi-head attention block, which consists of a Local-LSTM, a multi-head attention and a 1-D convolution as shown in Fig. 3-2.

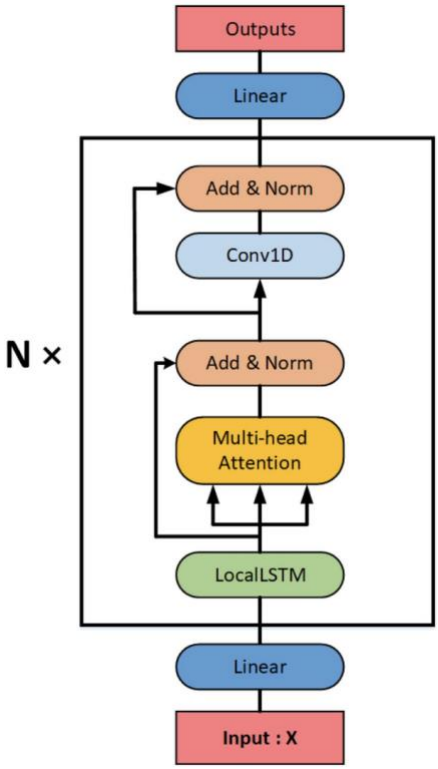


Figure 3-2: LSTM-augmented multi-head attention block [62]

The Local-LSTM is proposed to learn the local structure of speech features by adopting the LSTM into the short-term spectrogram, which can also bring the positional information for multi-

head attention mechanism to improve the performance of speech denoising. Then, the multi-head attention is used to learn the long-term relationship among speech features, generating the contextual feature representation by considering each position of features. Next, the 1-D convolutional neural network is employed to extract the local information between adjacent hidden states. Finally, the layer normalization and residual connection are used to boost the model training and avoid gradient vanishing problem.

Authors of [63] investigated self-adaptation models for speech enhancement by using multi-head attention mechanism as indicated in Fig. 3-3. This model has two branches, the CNN-branch applied to extract the spatial information of speech features, and the speaker-aware branch proposed to extract the auxiliary speaker-aware features. In the following, the outputs of two branches are concatenated to pass through a BLSTM for obtaining the time-independent speaker features. In this work, the multi-head attention block is used to extract the long-range dependencies of speech features, generating the contextual information which will be packed together with the time-independent speaker information to obtain the mask for denoising by following one linear layer. Additionally, the multi-task learning is used to improve the denoising performance by dealing with the speech enhancement and speaker identification tasks at the same time.

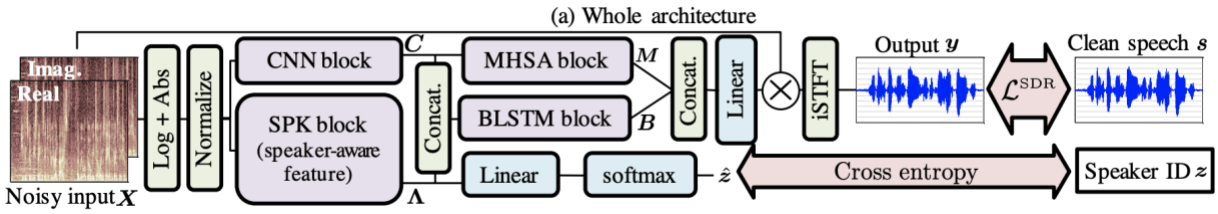


Figure 3-3: Self-adaptation model for speech enhancement using multi-head attention [63]

Recently, some works combined the multi-head attention mechanism with U-Net architecture to promote the performance of U-Net based models in speech enhancement. Authors of [64] incorporated the language model into speech enhancement framework as shown in Fig. 3-4, where the linguistic information from language model will benefit the speech enhancement during noisy environments. More specifically, the symbolic encoder is proposed to learn the linguistic information of spoken utterances, generating the discrete symbolic sequences to represent the high-level phoneme features of speech sequences by using vector quantized variational autoencoder algorithm. Then, the encoder-decoder based U-Net is employed to perform the speech denoising,

where the encoder is used to obtain the compressed features and decoder is applied to recover the intermediate features back to enhanced speech.

To make full use of linguistic information from symbolic sequences, the multi-head attention block is inserted before each decoder layer to connect its inputs with the symbolic information. More specifically, the multi-head attention block receives the features from encoder layer or previous decoder layer as the query as well as the features from symbolic sequence as the key and value, which builds the connection between speech characteristics and corresponding content information to reinforce the performance of speech enhancement.

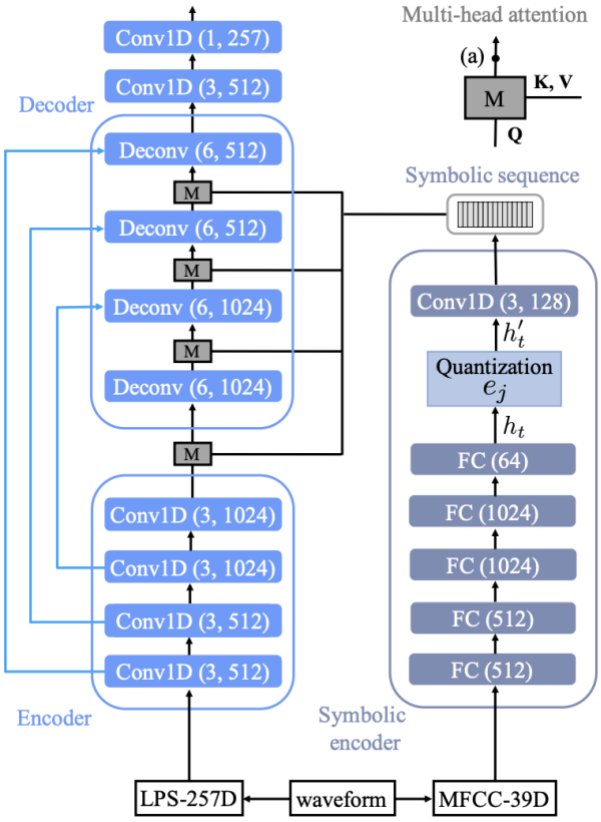


Figure 3-4: Symbolic sequential model with multi-head attention for speech enhancement [64]

Different from the work in [64], the authors of [65] investigated the multi-head attention method to connect information from encoder with the one from decoder in U-Net based speech enhancement model as shown in Fig. 3-5. First, the authors built a dual-path attention based multi-head attention as indicated in the left part of Fig. 3.6, where one is used to extract the temporal information of speech spectrogram along the time-axis dimension and the other one is applied to

capture the frequency information along the frequency-axis dimension. By using this dual-path attention block, the abundant information of speech is learnt for long-range speech modeling. Then, the cross-attention block is applied before each decoder layer to build a better skip connection as shown in the right part of Fig. 3-6, where the attention block regards the information of corresponding encoder layer as the key and value, and treats the features from decoder layer as the query.

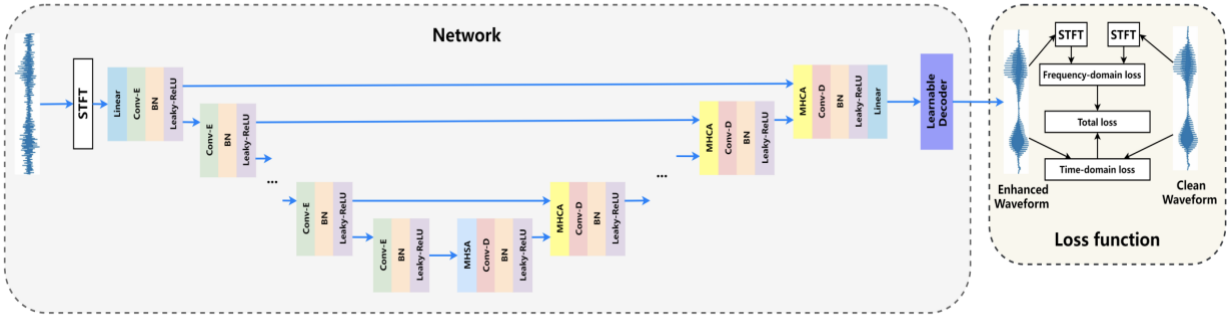


Figure 3-5: U-Net with multi-head self-attention and cross-attention [65]

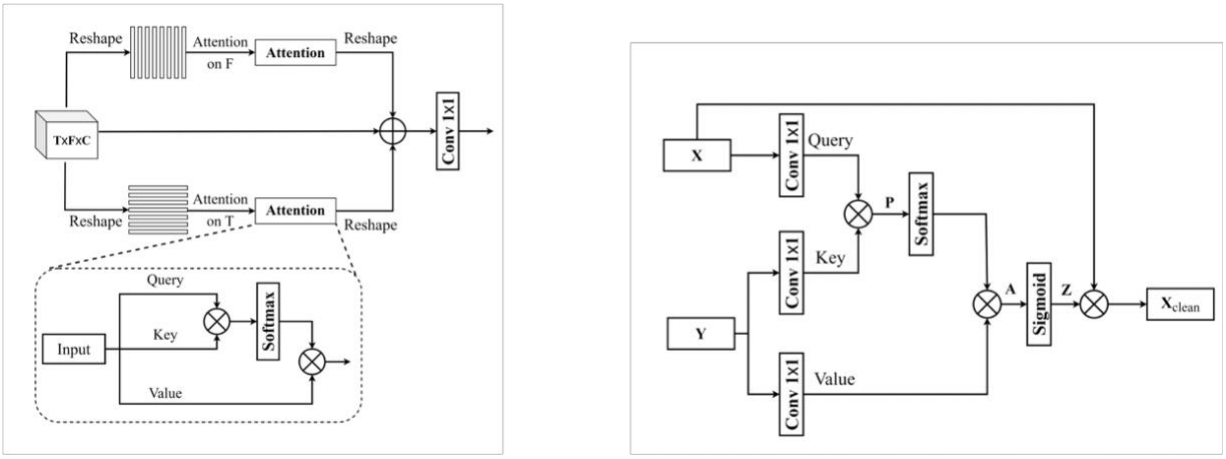


Figure 3-6: Multi-head self-attention and cross-attention [65]

Within the cross-attention block in this work, the query information is amended to the key information to obtain the attention score, which will be used to filter the encoder features. In other words, the proposed attention block is used to remove the redundant information of the encoder and only reuse the information which benefits the decoder to reconstruct the enhanced speech.

Finally, to utilize the time and frequency information for training the model, the mixed loss function is used based on the combination of time-domain loss and frequency-domain loss.

3.2. Proposed multi-head attention U-Nets

In this section, we propose two new U-Net based architectures for speech enhancement, which employ the multi-head attention to extract the contextual information of long-range speech sequences. The first proposed model inserts the multi-head attention block (MHAB) between the encoder and decoder to further extract high-level features from the encoder outputs, where the MHAB is comprised of a sample multi-head attention and a frame multi-head attention to successively capture short- and long-term information. In contrast, the second U-Net structure replaces the convolutional encoder-decoder layers with multi-head attention layers, which could effectively alleviate the limited receptive filed caused by local operation of convolution. To perform the down-sampling and up-sampling operation in encoder and decoder, we insert a down-sampling block and an up-sampling block into encoder and decoder layer, respectively. Both proposed models will be described as follows.

3.2.1 Proposed MHAUNet-1

Different from the self-attention mechanism between encoder and decoder of the U-Net described in chapter 2, we propose here a new U-Net applying the multi-head attention after the encoder to further extract contextual information from different aspects, named as multi-head attention U-Net-1 (MHAUNet-1) as shown in Fig. 3-7.

Our proposed MHAUNet-1 consists of convolutional encoder-decoder structure and a two-stage multi-head attention block (TSMHAB) in between, where the multi-head attention block employs a sample and frame multi-head attention to extract short- and long-term information. To form a two-stage multi-head attention block (TSMHAB), we first propose a multi-head attention block (MHAB-1) as a backbone. Our proposed MHAB is comprised of a multi-head attention and a GRU block as shown in Fig. 3-8.

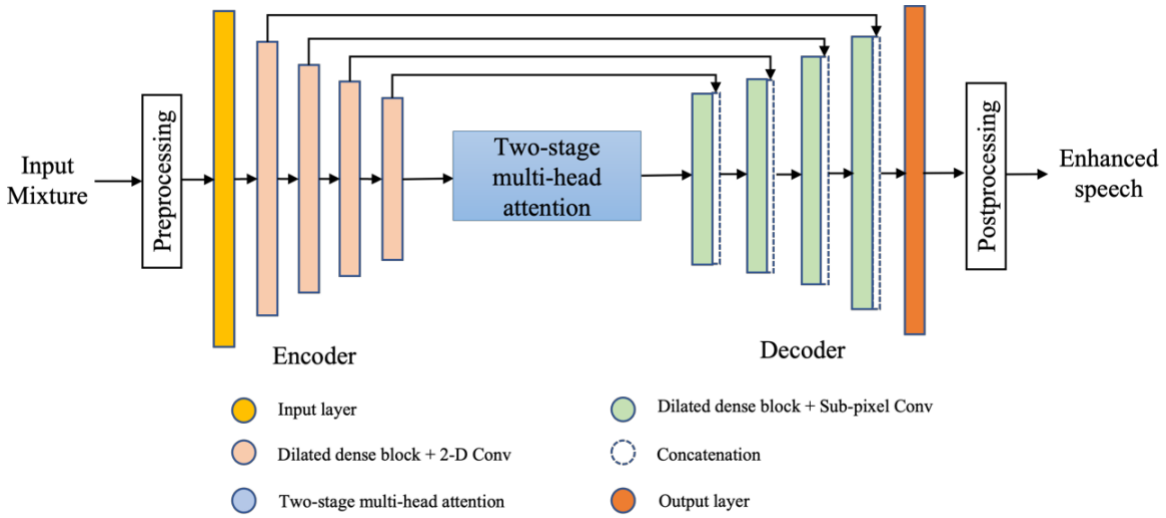


Figure 3-7: Proposed MHAUNet-1

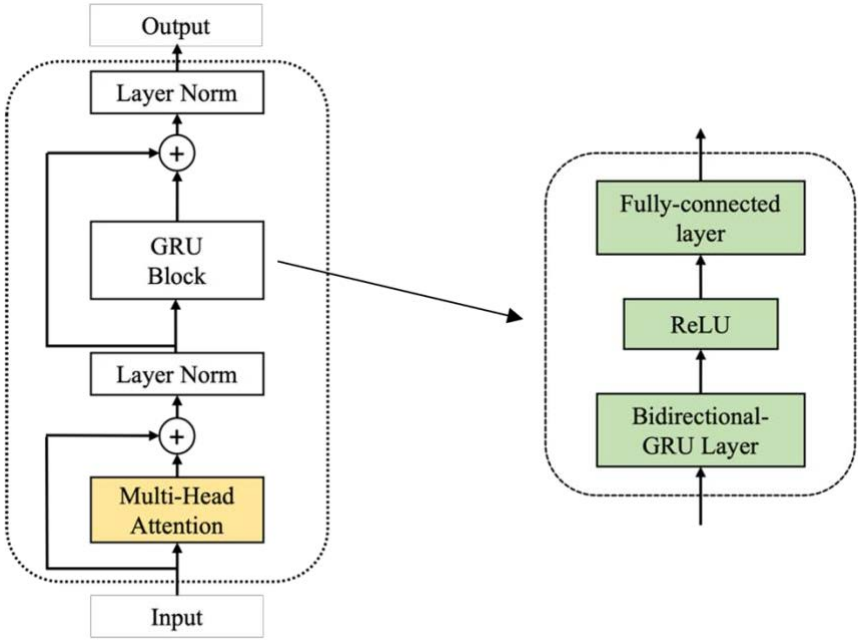


Figure 3-8: Proposed MHAB-1

Unlike self-attention we adopted in chapter 2, the multi-head attention is employed to extract the global dependency of time sequences from various aspects. The output of certain sample has a relationship with itself and other samples. The attention mechanism decides to focus on how many and which neighboring samples would have an impact on this sample. However, the sequential information is ignored in the attention procedure, which is important for the modeling of speech

sequence. To solve this problem, we adopt a GRU block to track the positional information of speech sequences, which consists of a bi-directional GRU (Bi-GRU) layer, a fully-connected (FC) layer and a ReLU nonlinearity in between.

In the proposed MHAB-1, the input features $X \in \mathbb{R}^{T \times F}$ are first passed through the MHA to get the attentive feature representation $X_{att} \in \mathbb{R}^{T \times F}$ without positional information, where T denotes the sequency length and F is the feature dimension of each time step. Then, the GRU block is used to learn the sequential information of features X_{att}, where a Bi-directional GRU layer is utilized to increase the feature dimension of X_{att} from F into $4F$. In that case, the feature space is projected into larger one for learning more useful information. Finally, the FC layer is employed to project the higher feature dimension $4F$ back to smaller one F. Additionally, the residual connection and layer normalization are employed after MHA and GRU block to avoid the gradient vanishing and speed up the convergency. The whole procedures can be formulated as follows:

$$X_{att} = MultiHeadAttention(X) \tag{3-4}$$

$$Y_{GRU} = BiGRU(LN\ (X + X_{att})) \tag{3-5}$$

$$Y = LN(ReLU(Y_{GRU})W^{o} + X_{att}) \tag{3-6}$$

where $Y_{GRU} \in \mathbb{R}^{T \times 4F}$ is the output features from the Bi-GRU layer, $W^{o} \in \mathbb{R}^{4F \times F}$ denotes the linear matrix for projecting the feature dimension, $LN(\cdot)$ and $ReLU\ (\cdot)$ are layer normalization and ReLU non-linearity operation, $Y \in \mathbb{R}^{T \times F}$ means the final outputs of MHAB.

Based on the multi-head attention block, we propose a two-stage multi-head attention block (TSMHAB) by using the dual-path structure to efficiently extract of contextual information of long-range speech sequences as shown in Fig. 3-9. The dual-path structure is proposed to solve the high computation problem of RNNs on modeling long-range speech separation. In this structure, the long speech sequences are first divided into multiple overlapped speech frames, where the local relationship is learnt in each individual frame and then the global information is captured by fusing different frames.

We employ the multi-head attention block mentioned above to perform the dual-path extraction for speech enhancement. As shown in Fig. 3-9, our proposed TSMHAB consists of sample multi-head attention and frame multi-head attention to successively extract local relationship within separate samples and global contextual information among different frames, respectively. In the proposed TSMHAB, the input features I with a dimension format of $[B, C, N, F]$ is first reshaped to the data format with $[B * N, F, C]$ which can be processed by the MHA blocks. Then, the sample

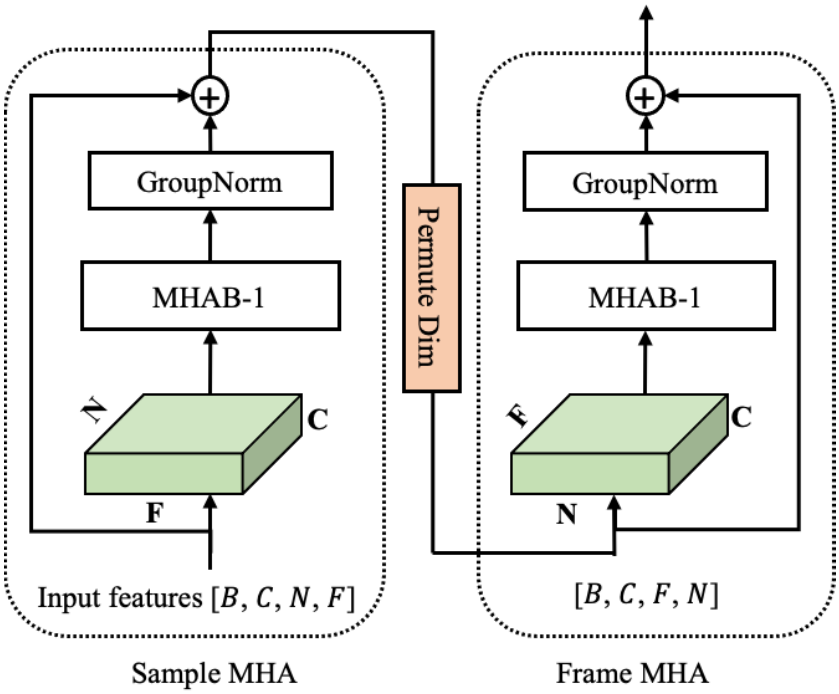


Figure 3-9: Proposed two-stage multi-head attention block

MHA is adopted to parallelly capture the local relationship in individual frames, which implements on the dimension F of the features. Next, the output features from sample MHA are permuted to the data format with $[B * F, N, C]$ before passing through the frame MHA. Subsequently, the frame MHA is employed to summarize the information of different frames to extract the global contextual information along the dimension N. Besides, the group normalization is applied after each MHA block, and the residual connection from the inputs of MHA block is added with outputs of MHA block for avoiding gradient vanishing. The procedures can be formulated as follows:

$$Out_{L-MHA} = SMHA\ (Reshape(I)[:, f, :]) + Reshape(I) \tag{3-7}$$

$$Out_{G-MHA} = FMHA\ (Reshape(Out_{L-MHA})[:, n, :]) + Reshape(Out_{L-MHA}) \tag{3-8}$$

where $f = 1, 2, \cdots, F$, is the index of sample in each chunk, and $n = 1, 2, \cdots, N$ denotes the chunk index. $Reshape(\cdot)$ denotes the changes of feature layout, equivalent to the operation of dimension permutation as shown in Fig. 3-9.

We combine the proposed TSMHAB with the encoder-decoder based U-Net to construct our MHAUNet-1 as shown in Fig. 3-7, which has a U-Net structure similar to the DBAUNet we proposed in chapter 2. More specifically, the input noisy waveform is first transformed into

overlapped chunks to form a 3-D tensor $X \in \mathbb{R}^{C \times N \times F}$, where $C = 1$ denotes the feature channel, N is the number of chunks and F the length of each chunk. Then, the convolutional encoder is adopted to extract the compressed features of X by employing one input layer and four down-sampling layers. More specifically, the input layer increases the feature channel of inputs from 1 into 64 by using 1×1 convolutional layer. Then the down-sampling layer decreases the chunk length by a factor of 2 using a dilated dense block and a shifted 2-D convolutional layer. After the encoder, we obtain the compressed features $I \in \mathbb{R}^{64 \times N \times H}$, where N depends on the length of original noisy waveform and $H = F/2^4$ is the chunk size after passing through four down-sampling layers. Next, the features I are further extracted by the proposed TSMHAB to learn the contextual information, where the local and global information is successively captured by sample MHA and frame MHA to generate the better feature representation $I_{context} \in \mathbb{R}^{64 \times N \times H}$.

The decoder is to reconstruct the enhanced speech from $I_{context}$, which includes four up-sampling layers and one output layer. Each up-sampling layer is to increase the chunk size by a factor of 2 using dilated dense block and sub-pixel convolution. The output layer reduces the feature channel from 64 into 1. To reuse the multi-scale features from encoder, the skip connection used in each encoder layer is also applied to the corresponding decoder layer. After decoder, we obtain the recovered speech features $Y \in \mathbb{R}^{1 \times N \times F}$ which will be transformed into enhanced speech waveform by postprocessing.

3.2.2 Proposed MHAUNet-2

Most U-Net architectures are based on convolutional encoder-decoder structure. However, the intrinsic locality problem of convolutional operation makes the U-Net difficult to explore the long-range relationship for long speech sequences. Although our proposed MHAUNet-1 employs the multi-head attention to further capture the long-range dependency of the encoder outputs, some potential information is missed in the compressed features in a convolutional encoder.

To tackle this shortage, we construct multi-head attention based encoder layers and decoder layers to make an efficient and improved U-Net for speech enhancement which can capture long-range relationship of features in the level of encoder and decoder. The whole framework of the proposed multi-head attention U-Net-2 (MHAUNet-2) is shown in Fig. 3-10, which consists of an encoding module, a decoding module, and a U-shape MHA module in between.

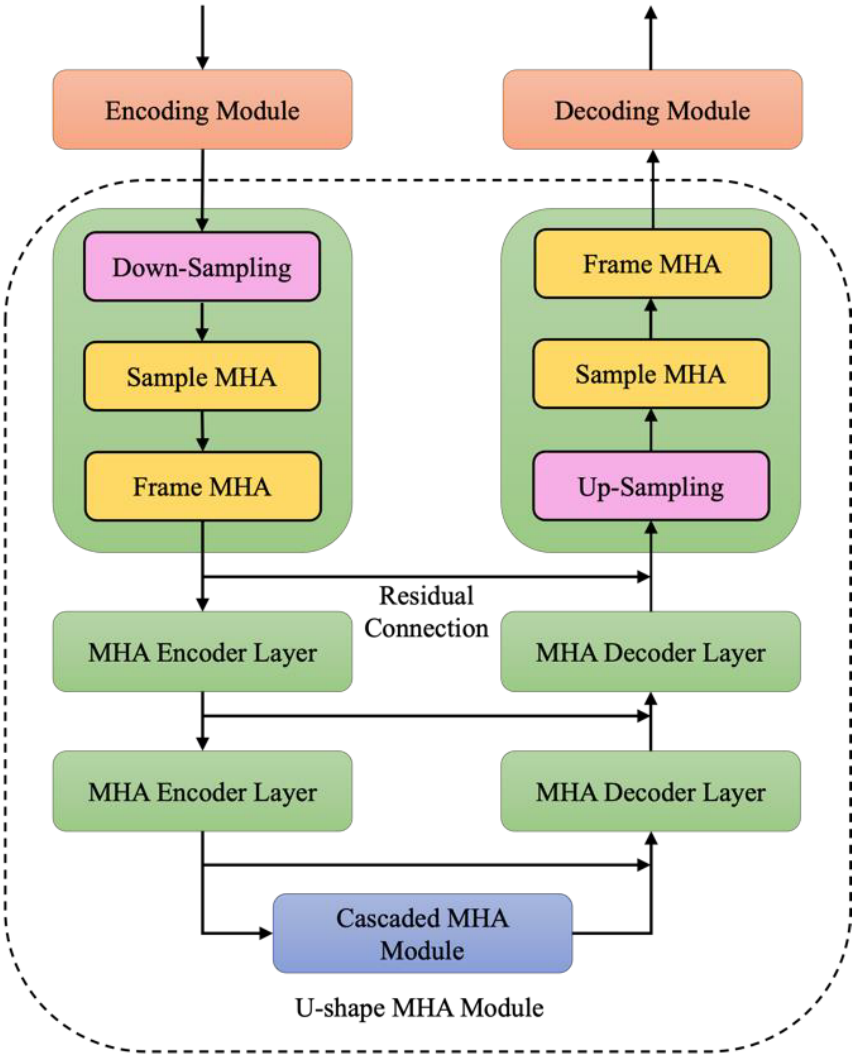


Figure 3-10: Proposed MHAUNet-2

The encoding module consists of two convolutional layers with a dilated-dense block in between as shown in Fig. 3-11. The first convolutional layer increases input channels from 1 to 64 by using 2-D convolutions with a kernel of size (1, 1), followed by a dilated-dense block with four dilation convolution layers for extracting low-level features. We use dilated depth-wise separable convolutions in dense block instead of conventional convolutions for increasing the receptive field of features while involving lower model parameters. Then, the dimensionality of frame size and channel is halved by 2-D convolutions with kernel size (1, 3) using a stride of (1, 2), which could be helpful for decreasing the computational complexity of the following U-shape MHA module. All convolutional layers are followed by layer normalization and PReLU nonlinearity.

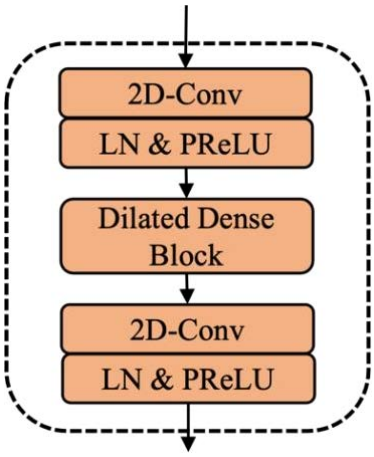


Figure 3-11: Encoding module

The output features from encoding module are processed by the U-shape MHA module consisting of multiple MHA encoder layers, cascaded MHA module and multiple MHA decoder layers. To form the encoder and decoder layers, we first improve the MHAB-1 used in MHAUNet-1 in some aspects as shown in Fig. 3-12. More specifically, we replace the ReLU non-linearity of GRU block with GELU non-linearity which avoids the zero values caused by negative inputs but includes all benefits of ReLU function. Additionally, we adopt a pre-norm strategy by applying the layer normalization before MHA and GRU block for more efficient training. Furthermore, the third layer normalization is added to outputs of MHAB-2 for regularization.

Based on the MHAB-2, we constitute the MHA based encoder layer using a down-sampling layer, a sample MHA and a frame MHA as shown in Fig. 3-13. In details, the feature from the encoding module has a size of $[C, K, S]$, where C, K, S denote the number of channels, the number of frames and frame length, respectively. A down-sampling block is applied to halve the dimension S by using 2-D convolutions with kernels of size (1, 3) and a stride of (1, 2), which is followed by layer normalization and PReLU nonlinearity.

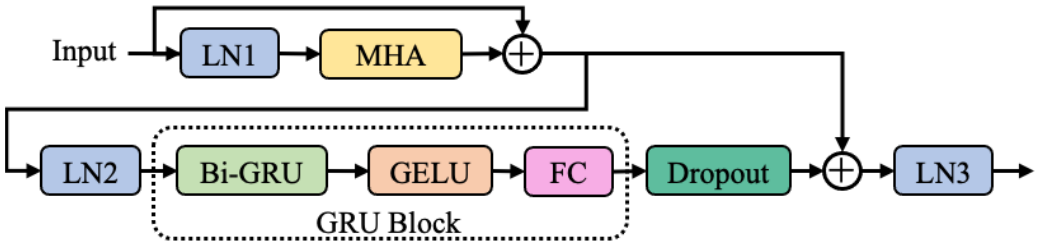


Figure 3-12: Proposed MHAB-2

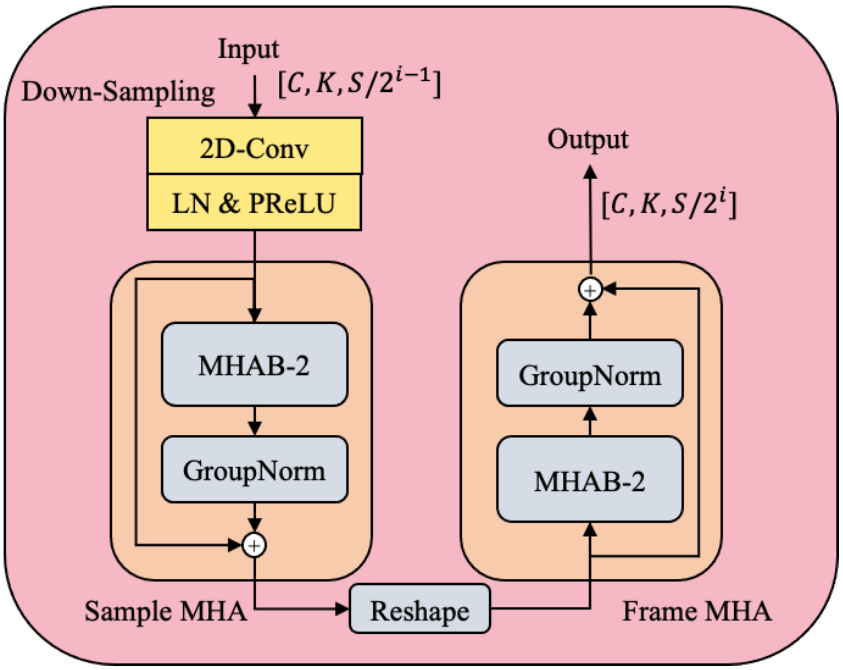


Figure 3-13: MHA encoder layer

After that, the sample MHA is first employed on separate frames to parallelly learn the short-term information in each frame by performing on the dimension S of the input feature. Next, the frame MHA summarizes the information of different frames to learn long-term dependencies, which implements on the dimension K of the feature. After passing through D MHA-based encoder layers, the shape of feature becomes $[C, K, S/2^D]$ when it is fed into the cascaded MHA blocks. Each cascaded MHA block has a pair of sample and frame MHA blocks with cascaded connection to further extract high-level features of long-range speech sequences.

To recover the enhanced speech features from compressed latent features, the U-shape MHA module uses the same number of MHA-based decoder layers as that of encoder layers to reconstruct the feature representation, where each layer consists of sample and frame MHA blocks and the up-sampling block in front of them, as shown in Fig. 3-14.

In each MHA decoder layer, the up-sampling block doubles the dimension of frame length by using the sub-pixel convolution with an expansion factor 2. There is also a residual connection added between the corresponding MHA encoder and decoder layers for improving signal reconstruction and avoiding gradient vanishing problems during training. After D decoder layers, the shape of feature can be recovered from $[C, K, S/2^D]$ to $[C, K, S]$.

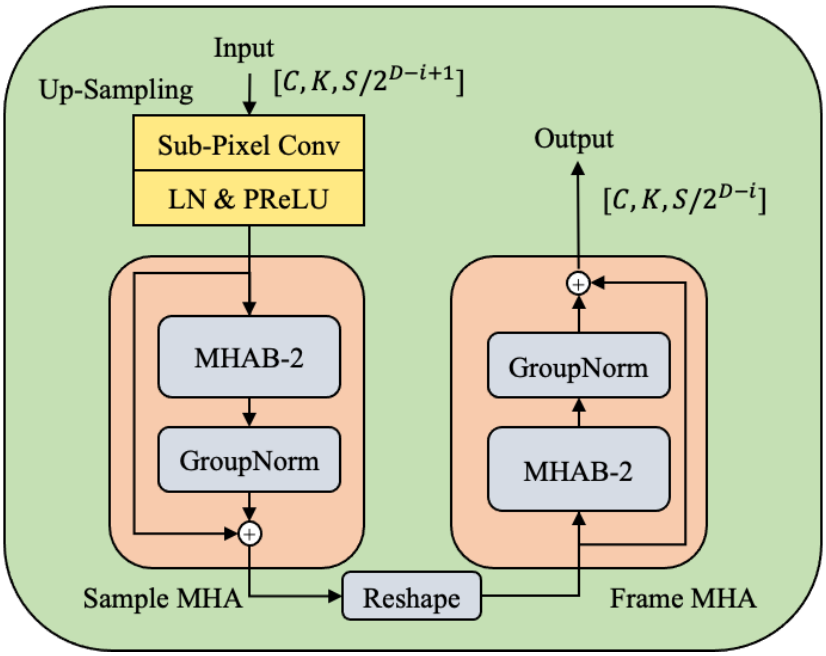


Figure 3-14: MHA decoder layer

The decoding module is a symmetric representation of the encoding module, including two convolutional layers and a dilated-dense block in between as shown in Fig. 3-15. First, the sub-pixel convolution is used to double the last dimension of feature representation, followed by the dilated-dense block which has the same structure as the one in the densely encoding module. Finally, the second convolutional layer decreases the number of channels of the feature from 64 to 1 by using a filter of size (1, 1), and finally obtains the enhanced speech by the postprocessing described in chapter 2.

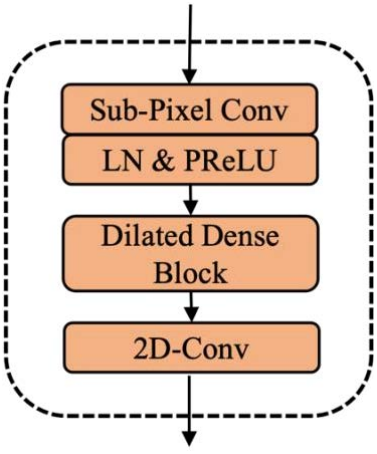


Figure 3-15: Decoding module

3.3 Experiments

3.3.1 Experimental setup

To verify our proposed MHAUNet-1 and MHAUNet-2 model, we conduct thorough experimental work with comparisons to other existing models on the VCTK dataset including female and male speakers. In the VCTK dataset, 11572 utterances are selected for training the model, which includes 14 male and 14 female speakers. A total of 10 types of noises from DEMAND and artificially generated noise dataset are used for generating the noisy training dataset which has four SNRs levels including 15 dB, 10 dB, 5 dB and 0 dB. To evaluate the trained models, 824 utterances spoken by one male and one female are employed by mixing with 5 types of unseen noise with different SNRs including 17.5 dB, 12.5 dB, 7.5 dB and 2.5 dB. Like chapter 2, some commonly used evaluation metrics are used to evaluate the performance of speech enhancement. PESQ and STOI are two most popular evaluation metrics to evaluate the speech perceptibility and speech intelligibility. SSNR computes the SNRs and energies of segments from the estimated and clean speech. We also adopt the composite metrics including CSIG for speech distortion, CBAK for background noise and COVL for overall speech quality.

For preprocessing stage, we resample the utterances at 16 kHz and randomly select 4 seconds of each utterance for training. To construct the inputs of our models, we split each 4-second segment into multiple overlapped chunks, where each chunk has a length of 512 samples with a shift of 256 samples to the next chunk. We set batch size as 2 due to the limited hardware computational resources.

To have a fair comparison, both MHAUNet-1 and MHAUNet-2 adopt four encoder and decoder layers. In MHA block, four heads are used for extracting features from different subspaces, and 64 hidden units are used in GRU block. We train our proposed models for 100 epochs by using Adam optimizer. We use the gradient clipping to restrain the maximum norm of gradient to 5 for avoiding the gradient explosion problem. The dynamic learning rate is scheduled during the training. Specifically, we first linearly increase the value of learning rate from 0 to $4e^{-4}$ within the first 4000 training steps, and then decay it by 0.98 for every two epochs. The details can be formulated as follows:

$$Lr = \begin{cases} k_1 \cdot d^{-0.5} \cdot n \cdot n\_warmups^{-1.5} & , \quad n \leq n\_warmups \\ k_2 \cdot 0.98^{\lfloor epoch/2 \rfloor} & , \quad n > n\_warmups \end{cases} \tag{3-9}$$

where n is the number of training steps, and k_1 and k_2 are hyperparameters, which are set as 0.2 and $4e^{-4}$, respectively. *n_warmups* denotes the number of warm-ups, which is set as 4000. Finally, d denotes a hyperparameter which is set to 64 in our experiments.

To train our proposed MHAUNets, the loss function calculated on waveform domain and time-frequency domain is used, where waveform domain loss is based on the mean square error (MSE) between the clean and denoised waveform, and the time-frequency domain loss is based on the spectrograms computed by clean and enhanced waveform by STFT.

$$\mathcal{L}^F = \frac{1}{TF}\sum_{t=0}^{T-1}\sum_{f=0}^{F-1}[\left(|S_{real}(t,f)| + \left|S_{img}(t,f)\right|\right) - \left(\left|\hat{S}_{real}(t,f)\right| + \left|\hat{S}_{img}(t,f)\right|\right)] \tag{3-10}$$

$$\mathcal{L}^T = \frac{1}{N}\sum_{i=0}^{N-1}(x_i - \hat{x}_i)^2 \tag{3-11}$$

$$\mathcal{L} = \varphi * \mathcal{L}^T + (1-\varphi)\mathcal{L}^F \tag{3-12}$$

where $\mathcal{L}^T$ and $\mathcal{L}^F$ denote the loss on waveform and time-frequency domain, respectively. $\{S_{real}, S_{img}\}$ mean real and imaginary components of the spectrogram of clean waveform. $\{\hat{S}_{real}, \hat{S}_{img}\}$ represent real and imaginary parts of the spectrogram of enhanced waveform. T and F are the number of frames and that of frequency bins. $\{x, \hat{x}\}$ denote the samples of clean and enhanced speech waveform. N is the number of speech samples. $\mathcal{L}$ denotes the final mixed domain loss, which is weighted sum over waveform domain loss and time-frequency domain with factor φ. We empirically set φ = 0.5 in our following experiments.

3.3.2 Comparison with baselines

We compare our proposed models with other time or T-F domain speech denoising models, all evaluated on the same dataset. The experimental results are presented in Table 3-1. First, the proposed MHAUNet-1 and MHAUNet-2 achieve an obvious improvement of PESQ score from 1.97 to 2.96 and to 3.00, respectively, indicating a competitive performance with other methods while involving fewer model parameters. DEMUCS achieves the best PESQ score, however, it employs many data augmentation technologies and involves much more trainable parameters

which are about 47 times larger than MHAUNet-1 and about 31 times larger than MHAUNet-2, respectively.

Table 3-1: Performance comparison with existing methods

Model	PESQ	STOI	CSIG	CBAK	COVL	SSNR	Para. (M)
Unprocessed	1.97	0.91	3.34	2.44	2.63	1.73	-
WaveNet [8]	-	-	3.62	3.23	2.98	-	-
CNN-GAN [54]	2.34	0.93	3.55	2.95	2.92	-	-
Wave U-Net [41]	2.40	-	3.52	3.24	2.96	9.97	10.00
Att. WU-Net [48]	2.63	-	3.95	3.30	3.29	-	-
MetricGAN [56]	2.86	-	3.99	3.18	3.42	-	-
DCUNet-16 [57]	2.93	-	4.10	**3.77**	3.52	**14.44**	2.3
PHASEN [66]	2.99	-	4.21	3.55	3.62	10.18	8.76
MHSA-SE [63]	2.99	-	4.15	3.46	3.51	-	-
LLUNet [67]	2.90	0.94	4.22	3.32	3.58	-	-
NAAGN [68]	2.90	-	4.13	3.50	3.51	10.25	-
DEMUCS [42]	**3.07**	0.95	**4.31**	3.40	3.63	-	33.5
T-GSA [69]	3.06	-	4.18	3.59	3.62	10.78	110
SASE [57]	2.76	0.94	4.09	3.32	3.43	-	10.35
CARN [70]	2.93	0.95	4.19	3.61	3.54	-	-
SADNUNet [49]	2.82	-	4.18	3.47	3.51	-	2.63
MHAUNet-1 (ours)	2.96	0.94	4.23	3.51	3.61	9.87	**0.69**
MHAUNet-2 (ours)	3.00	**0.95**	4.26	3.54	**3.65**	9.47	1.04

Regarding the speech intelligibility evaluation, the proposed MHAUNet-2 obtains the best STOI score of 95%. The proposed MHAUNet-1 and MHAUNet-2 obtain similar MOSs, while MHAUNet-2 achieving a superior performance in CSIG and COVL. The best CBAK value is achieved by DCUNet adopting the complex convolutional neural network and having much more model parameters. Moreover, from Table 3-1, we can observe that our proposed MHAUNet-1 and MHAUNet-2 achieve a superior performance to most U-Net based methods working on time

domain or T-F domain. More specifically, both MHAUNet-1 and MHAUNet-2 obtains a better PESQ score than WaveUNet, Attention WaveUNet, DCUNet, LLUNet, SADNUet and SASE. In terms of composite evaluation scores, MHAUNet-1 and MHAUNet-2 achieve a better performance in COVL than other U-Net based models.

Comparing with other attention-based models, we observe that our proposed multi-head attention based models yield a competitive performance on speech enhancement. Especially, the MHAUNet-2 presents more impressive performance in all evaluation metrics than NAAGN, CARN and Self-Adapt MHSA. Although T-GSA achieves a slightly better PESQ value (3.06) than MHAUNet-2, it involves much more trainable parameters (about 110 million parameters).

Compared with existing speech enhancement methods, our proposed MHAUNet-1 and MHAUNet-2 give competitive PESQ and composite evaluation metrics while having a very low model complexity. By introducing the sample and frame multi-head attentions to capture local relationship and global contextual information, the U-Net architecture can efficiently extract abundant contextual information of long-range speech sequences.

3.3.3 Ablation study

In this section, we present some findings through ablation studies by using different model configurations. First, our proposed MHAUNet-1 adopts the two-stage multi-head attention block (TSMHAB) to extract the contextual information of the output features from U-Net encoder. To explore the effectiveness, we explore other different blocks inserted between the encoder and decoder of U-Net, including LSTM layers, temporal convolutional network (TCN) block and dual-branch attention block (DBAB).

To have a fair comparison, we use the same encoder-decoder structure as the MHAUNet-1. For simplifying the experiment, we use two LSTM layers, two TCN blocks and one DBAB as the inserted blocks between encoder and decoder, which are named as UNet-LSTM, UNet-TCN and DBABUNet, respectively. Note that, the DBABUNet is our proposed self-attention based U-Net structure in chapter 2, where the proposed two-branch attention is used to parallelly extract spatial- and channel-wise information. Moreover, we design a pure U-Net structure without inserted block to verify whether capturing the contextual information is required, which is termed as Pure U-Net.

From Table 3-2, we can observe that our proposed MHAUNet-1 achieves the best score in all evaluation metrics compared with other reference models with different inserted blocks. Especially, the Pure U-Net obtains the worst performance of speech denoising, indicating that the U-Net structure without inserted block is hard to extract contextual information due to the limited receptive filed of convolutional operation.

Moreover, the attention-assisted MHAUNet-1 and DBAUNet outperform the reference models using inserted LSTM or TCN layers, showing that the attention mechanism with parallel operation can efficiently extract long-range dependency of speech sequences. Different from DBAUNet using self-attention, the new proposed MHAUNet-1 adopts the multi-head attention to extract abundant features from different subspaces, which further improves the quality of extracted contextual information.

Table 3-2: Performance of different model configuration

	PESQ	STOI	CSIG	CBAK	COVL	SSNR
MHAUNet-1	**2.96**	**0.94**	**4.23**	**3.51**	**3.61**	**9.87**
DBAUNet	2.84	0.94	4.14	3.43	3.50	9.25
UNet-LSTM	2.67	0.94	3.92	3.33	3.30	9.21
UNet-TCN	2.55	0.94	3.81	3.27	3.17	9.20
Pure U-Net	2.54	0.94	3.85	3.26	3.20	8.91

Next, our proposed MHAUNet-2 applies the multi-head attention based encoder and decoder layers, where the down-sampling and up-sampling are incorporated in front of sample and frame MHA block to decrease and increase the dimension of the chunk. To explore the influence of down- or up- sampling position on the denoising performance, we design other two configurations for comparison, where one inserts the sampling block in the center of the sample and frame MHA block and the other employs the sampling block behind the two MHA blocks. The MHAUNets with different sampling positions are shown in Fig. 3-16, named F-UNet, C-UNet and B-UNet, respectively.

As seen from the experimental results listed in Table 3-3, similar performances are achieved by the three models. It also shows that the sampling position has very limited impacts on the performance. Especially, the F-UNet achieves a slightly better performance in all evaluation

metrics than other two comparison models. One of the possible reasons is that the down- or up-sampling block can introduce the relative positional information by using shifted convolutional operation, which could supplement the positional information for multi-head attention mechanism performed in sample and frame MHA.

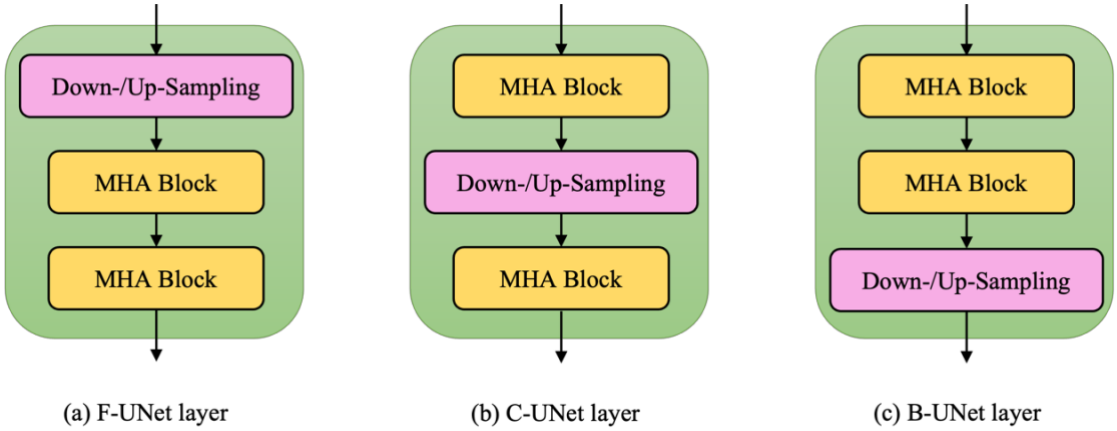


Figure 3-16: Different configurations of encoder-decoder layer

Table 3-3: Performance of different encoder-decoder configurations

	PESQ	STOI	CSIG	CBAK	COVL	SSNR	Para. (M)
Unprocessed	1.97	0.91	3.34	2.44	2.63	1.73	-
F-UNet (ours)	**3.00**	**0.95**	**4.26**	**3.54**	**3.65**	9.47	1.04
B-UNet	2.96	0.94	4.25	3.53	3.62	**9.71**	1.04
C-UNet	2.97	0.95	4.23	3.54	3.62	9.58	1.04

Then, in our proposed MHAUNet-2, the cascaded MHA blocks (CMHABs) are inserted between encoder and decoder to further extract high-level features of the encoder, which is inspired by the MHAUNet-1. In MHAUNet-1, the MHA blocks are incorporated between encoder and decoder to extract contextual information. Different from the MHAUNet-1 using convolutional encoder-decoder, our MHAUNet-2 builds an MHA-based encoder-decoder to generate hierarchical features while considering global contextual information. To further demonstrate the effectiveness of cascaded MHA blocks in our model, we set a reference architecture by simply

removing all cascaded MHA blocks between encoder and decoder in the proposed MHAUNet-1 and MHAUNet-2.

From Table 3-4, we can see that our proposed MHAUNet-2 has best scores in PESQ, STOI and composite evaluation scores compared with reference models, showing the efficiency of MHA-based U-Net on modeling long-range speech sequences. Furthermore, compared with MHAUNet-1 using convolutional encoder and decoder, MHAUNet-2 yields a marginal improvement in evaluation metrics, indicating that MHA-augmented encoder-decoder could be an efficient alternative of convolutional encoder-decoder.

Table 3-4: Explore the efficiency of attention blocks

	PESQ	STOI	CSIG	CBAK	COVL	SSNR
MHAUNet-2	**3.00**	**0.95**	**4.26**	**3.54**	**3.65**	**9.47**
MHAUNet-2 (no middle MHAB)	2.90	0.94	4.22	3.49	3.58	9.15
MHAUNet-1	2.96	0.94	4.23	3.51	3.61	9.87
MHAUNet-1 (no middle MHAB)	2.54	0.94	3.85	3.26	3.20	8.91

When we remove the CMHABs in MHAUNet-2 and TSMHAB in MHAUNet-1, there will be different degrees of declination in the denoising performance, showing that the MHA blocks inserted between encoder and decoder are efficient due to the further extraction of high-level contextual information from encoder features.

It is worth mentioning that the proposed MHAUNet-2 without CMHAB still achieves remarkable results in PESQ (2.90) as well as other evaluation metrics. However, when omitting the MHA blocks from MHAUNet-1, the PESQ score decreases from 2.96 into 2.54, and other scores show a similar declining trend, demonstrating that our proposed MHAUNet-2 is more robust to intermediate MHA blocks than MHAUNet-1. The possible reason is that the proposed MHA-based encoder-decoder layers could extract global contextual information in each depth of network, while the convolution-based U-Net requires to insert extra MHA blocks after the encoder to further extract abundant features.

Furthermore, we select the subset of testing data to observe the performance of our proposed models on given noisy conditions, where the non-stationary Café and Living noises are used to mix with clean utterance to generate the noisy mixtures with SNRs at 2.5 dB, 7.5 dB, 12.5 dB and

17.5 dB. The following tables give the results of the proposed models performed on four different SNR conditions. We can see that our proposed models can obviously remove the background noises at various degrees. Especially, our MHAUNet-2 achieves the best denoising performance in the condition of Café and Living noises at lower SNR levels including 2.5 dB and 7.5 dB. For example, under the 2.5 dB circumstance, the MHAUNet-2 improves the PESQ value from 1.15 into 1.87 in Cafe noise and from 1.18 into 2.21 in Living noise, which impressively improves the speech quality in challenging noisy environments. In addition, both MHAUNet-2 and MHAUNet-1 achieve a similar STOI improvement under two noisy conditions.

Table 3-5: Performance of proposed U-Nets at 2.5 dB

Methods	PESQ		STOI (%)	
	Café	Living	Café	Living
Unprocessed	1.15	1.18	78.13	84.25
DBAUNet	1.72	1.81	85.87	88.99
MHAUNet-1	1.82	2.08	87.27	90.07
MHAUNet-2	**1.87**	**2.21**	**87.59**	**90.43**

Table 3-6: Performance of proposed U-Nets at 7.5 dB

Methods	PESQ		STOI (%)	
	Café	Living	Café	Living
Unprocessed	1.33	1.41	87.59	90.67
DBAUNet	2.25	2.35	91.44	93.15
MHAUNet-1	2.38	2.66	91.84	93.53
MHAUNet-2	**2.40**	**2.75**	**91.91**	**94.41**

In higher SNR conditions including 12.5 dB and 17.5 dB, both MHAUNet-1 and MHAUNet-2 achieve a competitive performance on the speech quality and intelligibility. More specifically, MHAUNet-1 and MHAUNet-2 get the same PESQ value in Café condition with a SNR at 12.5 dB. However, the best PESQ score in living noisy environment is achieved by MHAUNet-1. For the 17.5 dB condition, MHAUNet-2 still shows the great performance, achieving the best score in

two kinds of noisy situations except that the best STOI performance is obtained by MHAUNet-1 in café environment.

Table 3-7: Performance of proposed U-Nets at 12.5 dB

Methods	PESQ		STOI (%)	
	Café	Living	Café	Living
Unprocessed	1.54	1.66	91.23	91.69
DBAUNet	2.55	2.55	93.51	93.20
MHAUNet-1	**2.74**	**2.92**	94.15	94.04
MHAUNet-2	**2.74**	2.89	**94.35**	**94.41**

Table 3-8: Performance of proposed U-Nets at 17.5 dB

Methods	PESQ		STOI (%)	
	Café	Living	Café	Living
Unprocessed	1.95	2.24	94.38	95.17
DBAUNet	2.90	3.11	95.03	96.15
MHAUNet-1	3.11	3.37	**95.62**	96.80
MHAUNet-2	**3.14**	**3.41**	95.55	**96.97**

Finally, we present the enhanced spectrograms of our proposed model to observe whether the noise component is efficiently removed. Here, we select an utterance spoken by a male speaker under the Living noisy environment at SNR level of 2.5 dB. As shown in Fig. 3-17, three proposed models can effectively remove the background noises and maintain a similar harmonic structure.

By carefully comparing the differences between the obtained enhanced spectrogram from the proposed models, we can find that the proposed MHAUNet-2 can remove more noises in the low frequency band than DBAUNet and MHAUNet-1 as presented in the yellow region. Meanwhile, the proposed MHAUNet-2 could restore more approximate structures than other two models as indicated in red region. This finding proves that the proposed MHAUNet-2 is more efficient in reconstructing more similar spectrum details as clean speech in poor noise environment.

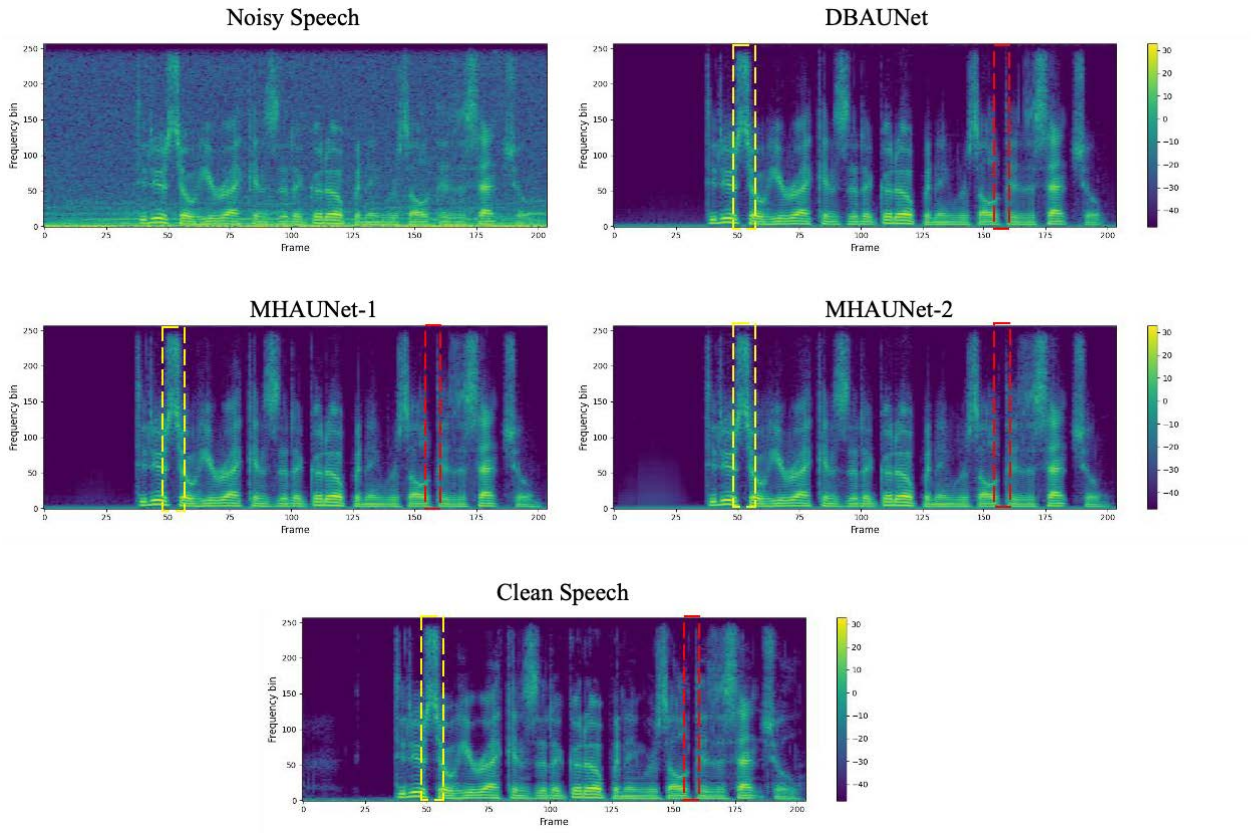


Figure 3-17: Comparison of spectrograms of proposed models at 2.5 dB

3.4 Summary

In this chapter, we have proposed two MHA-assisted U-Nets for single-channel speech enhancement in the time domain, including MHAUNet-1 and MHAUNet-2. The proposed MHAUNet-1 is comprised of convolutional encoder and decoder, and dual-path MHA in between, where the dual-path MHA adopts sample MHA and frame MHA to successively extract the short-term information of each frame and long-term dependency across different frames. However, the encoder and decoder of the proposed MHAUNet-1 require to be deep enough for having large receptive field due to the local operation of convolution, which constrains the effectiveness of extracting the contextual information. To tackle this problem, our proposed MHAUNet-2 constructs MHA-based encoder and decoder layers, where the MHA combines with down- or up-sampling to extract the global dependency of speech features in each depth of the network.

Different from MHAUNet-1, the MHAUNet-2 performs the extraction of contextual information at the encoder and decoder level.

The experimental results demonstrate that our proposed MHAUNet-1 and MHAUNet-2 have outperformed most existing speech enhancement models in most evaluation criteria. Especially, the MHAUNet-2 obtains a better denoising performance than MHAUNet-1, indicating that the multi-head attention based encoder-decider is an efficient alternative to convolution-based one. Moreover, our proposed MHAUNet-1 and MHAUNet-2 involve comparatively fewer trainable parameters than existing methods.

Chapter 4
Conclusion and Future Work

4.1 Conclusion

In this thesis, various deep-learning methods for speech enhancement have been introduced, especially the U-Net neural network and attention mechanism. The U-Net is a symmetric structure consisting of a down-sampling encoder and an up-sampling decoder, which is often combined with other types of neural networks as an efficient component between encoder and decoder. The attention mechanism has been a popular approach to deal with long-term sequences separately, which usually perform jointly with other neural networks. The proposed deep learning based SE methods in this thesis are designed based on the U-Net and the attention mechanism.

In chapter 2, a U-Net with a dual-branch attention block between encoder and decoder is proposed for speech enhancement. The U-Net adopts the dense block with multiple densely connected convolutions in each encoder and decoder layers, where the inner convolution is the dilated depth-wise separable convolution with capability of reducing the number of parameters and expanding the receptive field. Moreover, the decoder uses the sub-pixel convolution for up-sampling to avoid checkerboard distortion. The dual-branch attention block employs channel-path and spatial-path attention to respectively learn the information along these two dimensions and combines the features from the two paths in an average manner. Experimental results show that the U-Net with dual-branch attention block achieves comparable performance with much fewer model parameters compared to existing models, and the proposed U-Net outperforms the pure U-Net and the U-Nets with LSTM, TCN, or one-path attentions.

In chapter 3, two MHA based U-Nets are proposed based on the proposed U-Net structure in chapter 2, namely MHAUNet-1 and MHAUNet-2, where the former incorporates MHA blocks in the U-Net and the latter applies MHA blocks in encoder-decoder layers. For the MHAUNet-1, the two MHA blocks are sequentially connected to extract features within individual frames and contextual information between different frames, named sample MHA and frame MHA, respectively. In contrast, the proposed MHAUNet-2 applies MHA block as well as down- or up-

sampling layer to construct an MHA-based encoder-decoder layer, which can alleviate the limited receptive field existing in convolutional encoder-decoder of MHAUNet-1. By conducting extensive experiments on the public dataset, it was shown that our proposed MHAUNet-1 and MHAUNet-2 can further improve the performance of the dual-branch attention U-Net proposed in chapter 2 and moreover, they outperform most existing methods in many evaluation criteria with comparably light model complexities.

4.2 Future work

This thesis has investigated the U-Net based architectures for single-channel speech enhancement in the time domain. With a few modifications, our proposed models can be easily transferred from time domain into T-F domain, which takes the spectrogram feature of speech waveform as input. Moreover, in some scenarios, we need to enhance a target speech signal corrupted by background noises with multiple microphones, like multi-person meeting room. Our models can be flexibly implemented on the multi-channel speech enhancement task by adjusting a few network configurations. Furthermore, the speech enhancement systems are usually regarded as a preprocessing stage of speech recognition to improve the recognition accuracy of words. It would be interesting to observe the recognition performances of our proposed models combined with speech recognition system.

Finally, casual speech enhancement has been gradually studied by researchers due to the increasing real-time applications. That means the speech denoising performs in real time when recording the voice of speakers, which requires the SE systems only process the current time step and its previous ones. To achieve this goal, we can adopt our proposed modes to build casual SE systems by using casual convolutions, masked attention mechanism and unidirectional RNN layers.

Bibliography

[1] J. Lim and A. Oppenheim, "All-pole modeling of degraded speech," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 3, pp. 197–210, 1978.

[2] Y. Ephraim, "Statistical-model-based speech enhancement systems," *Proceedings of the IEEE*, vol. 80, no. 10, pp. 1526–1555, 1992.

[3] 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. IEEE, 1979, pp. 208–211.

[4] M. Dendrinos, S. Bakamidis, and G. Carayannis, "Speech enhancement from noise: A regenerative approach," *Speech Communication*, vol. 10, no. 1, pp. 45–57, 1991.

[5] N. Saleem and M. I. Khattak, "Deep neural networks for speech enhancement in complex-noisy environments." *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 1, pp. 84–90, 2020.

[6] Y. Zhao, D. Wang, I. Merks, and T. Zhang, "DNN-based enhancement of noisy and reverberant speech," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 6525–6529.

[7] 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)*. IEEE, 2017, pp. 1–6.

[8] D. Rethage, J. Pons, and X. Serra, "A wavenet for speech denoising," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5069–5073.

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

[10] J. Abdulbaqi, Y. Gu, S. Chen, and I. Marsic, "Residual recurrent neural network for speech enhancement," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6659–6663.

[11] T. Lan, Y. Lyu, W. Ye, G. Hui, Z. Xu, and Q. Liu, "Combining multi-perspective attention mechanism with convolutional networks for monaural speech enhancement," *IEEE Access*, vol. 8, pp. 78 979–78 991, 2020.

[12] C. Zheng, X. Peng, Y. Zhang, S. Srinivasan, and Y. Lu, "Interactive speech and noise modeling for speech enhancement," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 16, 2021, pp. 14549–14557.

[13] U. Kjems, J. B. Boldt, M. S. Pedersen, T. Lunner, and D. Wang, "Role of mask pattern in intelligibility of ideal binary-masked noisy speech," *The Journal of the Acoustical Society of America*, vol. 126, no. 3, pp. 1415–1426, 2009.

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

[15] 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.

[16] 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.

[17] 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)*. IEEE, 2015, pp. 708–712.

[18] 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 *Journal of the Acoustical Society of America*, vol. 134, no. 5, pp. EL452–EL458, 2013.

[19] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *ICML*, 2010

[20] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458,* 2015.

[21] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint arXiv:1603.07285*, 2016.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] K. Cho, B. Van Merri¨enboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[24] 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*, vol. 93, p. 27403, 1993.

[25] K. Ito and L. Johnson, “The LJ speech dataset,” 2017.

[26] 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.

[27] C. Valentini-Botinhao, X. Wang, S. Takaki, and J. Yamagishi, “Investigating RNN-based speech enhancement methods for noise-robust text-to-speech.” in *SSW*, 2016, pp. 146–152.

[28] C. Veaux, J. Yamagishi, and S. King, “The voice bank corpus: Design, collection and data analysis of a large regional accent speech database,” in *2013 International Conference Oriental COCOSDA Held Jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*. IEEE, 2013, pp. 1–4.

[29] J. Thiemann, N. Ito, and E. Vincent, “The diverse environments multi-channel acoustic noise database (demand): A database of multichannel environmental noise recordings,” in *Proceedings of Meetings on Acoustics ICA2013*, vol. 19, no. 1. Acoustical Society of America, 2013, p. 035081.

[30] N. Saleem and M. I. Khattak, “A review of supervised learning algorithms for single channel speech enhancement,” *International Journal of Speech Technology*, vol. 22, no. 4, pp. 1051–1075, 2019.

[31] P. C. Loizou, Speech enhancement: theory and practice. *CRC press*, 2007.

[32] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, “An algorithm for intelligibility prediction of time–frequency weighted noisy speech,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011.

[33] [33] J. H. Hansen and B. L. Pellom, “An effective quality evaluation protocol for speech enhancement algorithms.” in *ICSLP*, vol. 7. Citeseer, 1998, pp. 2819–2822.

[34] Y. Hu and P. C. Loizou, “Evaluation of objective quality measures for speech enhancement,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 229–238, 2007.

[35] N. Bjorck, C. P. Gomes, B. Selman, and K. Q. Weinberger, “Understanding batch normalization,” *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[36] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.

[37] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Instance normalization: The missing ingredient for fast stylization,” *arXiv preprint arXiv:1607.08022*, 2016.

[38] Y. Wu and K. He, "Group normalization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[39] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer, 2015, pp. 234–241.

[40] D. Stoller, S. Ewert, and S. Dixon, "Wave-U-Net: A multi-scale neural network for end-to-end audio source separation," *arXiv preprint arXiv:1806.03185*, 2018.

[41] C. Macartney and T. Weyde, "Improved speech enhancement with the Wave-U-Net," *arXiv preprint arXiv:1811.11307*, 2018.

[42] A. Defossez, G. Synnaeve, and Y. Adi, "Real time speech enhancement in the waveform domain," *arXiv preprint arXiv:2006.12847*, 2020.

[43] A. D´efossez, N. Usunier, L. Bottou, and F. Bach, "Music source separation in the waveform domain," *arXiv preprint arXiv:1911.13254*, 2019.

[44] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 933–941.

[45] P. Hewage, A. Behera, M. Trovati, E. Pereira, M. Ghahremani, F. Palmieri, and Y. Liu, "Temporal convolutional neural (TCN) network for an effective weather forecasting using time-series data from the local weather station," *Soft Computing*, vol. 24, no. 21, pp. 16 453–16 482, 2020.

[46] M. N. Ali, A. Brutti, and D. Falavigna, "Speech enhancement using dilated wave-u-net: an experimental analysis," in *2020 27th Conference of Open Innovations Association (FRUCT)*. IEEE, 2020, pp. 3–9.

[47] A. Pandey and D. Wang, "TCNN: Temporal convolutional neural network for real-time speech enhancement in the time domain," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6875–6879.

[48] R. Giri, U. Isik, and A. Krishnaswamy, "Attention Wave-U-Net for speech enhancement," in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019, pp. 249–253.

[49] X. Xiang, Z. Xiaojuan, and C. Haozhe, "A nested U-Net with self-attention and dense connectivity for monaural speech enhancement," *IEEE Signal Processing Letters*, 2021.

[50] A. Pandey and D. Wang, "Dense cnn with self-attention for time-domain speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1270–1279, 2021.

[51] A. Pandey and D. Wang, "Densely connected neural network with dilated convolutions for real-time speech enhancement in the time domain," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6629–6633.

[52] A. Aitken, C. Ledig, L. Theis, J. Caballero, Z. Wang, and W. Shi, "Checkerboard artifact free sub-pixel convolution: A note on sub-pixel convolution, resize convolution and convolution resize," *arXiv preprint arXiv:1707.02937*, 2017.

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

[54] N. Takahashi, P. Agrawal, N. Goswami, and Y. Mitsufuji, "Phasenet: Discretized phase modeling with deep neural networks for audio source separation." in *Interspeech*, 2018, pp. 2713–2717.

[55] M. H. Soni, N. Shah, and H. A. Patil, "Time-frequency masking-based speech enhancement using generative adversarial network," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5039–5043.

[56] S.-W. Fu, C.-F. Liao, Y. Tsao, and S.-D. Lin, "MetricGAN: Generative adversarial networks based black-box metric scores optimization for speech enhancement," in *International Conference on Machine Learning. PMLR*, 2019, pp. 2031–2041.

[57] H.-S. Choi, J.-H. Kim, J. Huh, A. Kim, J.-W. Ha, and K. Lee, "Phase-aware speech enhancement with deep complex u-net," in *International Conference on Learning Representations*, 2018.

[58] M. Strauss and B. Edler, "A flow-based neural network for time domain speech enhancement," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5754–5758.

[59] G. Liu, K. Gong, X. Liang, and Z. Chen, "CP-GAN: Context pyramid generative adversarial network for speech enhancement," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6624–6628.

[60] J. Lin, A. J. Van Wijngaarden, M. C. Smith, and K.-C. Wang, "Speaker-aware speech enhancement with self-attention," in *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 486–490.

[61] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[62] W. Yu, J. Zhou, H. Wang, and L. Tao, "SETransformer: speech enhancement transformer," *Cognitive Computation*, vol. 14, no. 3, pp. 1152–1158, 2022.

[63] Y. Koizumi, K. Yatabe, M. Delcroix, Y. Masuyama, and D. Takeuchi, "Speech enhancement using self-adaptation and multi-head self-attention," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 181–185.

[64] C.-F. Liao, Y. Tsao, X. Lu, and H. Kawai, "Incorporating symbolic sequential modeling for speech enhancement," *arXiv preprint arXiv:1904.13142*, 2019.

[65] X. Xu and J. Hao, "U-former: Improving monaural speech enhancement with multi-head self and cross attention," *arXiv preprint arXiv:2205.08681*, 2022.

[66] 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, no. 05, 2020, pp. 9458–9465.

[67] A. E. Bulut and K. Koishida, "Low-latency single channel speech enhancement using U-Net convolutional neural networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6214–6218.

[68] F. Deng, T. Jiang, X. Wang, C. Zhang, and Y. Li, "NAAGN: Noise-aware attention-gated network for speech enhancement." in *INTERSPEECH*, 2020, pp. 2457–2461.

[69] J. Kim, M. El-Khamy, and J. Lee, "T-GSA: Transformer with gaussian-weighted self-attention for speech enhancement," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6649–6653.

[70] L. Zhou, Y. Gao, Z. Wang, J. Li, and W. Zhang, "Complex spectral mapping with attention based convolution recurrent neural network for speech enhancement," *arXiv preprint arXiv:2104.05267*, 2021.