Resource efficient deep learning approaches for monaural speech separation

Peiran Shi

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

August 2024

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Peiran Shi

Entitled: Resource efficient deep learning approaches for monaural speech separation

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:

Dr. William Lynch — Chair

Dr. William Lynch — Examiner

Dr. M. Omair Ahmad — Examiner

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

Dr. Mirco Ravanelli — Thesis Supervisor(s)

Approved by

Dr. Jun Cai, GPD — Chair of Department or Graduate Program Director

Dr. Mourad Debabbi — Dean of Gina Cody School of Engine

Abstract

Resource efficient deep learning approaches for monaural speech separation

Peiran Shi

Speech separation is a critical task in processing naturalistic audio streams, aiming to extract individual speech sources from mixed speech signals. Monaural speech separation, which deals with audio from a single microphone, focuses on isolating overlapping speech signals, a process essential for applications such as automatic speech recognition and voice assistant devices. Recent advances in deep learning have significantly improved speech separation, typically by training neural networks to estimate high-quality separated speech from mixed signals using supervised learning. However, most state-of-the-art neural networks operate in the time domain and are computationally expensive due to their sequential processing methods and complex structures. Despite the common perception that time-domain models outperform those in the time-frequency domain, this thesis focuses on developing resource-efficient models in the time-frequency domain, aiming to enhance their performance within a deep learning framework.

In the first contribution of this thesis, we propose RCFormer, a Conformer-based neural network with a redundancy approach, designed for monaural two-speaker speech separation. The RCFormer employs multiple pairs of intra-frame and sub-band Conformer blocks to successively capture both frame-level and sub-band-level information from the input spectrogram. To address the challenge of sparse information in the input spectrogram, a redundancy approach is introduced to create a denser representation by stacking the input spectrogram embeddings. The proposed architecture integrates Conformer blocks between a dilated dense convolutional encoder and decoder, with the Conformer block outputs fed into a masking module that generates masks to filter the encoder outputs, which are then transformed into separated speech signals via the decoder. Extensive experiments demonstrate

that RCFormer achieves competitive, and often superior, performance compared to existing state-of-the-art methods across all evaluation metrics, while also featuring significantly fewer trainable parameters.

While many models achieve competitive performance with fewer trainable parameters, few researchers have addressed the computational workload and processing time associated with these models. In the second contribution of this thesis, we propose FSBNet for two-speaker speech separation, which integrates sub-band and full-band modules. FSBNet consists of an encoder, multiple full-band and sub-band blocks (FSB blocks), and a decoder. The FSB block features a sub-band module that extracts temporal information within each sub-band and computes high-level cross-band dependencies through compact latent summaries, and a full-band module that captures long-range dependencies across the entire spectrogram using a self-attention mechanism. The contextual information obtained from the FSB blocks is then processed into two complex spectrograms representing the separated speech signals, which are re-synthesized into audio using the inverse short-time Fourier transform (ISTFT). Experimental results demonstrate that FSBNet achieves competitive performance compared to both time-domain and time-frequency domain approaches, with significant improvements in model size reduction and processing time efficiency. Notably, this architecture outperforms most efficient time-domain models for the first time since 2019.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisors, Prof. Wei-Ping Zhu and Prof. Mirco Ravanelli, for their invaluable guidance, unwavering support, and encouragement throughout my academic journey. Their teachings in theoretical knowledge and insightful suggestions have been instrumental in shaping my research and overcoming challenges along the way.

I would also like to extend my heartfelt thanks to all the laboratory members and friends for their kindness and support during my master's studies. Their assistance and camaraderie have made this journey both productive and enjoyable.

A special thank you goes to the SpeechBrain toolkit, developed by Prof. Mirco Ravanelli's group, which played a crucial role in the success of my experiments.

Lastly, I wish to express my deepest love and appreciation to my parents. Their selfless love, encouragement, and unwavering support have been my source of strength, helping me to overcome all obstacles and disappointments in life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Speech separation is a crucial front-end task in processing naturalistic audio streams. It aims to extract individual speech sources from a mixed signal. As a special scenario of general source separation [1], [2], which deals with a variety of interferences such as music or environmental noise, speech separation specifically focuses on separating overlapping speech signals in a given mixed speech signal [3], [4]. This process is vital for collecting clean and clear speech data from the environment, supporting downstream tasks including automatic speech recognition, voice assistant devices, and other audio systems.

Monaural speech separation [3], [5], also known as single-channel speech separation, is the most widely studied branch of speech separation. It addresses audio captured from a single microphone, making the task particularly challenging due to the lack of spatial information that could otherwise help differentiate between speakers [6]. Initial solutions to this challenge relied heavily on signal processing and statistical techniques. Conventional signal processing methods such as spectral subtraction [7], computational auditory scene analysis [8], and time-frequency masking used heuristic and knowledge-based information to extract and segment different speaker streams [9]. Statistical methods like Independent Component Analysis (ICA) [10], matrix factorization [11], and hidden Markov models (HMM) [12] were also employed to model the mixture speech and improve performance. However, these conventional methods heavily rely on handcrafted features, which have several limitations: first, they are less effective in diverse datasets and real-world environments, and they struggle to adapt to new speakers. Second, feature engineering is computationally intensive

and requires domain expertise. Moreover, both feature engineering and statistical methods are often speaker-dependent [12]. It can only works on very small range of samples.

In recent years, deep learning has become highly popular in various fields, including speech separation. Its data-driven nature makes it effective for solving prediction and estimation problems, both with and without supervision. This advancement has significantly improved speech separation results. Deep learning techniques such as fully-connected neural networks, convolutional neural networks, recurrent neural networks, and attention-based Transformer networks have been extensively applied to speech separation, enhancement, and other speech processing tasks. Inspired by developments in computer vision and natural language processing, some methods from these fields have been adapted for speech processing. [13] Given that speech signals are one-dimensional sequences with rich temporal information, sequence models from NLP and pattern recognition models from computer vision can be utilized. However, the unique challenges of speech separation necessitate specialized approaches tailored to this domain.

Deep learning methods for speech separation can be divided into time-domain and time-frequency-domain models. In time-frequency domain models, the input mixture is transformed into a spectrogram using short-time Fourier transform (STFT). The complex spectrogram or its magnitude component is then used to train the separation model. The enhanced spectrogram is subsequently used to reconstruct individual speeches via inverse short-time Fourier transform (ISTFT). In contrast, time-domain models directly process the time-domain waveform of the mixed speech signal. This approach involves down-sampling the input waveform with a one-dimensional convolutional layer, passing it through the separation network, and finally up-sampling it with a one-dimensional convolution-transpose layer to produce the separated speeches.

This thesis investigates resource efficient neural networks for single-channel, two-speaker speech separation, emphasizing time-frequency domain methods to minimize computational demands. To tackle the challenges of long-range dependencies in speech signals and the sparsity of spectrograms, we develop tailored model structures and neural network modules that extract essential information. These models achieve competitive performance against both time-frequency and time-domain approaches. The following sections provide an overview of existing speech separation methods and key components incorporated into our approach.

1.1 Neural networks for speech separation

1.1.1 Time domain models for speech separation

The time domain models directly estimate the separated speech signals from the mixture speech waveform. The problem of single-channel speech separation task in time domain can be formulated in terms of estimating S sources $x_1(t), x_2(t), ..., x_s(t) \in R^{1\times T}$, given the discrete waveform $x(t) \in R^{1\times T}$, where

$$y(t) = \sum_{i=1}^{S} x_i(t) \tag{1}$$

where T indicates the length of speech waveform and S indicates the number of speakers in the mixture. We aim to directly estimate $x_i(t), i = 1, 2, ..., S$, from $y(t)$.

Many existing time domain models applies an encoder-decoder based masking approach, which contains an encoder, a decoder, and the masking module, as shown in Fig.1.1. The encoder block estimates a learnable representation h for the one-dimensional long sequence input mixture x. The masking module estimates optimal masks m_1, m_2 to separate the sources present in the mixtures. The decoder finally reconstructs the estimated sources $\hat{x}_1$,$\hat{x}_2$ by multiplying the masks and the encoded mixture representation.

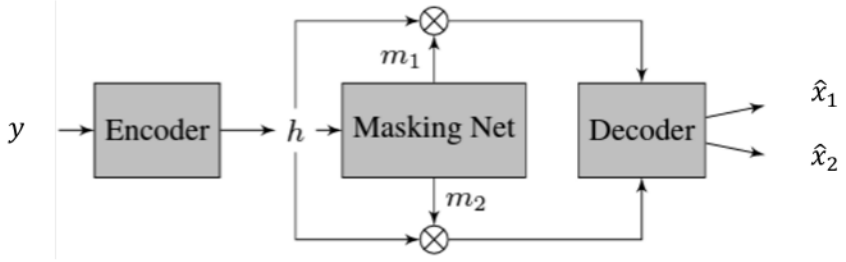


Figure 1.1: Illustration of encoder-decoder based masking approach [14]

CNN-based models

The convolutional neural network (CNN) was initially proposed for visual recognition tasks [15], designed to extract spatial features from input data. It consists of two main components: convolutional layers and pooling layers. The convolutional layers apply a series of filters that slide

over the input data, performing convolution operations to produce feature maps. The pooling layers then reduce the dimensionality of these feature maps and enlarge the receptive field by performing down-sampling.

In time-domain speech separation, although the input mixture is a one-dimensional long sequence rather than an image, CNNs can still be effectively used. By employing a learnable encoder, the CNNs can transform the input into a more image-like representation, allowing it to leverage its strengths in feature extraction. Many CNN-based structures which are proved to be effective in computer vision are also be applied in speech separation. For example, the authors of [16] introduced a novel end-to-end approach for speech separation based on Temporal Convolutional Networks (TCNs), named FurcaPy. Originally applied in action segmentation [17], TCNs are designed to capture long-range dependencies within input sequences by utilizing stacked dilated 1-D convolutional layers with an exponentially increasing dilation factor, thereby expanding the receptive field. Based on the TCN framework, FurcaPy incorporates a pyramid-like structure composed of three distinct gated TCNs, complemented by a ”weightor” module. This ”weightor” network dynamically determines the weights of the different gated TCNs for each individual utterance, allowing the model to adaptively emphasize the most relevant temporal features for effective speech separation.

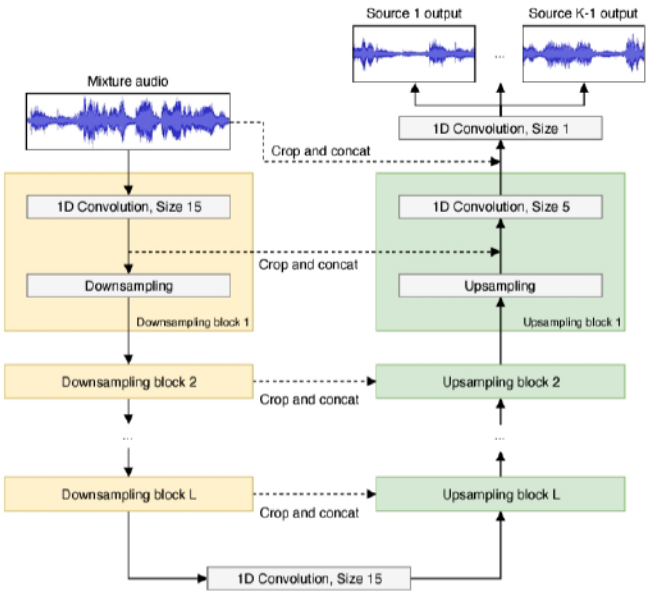


Figure 1.2: The structure of the Wave-U-net system [18]

Another widely recognized architecture, known as U-Net, was originally developed for biomedical image segmentation [19]. This structure has demonstrated impressive performance in the domain of speech separation as well. As shown in Fig.1.2, the authors of [18] introduce a U-Net-based architecture that employs iterative resampling of feature maps to capture and integrate features across different time scales. Additionally, several enhancements have been implemented, including an output layer designed to enforce source additivity, an advanced up-sampling technique, and a context-aware prediction framework aimed at minimizing output artifacts.

Unlike the models in [16] and [18], which either incorporate feed-forward networks or rely on the less robust U-Net architecture, the authors of [20] proposed Conv-TasNet, a totally CNN-based model for speech separation that represents the *state-of-the-art* in CNN-based approaches. As shown in Fig.1.3, this model shares a similar high-level structure with the masking approach, consisting of an encoder, a separator based on masking, and a decoder. The encoder adopts a 1-D neural network to generate a high-dimensional representation of the input. The separator is composed of stacked dilated 1-D blocks with an exponentially increasing dilation factor, which enlarges the receptive field. This ensures a sufficiently large temporal context window, allowing

the model to capture the long-range dependencies of the speech signal. The separator is applied to estimate the masks and then perform element-wise multiplication between the masks and the encoded input. Finally, the decoder, which also uses a 1-D neural network, reconstructs the final separated speeches.

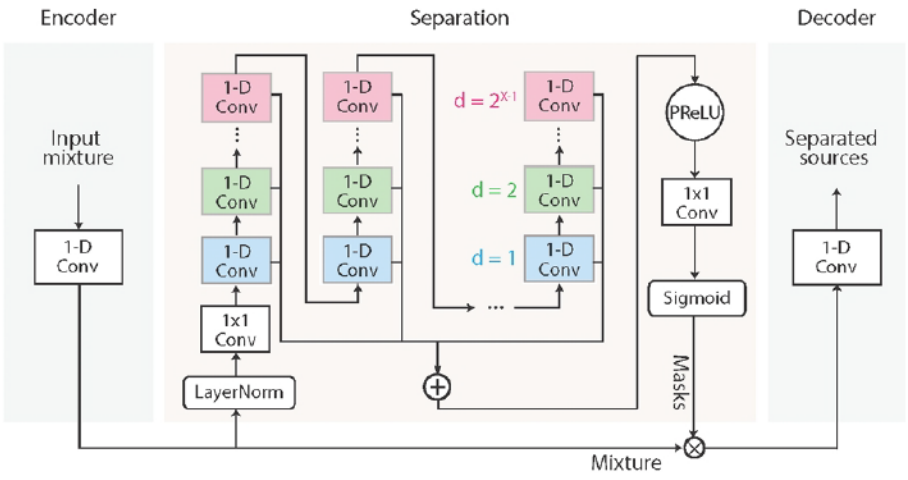


Figure 1.3: The flowchart of the Conv-TasNet system [20]

RNN-based models

Recurrent Neural Networks (RNNs), including variants like Gated Recurrent Units (GRUs) [21] and Long Short-Term Memory (LSTM) networks [22], are well-suited for processing long sequences of data due to their inherent memory mechanism. This mechanism involves connections that form directed cycles, allowing RNNs to maintain a memory of previous inputs and effectively capture temporal dependencies within the data. Unlike CNNs, which require multiple deep layers to capture high-level global information due to their local convolution operations, RNNs can process speech representations in a single pass, making them more resource efficeint and convenient to train. Consequently, some researchers have adopted RNNs to directly capture the long-term dependencies in input speech.

The authors proposed the Dual-Path Recurrent Neural Network (DPRNN) [23], an efficient method for modeling long sequences in speech separation tasks, which is the *state-of-the-art* model based on RNNs. As illustrated in Fig.1.4, DPRNN combines RNNs with a dual-path structure to

handle long sequential inputs in a straightforward manner. The DPRNN framework consists of three main stages: segmentation, block processing, and overlap-add. First, in segmentation stage, the long sequence input is divided into overlapping chunks, which are then concatenated to form a 3-D tensor. Then, the block processing stage involves applying several DPRNN blocks iteratively to extract both local and global features. Each DPRNN block contains two bi-directional RNNs: an intra-chunk RNN and an inter-chunk RNN. The intra-chunk RNN processes local information by operating on each chunk independently and in parallel, while the inter-chunk RNN captures global dependencies by processing information across different chunks. Finally, the overlap-add method is used to reconstruct the output into a continuous sequence, resulting in the separated speech signals. A mask approach mentioned Fig.1.1 is also adopted in DPRNN to obtain the final reconstructed speeches.

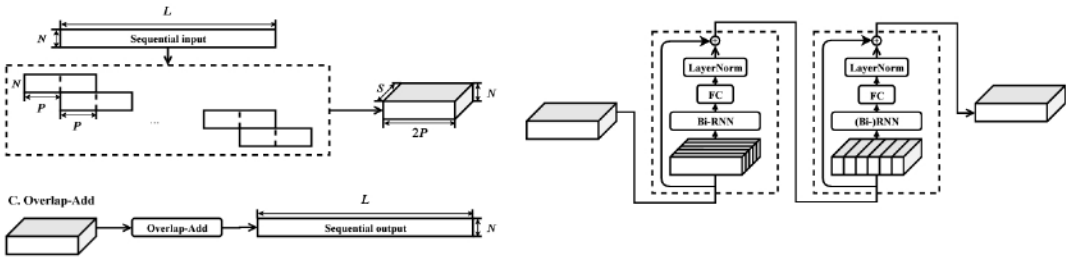


Figure 1.4: The flowchart of the DPRNN system [23]

Inspired by the DPRNN model, some researchers have integrated various extensions of RNNs into the dual-path structure, achieving notable performance improvements [24], [25]. While this approach retains a structure similar to DPRNN, the authors employ a more complex Bi-LSTM [26] network, which enhances performance further.

Attention-based models

The attention mechanism is first proposed to improve the performance of sequence-to-sequence models for tasks like machine translation [27]. It allows the model to dynamically focus on different parts of the input sequence when producing each output element. Unlike typical neural network which processes the entire input uniformly, the attention mechanism enables the model to access different parts of the input depending on their relevance to the current output. Attention mechanisms

compute a set of weights that determine the degree of focus on each part of the input. These weights are generally derived from the similarity between the current output and each input state. The greater the similarity between the input and output states, the higher the attention weight assigned, enabling the model to prioritize more relevant information effectively.

In speech separation and enhancement tasks, the attention mechanism is usually regarded as an effective improvement method that can be integrated into various structures. For example, some researches integrate the attention mechanism with the above mentioned Wave-U-Net [18], named attention Wave U-Net [28]. As shown in Fig.1.5, it consists of several down-sampling blocks, followed by one convolutional layer at the bottom , and followed by a series of up-sampling blocks with skip connections from the down-sampling blocks to the up-sampling blocks. Among the model, $C_i, i = 1, ..., d$ represents for the output of convolution in the ith down-sampling block and U_{d-i} represents for the output in the ith up-sampling block, where d indicates the depth of the U-Net.

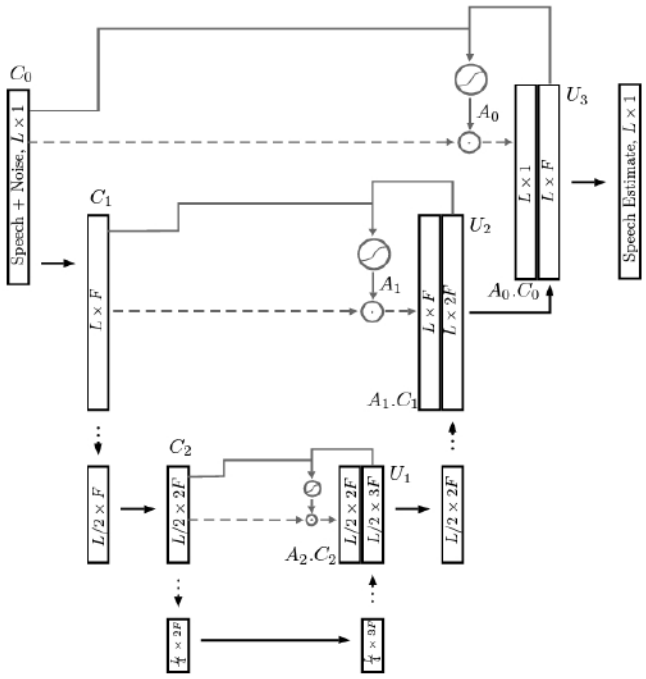


Figure 1.5: Attention Wave-U-Net architecture [28]

Different from the conventional U-shaped CNN, authors apply attention gates to identify the

relevant features between down-sampling block C_i and up-sampling block U_{d-i} by multiplying it with an attention mask. As shown in Fig. 1.6, first, the attention mask is calculated by W_x^{att}, W_g^{att}, and W_f^{att}. Then, the term-wise product between the attention mask and the down-sampling block is computed. Finally, the computation result concatenates with the up-sampling block U_{d-i}. This attention mechanism emphasizes important features in the input, rather than simply concatenating features at the same hierarchical level across the up-sampling blocks.

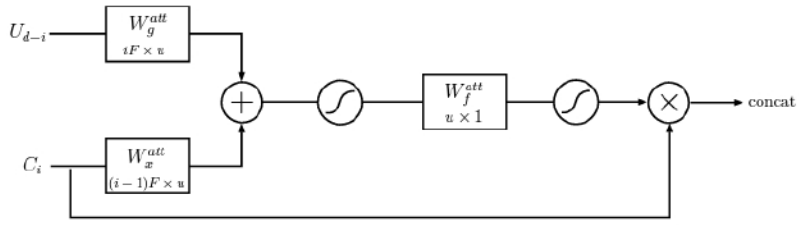


Figure 1.6: The Attention Mechanism in Attention Wave-U-Net [28]

However, the attention-based U-shaped CNN still inherits some drawbacks of conventional CNNs, particularly the need for deep layers to effectively extract long-term dependency of speech signals. This leads to a model that is complex, potentially unstable, and prone to over-fitting. In 2017, authors proposed a RNN-free model aiming at processing long range sequential data by using a self-attention mechanism, named Transformer [29]. As shown in Fig.1.7, The Transformer utilizes an encoder-decoder architecture. The encoder converts inputs into feature embeddings, which the decoder then transforms into outputs. Operating in an auto-regressive manner, the Transformer processes an input sequence of symbol representations. At each step, it generates a probability distribution for one symbol, which is used as additional input for the decoder to produce the subsequent symbol in the sequence.

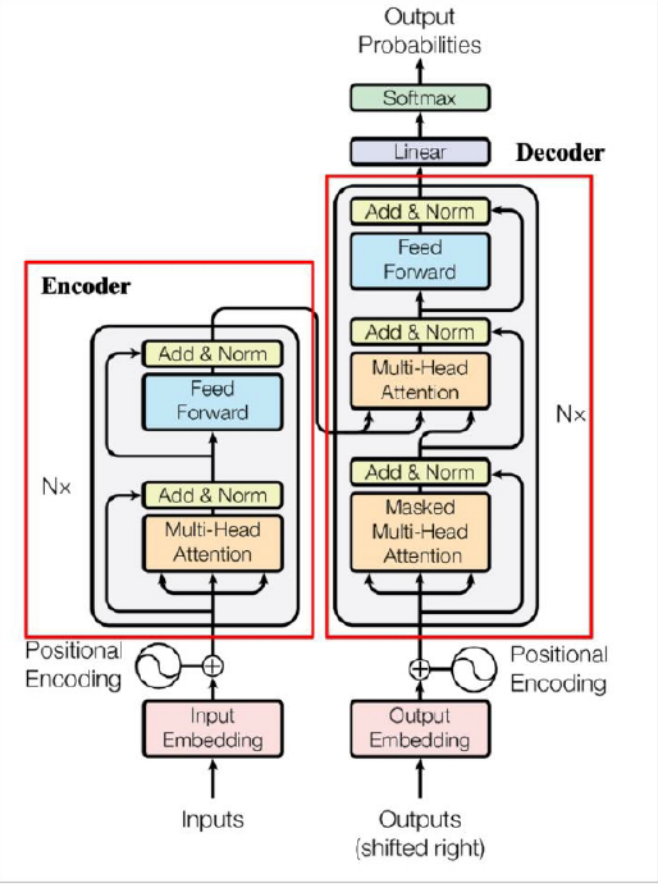


Figure 1.7: The architecture of Transformer [29]

The multi-head self-attention mechanism is a fundamental component designed to process the entire input sequence in parallel, eliminating the need for recurrence. An attention function can be defined as a mapping from a query and a set of key-value pairs to an output, where the queries, keys, values, and output are all vectors. As shown in Fig.1.8(left), the attention function is computed by taking the dot products of the query with all keys across a set of queries simultaneously, which are packed into a matrix Q. Similarly, the keys and values are packed into matrices K and V, respectively. Instead of performing a single attention function, h parallel attention layers employed to linearly project the queries, keys, and values h times. The outputs from these parallel layers are then concatenated and projected once more, resulting in the final values, as shown in Fig.1.8(right). Multi-head attention enables the model to simultaneously focus on information from different representation sub-spaces at various positions within the sequence.

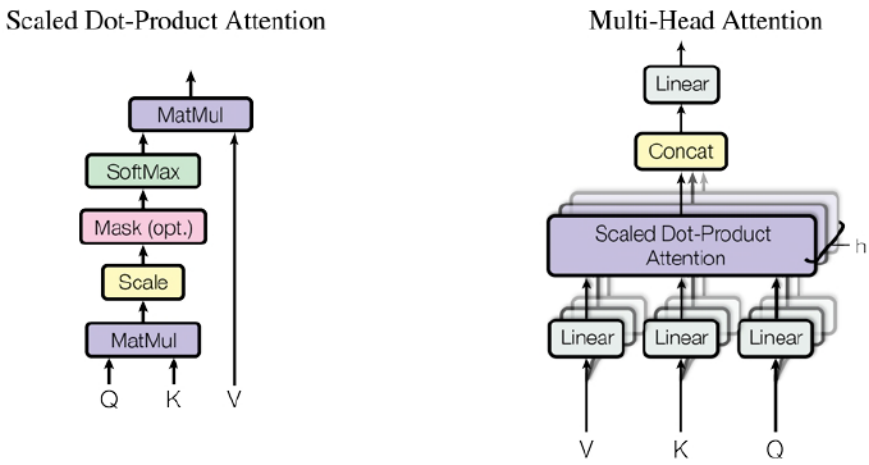


Figure 1.8: Scaled Dot-Product Attention (left). Multi-Head Attention consists of several attention layers running in parallel (right). [29]

Inspired by the remarkable effectiveness of the Transformer architecture in NLP tasks, and its superior performance compared to RNNs, authors of [14] integrated the Transformer encoder into a dual-path structure to propose a novel RNN-free, Transformer-based neural network for speech separation, named SepFormer. The SepFormer leverages a multi-scale approach within its attention-based Transformer encoder to capture both short-term and long-term dependencies. Unlike sequence-to-sequence translation tasks, speech separation merely requires direct transformation of a speech mixture into individual source speeches from different speakers, making the Transformer encoder sufficient for our task.

Like other time-domain speech separation models, SepFormer consists of an encoder, a masking network, and a decoder, with both the encoder and decoder implemented as 1-D convolutional layers. As illustrated in Fig.1.9, the masking network begins with layer normalization and a linear layer to improve feature trainability. It then chunks the 1-D features into overlapping segments and arranges them into 2-D features. The SepFormer block uses an intra-transformer to capture short-term dependencies within each chunk and an inter-transformer to model long-term relationships across chunks. After passing through several SepFormer blocks, features undergo PReLU activation, a linear layer, and an overlapping stage to return to 1-D. These are then processed by two fully

connected layers and a ReLU activation to generate separation masks for each speaker's signals.

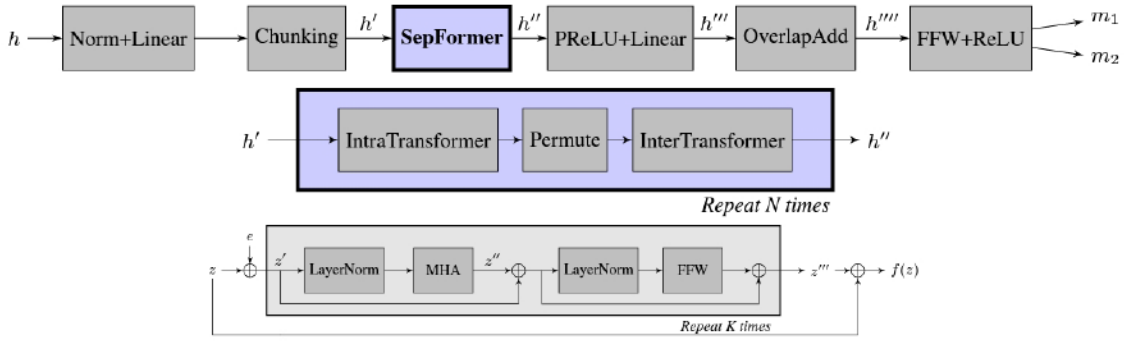


Figure 1.9: The architecture of SepFormer [14]

It is evident that SepFormer integrates the Transformer encoder in place of RNNs within a dual-path model, originally proposed in the above mentioned DPRNN. The impressive separation performance and training stability of SepFormer demonstrate the potential for widespread use of Transformer models in speech separation tasks.

1.1.2 Time-frequency domain models for speech separation

Different from time-domain models that directly estimate separated speech from the mixture waveform, time-frequency(T-F) domain models take the spectrogram features as the input. Spectrograms capture the essential harmonics of speech signals, which are crucial for effective speech separation. Considering a S-speaker mixture in time domain as $x_s(t), s = 1, ..., S$. The physical model in the time domain can be formulated as follows:

$$y[n] = \sum_{i=1}^{S} x^{(s)}[n] \tag{2}$$

where y denotes the mixture, and $x^{(s)}$ denotes source s, and n indexes N time samples. By using STFT, the physical model in T-F domain is formulated as:

$$Y(t, f) = \sum_{s=1}^{S} x^{(s)}(t, f) \tag{3}$$

where Y and $X^{(s)}$ respectively denote the complex spectra of y and $x^{(s)}$, t indexes T time frames, and f indexes F frequencies. S is assumed known in this study and given $Y(t,f)$, the goal of our task is to recover each source $X^{(s)}(t,f)$. We further represent the spectrogram of input mixture with Cartesian coordinate representation $Y = Y_{real} + jY_{img}$, where the spectrogram Y is decoupled into real part Y_{real} and imaginary part Y_{img}. The magnitude and phase features of the spectrogram can be written as follows:

$$Y_{mag} = \sqrt{Y_{real}^2 + Y_{img}^2} \tag{4}$$

$$Y_{phase} = \arctan\left(\frac{Y_{img}}{Y_{real}}\right) \tag{5}$$

To extract features and estimate individual speeches, researchers have proposed various methodologies in the T-F domain. These include approaches that focus solely on magnitude [5], [30], [31], those that address both magnitude and phase [32], [33], [34] and those that work with complex spectrogram [35]. Researchers have extensively investigated various methods to enhance feature extraction from spectrograms. Based on these approaches, several *state-of-the-art* models have been introduced, as follows.

Some researchers have processed only the magnitude spectra while disregarding phase information during the training of separation models. The phase information is typically utilized only in the reconstruction of the time-domain waveforms of the sources, as shown in Fig.1.10.

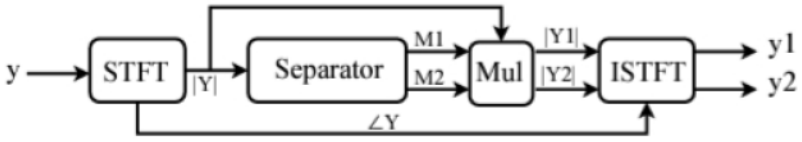


Figure 1.10: Magnitude estimation framework in T-F domain methods [36]

In supervised learning for speech separation, a notable challenge is the "label permutation problem," which arises due to the multiple valid ways to align separated outputs with reference signals.

To address this, the concept of permutation invariant training (PIT) [5] was introduced. As depicted in Fig.1.11, PIT involves computing the loss between the model's outputs and the reference sources for each possible permutation of the outputs. The permutation that results in the smallest loss is selected, and this minimum loss is used to update the model parameters.

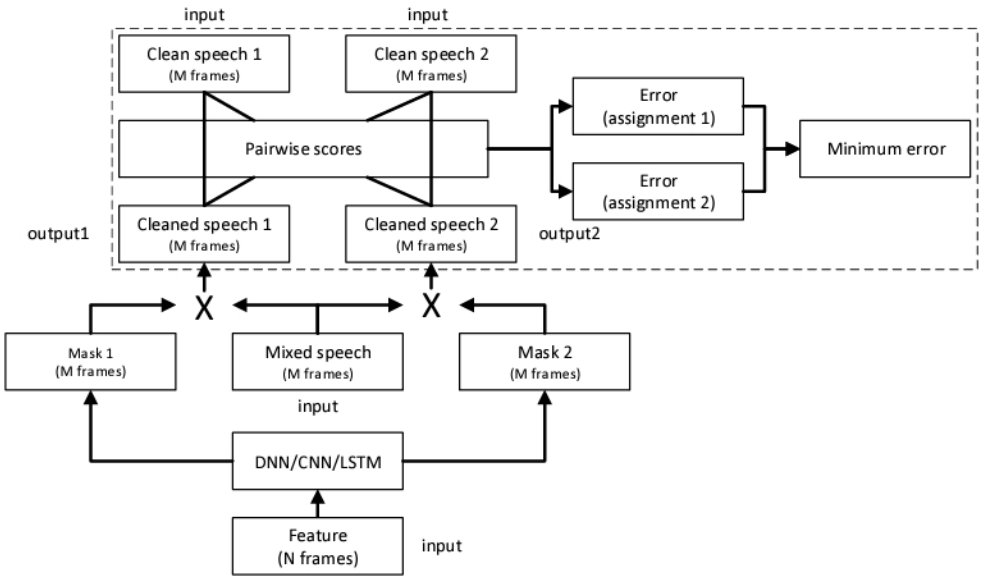
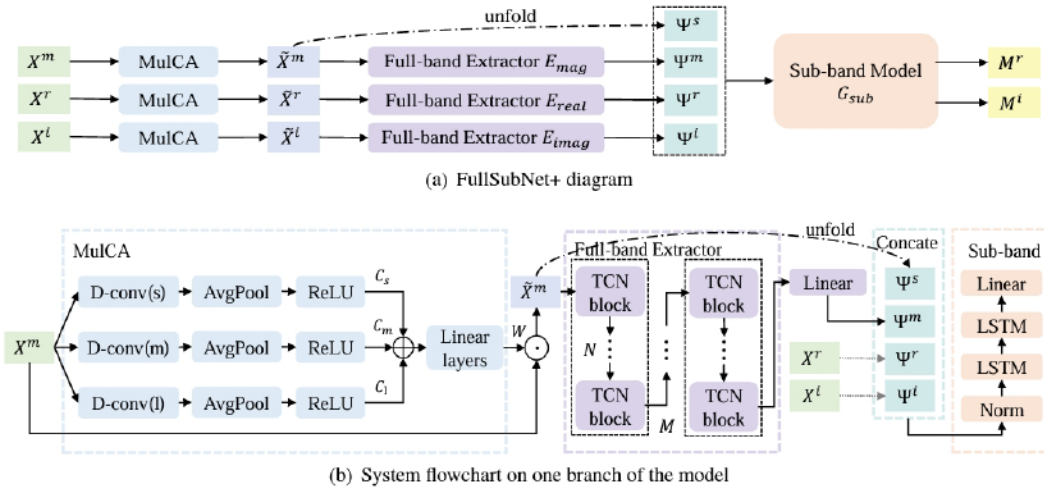


Figure 1.11: 2-speaker speech separation model with permutation invariant training [5]

The authors employed a simple feed-forward deep neural network (DNN) for feature learning and minimized the mean squared error (MSE) between the estimated and true magnitudes. By incorporating permutation invariant training (PIT), the speech separation model can be trained to be speaker-independent and effectively applied to various datasets, making it more suitable for real-world applications.

However, the phase component of the spectrogram remains crucial, as it provides information about the signal's temporal structure and fine-grained details. Specifically, it offers contextual insights into how different frequency components align over time, which is valuable for distinguishing overlapping speakers. Estimating phase directly is challenging due to its complexity and non-linearity. To address this, the authors proposed a model called FullSubNet+ [34], which uses

magnitude, real and imaginary spectrograms as inputs. By incorporating real and imaginary spectrograms, FullSubNet+ effectively makes full use of phase information. Although initially designed for speech enhancement, our experiments demonstrate that FullSubNet+ is also effective for speech separation. As shown in Fig.1.12, the inputs to the model are magnitude X^m, real component X^r and imaginary component X^i. First, a lightweight multi-scale time-sensitive channel attention (MulCA) module, comprising 1-D convolution layers with varying kernel sizes, average pooling, and the ReLU function, weights X^m, X^r, and X^i to focus on discriminative frequency bands. Next, stacked temporal convolutional network (TCN) blocks with exponentially increasing dilation factors capture long-range dependencies across the full band. Finally, two unidirectional LSTM layers and a fully connected layer predict masks for the enhanced speech signals. To boost performance, an $unfold$ function generates overlapped sub-bands of the input magnitude, which are fed into the network alongside the three outputs from the full-band extractor."



(a) FullSubNet+ diagram

(b) System flowchart on one branch of the model

Figure 1.12: The overall diagram and system flowchart on one branch of FullSubNet+ [34]

Integrating real and imaginary spectrograms with magnitude is an effective approach to improving performance in speech separation and enhancement tasks. However, it demands significant computational resources and training time. Since magnitude can be derived from the complex (real and imaginary) spectrogram, some researchers have focused on working directly with the 2-channel

complex spectrogram. This reduction of magnitude input allows for the use of more complex neural networks to enhance performance. The authors proposed a time-frequency (T-F) domain path scanning network (TFPSNet) [37]. Unlike dual-path models that only include intra-chunk and inter-chunk blocks, TFPSNet scans the complex spectrogram across frequency, time, and T-F paths, utilizing a Transformer architecture for modeling.

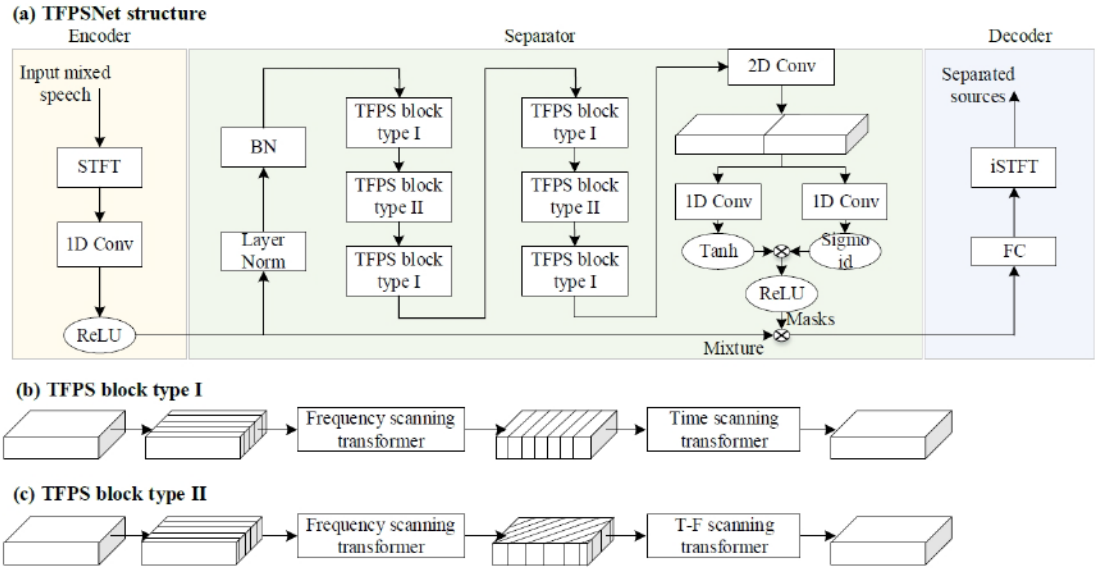


Figure 1.13: The overall architecture of TFPSNet [37]

As shown in Fig.1.13, The overall system includes an encoder, a masking-based separator, and a decoder, similar to time-domain models. The encoder first applies a Short-Time Fourier Transform (STFT) to convert the 1-D speech waveform into a 2-channel complex spectrogram. This spectrogram is then processed by a 1-D convolutional layer to produce a high-dimensional non-negative vector. In the masking-based separator, a set of masks is estimated from the encoded mixture using T-F domain path scanning (TFPS) blocks. Specifically, two types of TFPS blocks, each with three distinct T-F path scanning layers, are used to extract features. As shown in Fig.1.14, the frequency path scanning processes frequency bins within each frame, time path scanning processes time frames within each frequency bin independently, and T-F path scanning models transitions between adjacent frequency bins and frames along the diagonal. Each TFPS block uses a Transformer for path scanning. In the decoder, a fully connected layer (FC) reconstructs the separated speech

into 2 channels, followed by an ISTFT to obtain the final waveforms. TFPSNet, which focuses solely on complex spectrogram input, outperforms FullSubNet+ while maintaining a smaller model size.

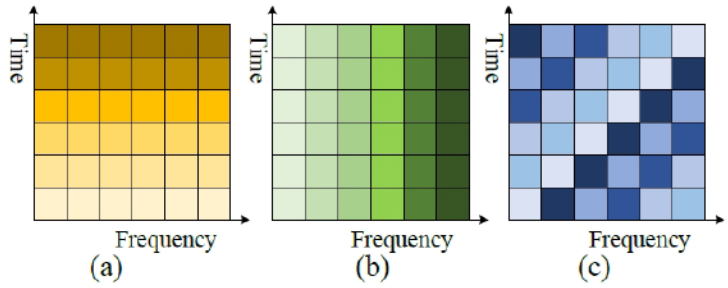


Figure 1.14: Illustration of path scanning in TFPS blocks. (a) Frequency path scanning (b) Time path scanning (c) T-F path scanning [37]

1.1.3 Efficient models for speech separation

While the quality of separated speech is the most crucial factor in evaluating speech separation models, efficiency is equally important, as it reflects the trade-offs between model performance and their potential for real-world applications. However, developing efficient models with low complexity is challenging due to the high-dimensional nature of speech signals, which consist of tens of thousands of time steps per second and exhibit long-range dependencies across multiple timescales [38]. The works discussed in Sections 1.1.1 and 1.1.2 each have limitations when it comes to model efficiency. RNNs, with time-recurrent mechanism, lack computational efficiency due to their inability to be parallelized during training. The Transformer architecture, which contains multiple feed-forward layers, significantly increasing both model size and processing time during training. CNN-based models offer advantages in terms of model size but are limited by the size of their receptive field, making it difficult to achieve global coherence [39].

Model efficiency is assessed primarily by two factors: model size and the computational resources required for training and inference. Researchers aiming to improve model efficiency typically employ strategies such as optimizing existing neural networks, designing novel and computationally efficient architectures, and working in the time-frequency (T-F) domain using spectrograms

as input.

The authors of [25] proposed a gated RNN-based dual-path model as a replacement for the original DPRNN [23]. Similarly, the authors of [28] introduced an attention mechanism to the original Wave-U-Net model [18]. Both approaches enhanced the performance of existing models while reducing model size by optimizing neural networks with novel methods. Building on this concept, the authors of [40] proposed an improved Transformer-based dual-path network, named DPTNet. DPTNet shares a similar structure with Sepformer [14], consisting of an encoder, a masking network, and a decoder [40]. The key difference lies in DPTNet's using of an improved Transformer, which replaces one linear layer in the feed-forward network with an recurrent neural network [41], as shown in Fig.1.15. This enhancement enables the improved Transformer to learn the order information of speech sequences without the need for positional encoding. As a result, the DPTNet model gains direct context-awareness for processing speech sequences [40].

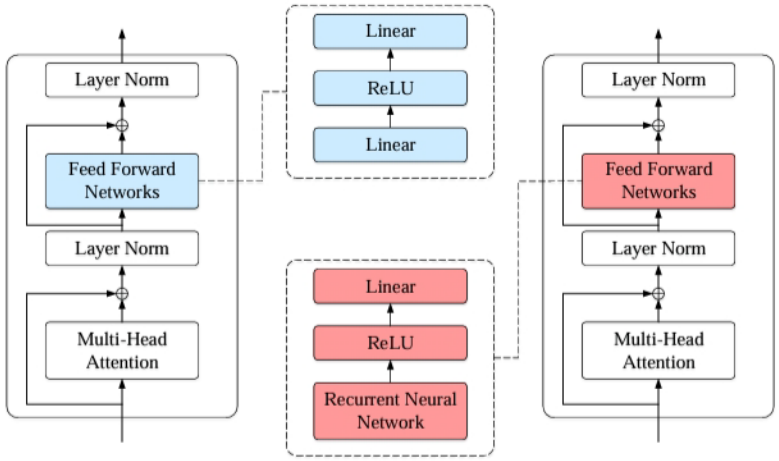


Figure 1.15: Architecture of the origin (left) and improved Transformers (right) [40]

The authors of [38], inspired by recent advances in neural state-space models (SSM) [42], introduced an efficient speech separation method called S4M (Speech Separation using State-Space

Model). S4M follows the mainstream encoder-decoder pipeline, as illustrated in Fig.1.16. Specifically, the encoder in S4M extracts multiple features at varying resolutions from a flat input mixture and feeds them into S4 blocks to capture representations with global long-range dependencies. Similarly, the S4 layer is employed in the decoder for feature reconstruction. S4M offers significant advantages over mainstream speech separation methods in terms of model complexity and computational cost, effectively capturing long-range dependencies for high-rate waveforms. This capability enhances the reconstruction of separated features, particularly in noisy conditions.

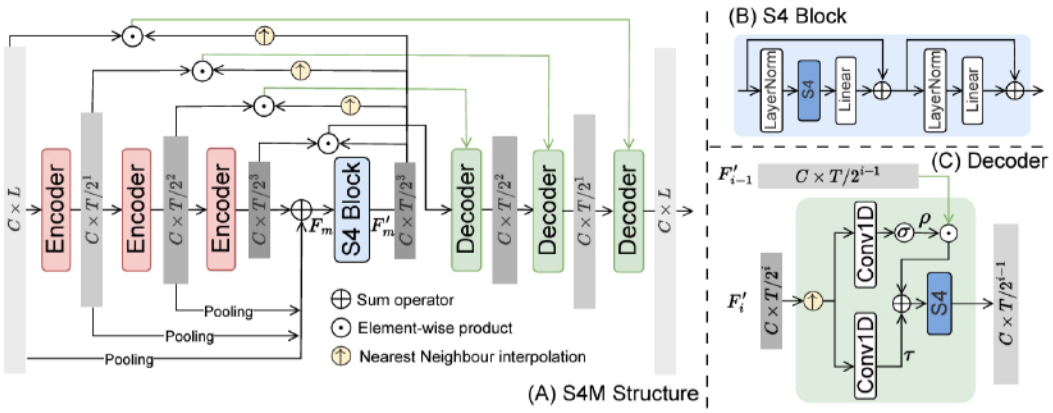


Figure 1.16: The block diagram of the (A) S4M model, (B) S4 Block, and (C) Decoder. [38]

The authors of [43] and [44] proposed a novel and computationally efficient structure called Skipping Memory Separator (SkiM). Inspired by the DPRNN [23], which uses inter-chunk blocks to model long-span features frame by frame, SkiM enhances efficiency by eliminating inter-chunk blocks and instead shares global-aware hidden and cell states across local networks. For long-span information modeling, the SkiM model skims the long sequence rather than analyzing it in detail, significantly reducing computational costs [43].

Similarly, RE-SepFormer applies a related module. As shown in Fig.1.17, RE-SepFormer replaces the inter-chunk Transformer blocks in the original SepFormer [14] with a summary representation based *MemoryTransformer* [44] blocks computed by averaging the tensor over the time axis. According to their experiments, this novel structure reduces memory usage by up to 28% for long sequences while maintaining performance. Additionally, this structure can be implemented in

a causal model, making it suitable for online applications.

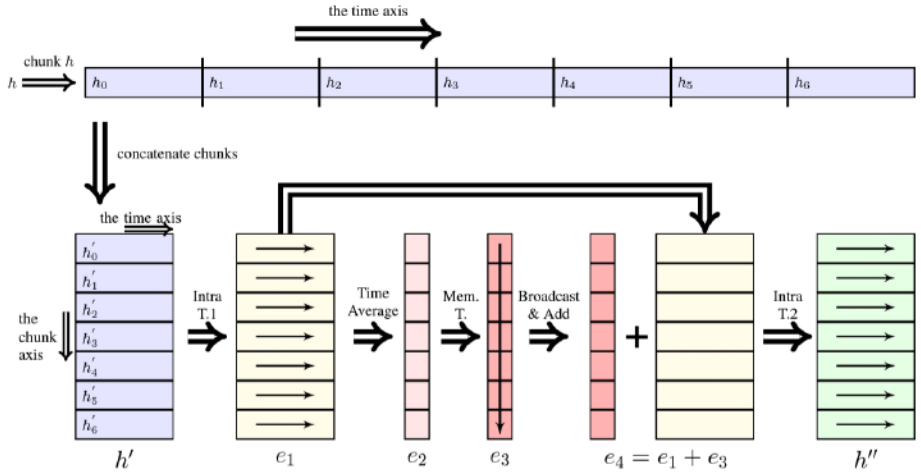


Figure 1.17: The The Resource-Efficient SepFormer (RE-SepFormer) module [44]

TFPSNet [37] and TF-GridNet [45] are the only two time-frequency (T-F) domain models that have achieved *state-of-the-art* performance so far. Both models use complex spectrograms as input. Unlike the pure dual-path structure of DPRNN [23], TFPSNet employs time scanning, frequency scanning, and time-frequency scanning to capture information across the entire spectrogram, while TF-GridNet integrates a dual-path structure with a full-band self-attention module [46], [47]. These models not only achieve *state-of-the-art* performance but also require only 10% of the model parameters compared to time-domain models. The success of TFPSNet and TF-GridNet demonstrates that using complex spectrogram input can provide sufficient information while maintaining a compact model size.

1.2 Evaluation of speech separation models

1.2.1 Datasets for speech separation

To assess the effectiveness of speech separation models, it is crucial to utilize a range of specialized speech datasets. These datasets are typically divided into three key subsets: training data, which is used to adjust model parameters; validation data, which helps in selecting optimal parameters; and testing data, which is employed to evaluate the final performance of the model. Speech

separation datasets consist of mixtures of fully overlapped speech from multiple speakers, along with the corresponding clean individual speech signals (targets). These datasets can be classified into two main categories: clean and noisy. Clean datasets feature speech mixtures that are free from any additional noise, providing a controlled environment for model training and evaluation. Noisy datasets include mixtures with added background noise, which introduces variability and complexity, thereby enabling models to be tested under more realistic conditions. Some commonly used speech datasets are introduced below.

WSJ0-2mix [48]: This dataset is a widely used benchmark for monaural, talker-independent speaker separation algorithms in anechoic conditions. The WSJ0-2mix dataset includes 20,000, 5,000, and 3,000 two-speaker mixtures for training, validation, and testing, respectively. The clean utterances are from the Wall Street Journal (WSJ0) corpus. In the training set, mixtures are created by randomly selecting utterances from different speakers and balancing gender representation. Each mixture features fully overlapping utterances with relative energy levels uniformly sampled from [−5, 5] dB, and a sampling rate of 8 kHz. The test set follows the same format but uses utterances from speakers not present in the training set, providing a more rigorous evaluation.

WHAM! and **WHAMR!** [49]: The WSJ0 Hipster Ambient Mixtures (WHAM!) and WSJ0 Hipster Ambient Mixtures with Reverberation (WHAMR!) are noisy speech separation datasets that pair each two-speaker mixture from the WSJ0-2mix dataset with unique background noise scenes. The WHAM noise recordings were collected in late 2018 from various urban locations around the San Francisco Bay Area, including restaurants, cafes, bars, and parks. WHAM! provides noisy versions of WSJ0-2mix, while WHAMR! adds reverberation effects.

LibriMix [50]: LibriMix is an open-source dataset for source separation in noisy environments, derived from the clean subset of LibriSpeech signals and WHAM noise. It includes two configurations: Libri2Mix and Libri3Mix, containing two-speaker and three-speaker mixtures, respectively. The Libri2Mix dataset has 50,800 mixtures for training, and 3,000 each for validation and testing. Mixtures have signal-to-noise ratios (SNRs) that follow a normal distribution, with a mean of 0 dB and a standard deviation of 4.1 dB in clean conditions, and a mean of −2 dB with a standard deviation of 3.6 dB in noisy conditions.

1.2.2 Evaluation metrics

To accurately assess the performance of speech separation models, robust evaluation methods are required to measure the quality of the estimated speech signals. These methods are generally classified into two categories: subjective listening tests and objective evaluation metrics. Subjective listening involves engaging a sufficient number of trained listeners to evaluate the quality of the separated speech. However, to ensure fairness and accuracy, the evaluation scores must be manually averaged, which is both time-consuming and prone to inaccuracies. Objective evaluation metrics offer a more efficient alternative, particularly for researchers. These metrics automatically assess the quality of the estimated speech by directly comparing it to the clean target speech, thereby providing a more reliable and consistent evaluation. We introduces some of the most commonly used objective evaluation metrics as follows:

Scale-Invariant Signal-to-Noise Ratio (SI-SNR) [51]: SI-SNR is utilized to measure speech quality by calculating a scale-invariant SNR value. Unlike conventional SNR, which requires prior setting of noise power, SI-SNR normalizes the amplitude of the signals, making it insensitive to volume differences. This metric is linearly correlated with speech quality—the higher the SI-SNR value, the better the quality of the speech. It is also widely used as the loss function in many speech separation models.

Perceptual Evaluation of Speech Quality (PESQ) [52]:The PESQ metric assesses the perceptual quality of processed speech, used for both separated and clean waveforms. It involves normalizing the estimated and clean speech signals to match voice energy levels, aligning them in time, and applying an auditory transformation to obtain loudness spectra. The difference between these spectra is then averaged over time and frequency to predict the subjective mean opinion score, with higher PESQ values indicating better performance.

Composite Mean Opinion Scores (MOSs) [53]: The MOSs consist of three components: CSIG, assessing signal distortion; CBAK, evaluating background noise distortion; and COVL, providing an overall speech quality rating. These components are typically combined into a composite objective measure that closely correlates with preset subjective ratings, offering insight into the different types of distortions in the estimated speech.

1.2.3 Evaluation for model efficiency

The objective of this thesis is to propose resource-efficient models for speech separation. Beyond evaluating performance, it is crucial to understand the trade-offs between performance and resource consumption. Assessing model efficiency involves examining how effectively a model performs relative to the resources it consumes. This includes a detailed evaluation of various metrics that reflect the model's size, computational and memory demands, as well as its speed during inference. The following key metrics are used to evaluate model efficiency in this thesis:

Number of parameters (#params): #Params reflects the model's complexity, specifically indicating the total count of trainable parameters. A lower number of parameters generally suggests a lighter model that is faster and requires fewer resources for training.

Giga Multiply-accumulate operations per second (GMACs/s) [54]: GMACs/s is a metric used to quantify the computational workload of a model, measuring the number of multiply-accumulate operations (MACs) a model performs, scaled to billions per second (Giga). Since MAC operations are fundamental in neural networks, models with lower GMACs/s are more efficient and preferable in scenarios where computational resources are limited.

Speed and Memory Utilization: Two critical metrics for assessing speed and memory utilization are memory cost and inference time. Memory cost refers to the amount of memory required to store the model and its intermediate computations, while inference time is the duration needed for the model to make predictions on a given input. To ensure a fair and meaningful comparison, memory cost and inference time should be evaluated under the same training environment, typically by conducting experiments on the same GPU.

1.3 Essential components for training

1.3.1 Activation function

Neural networks are considered complex functions that map inputs to outputs through high-level nonlinear representations. To enable the model to learn complex problems, nonlinear activation functions are employed, adding the necessary non-linearity to the network. Additionally, activation

functions must be differentiable to allow for the adjustment of weights and biases during backpropagation. The following sections introduce some of the most commonly used activation functions.

Rectified Linear Unit (ReLU) [55]: ReLU outputs the input directly if it is positive, and outputs zero for any negative input, as illustrated in Fig.1.18 (left). This selective activation allows neurons to fire only for positive inputs, introducing sparsity into the network and mitigating the gradient vanishing problem commonly encountered in deep networks. Although the gradient is technically undefined at zero, in practice, it is set to zero, which does not affect the backpropagation process.

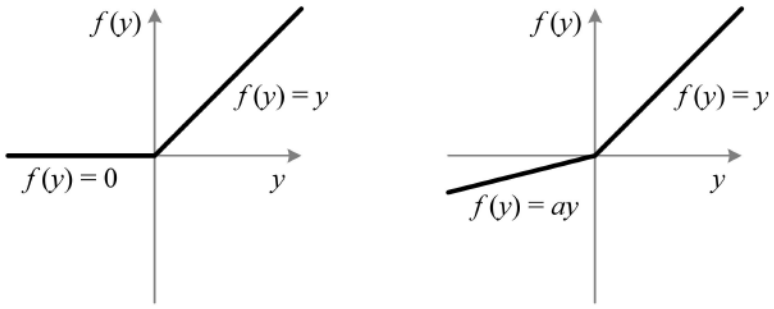


Figure 1.18: Illustration of ReLU function (left) and PReLU function (right) [56]

Parametric Rectified Linear Unit (PReLU) [56]: PReLU enhances the ReLU activation function by addressing the gradient vanishing issue with negative inputs. As illustrated in Fig.1.18 (right), PReLU multiplies negative inputs by a small, non-zero value, ensuring gradients are generated to update the model's weights and biases. Like ReLU, PReLU leaves positive inputs unchanged. The key difference is that this non-zero value in PReLU is a trainable parameter, enabling the network to learn the optimal slope for effective weight and bias adjustments.

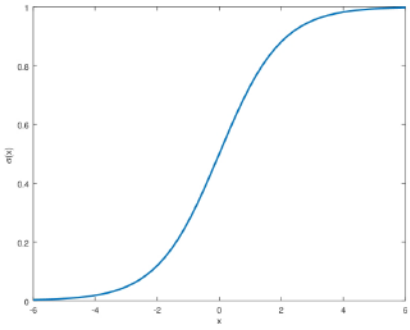


Figure 1.19: Illustration of Sigmoid Activation Function

Sigmoid Activation Function: The sigmoid function constrains outputs to a range between zero and one, as illustrated in Fig.1.19. It is a smooth function, but for large input values, the curve flattens significantly, causing the gradient to approach zero, leading to the gradient vanishing problem where weights and biases fail to update effectively. Moreover, the sigmoid function is computationally intensive due to its exponential nature.

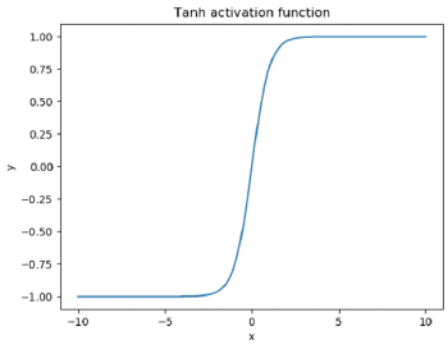


Figure 1.20: Illustration of Tanh Activation Function

Tangent hyperbolic function (Tanh): The Tanh activation function is a widely used nonlinear function in neural networks, particularly in tasks that require output values within a specific range, as shown in Fig.1.20. The tanh activation function maps input values to a range between −1 and 1, making it particularly useful in scenarios where both positive and negative activations are desirable. Its primary strength lies in its ability to introduce nonlinearity into the network, thereby enabling the modeling of complex relationships. This nonlinearity enhances the network's capacity to capture

intricate patterns within the data, which is crucial for solving complex tasks such as speech separation. Furthermore, by centering the data within the -1 to 1 range, the tanh function can reduce the bias shift effect, potentially leading to faster convergence during training.

1.3.2 Regularization

In deep learning-based approaches, it is crucial for trained models to perform well on unseen datasets drawn from the same distribution as the training data, a characteristic known as generalization ability. A common issue is that models often exhibit superior performance on the training data compared to the unseen testing data, leading to the problem of over-fitting. To mitigate over-fitting and enhance the model's generalization ability, regularization techniques are employed.

Dropout [57]: Dropout is a technique where certain neurons are randomly deactivated during training, temporarily removing their contributions during forward propagation and excluding their weight updates during backpropagation. Notably, dropout is only applied during training; all neurons remain fully active during testing. This method is widely used to prevent overfitting and promote faster model convergence.

Residual connection [58]: Residual connection is a type of skip-connection that learn residual functions relative to the layer inputs, rather than learning functions without reference to the inputs, as shown in Fig.1.21. During backpropagation, residual connections can effectively prevent the vanishing gradient problem caused by zero gradients. This technique is straightforward to implement in any neural network, enabling deeper architectures without adding additional parameters. Residual connections are now widely employed in CNNs, RNNs, attention-based transformers, and other models.

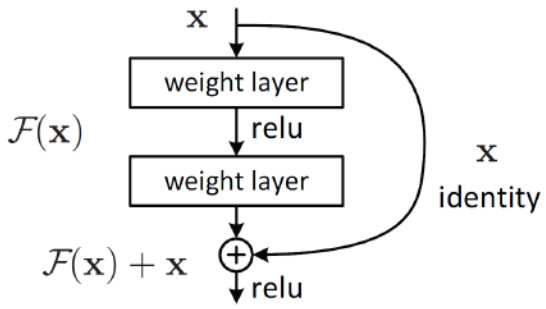


Figure 1.21: Illustration of Residual connection [58]

1.3.3 Loss function

To train the model effectively, it is essential to design a suitable loss function that measures the difference between the estimated outputs and the ground truth targets. The loss function guides the model in minimizing prediction errors during optimization. Since speech separation is a regression problem, this section will discuss several loss functions commonly used in this context. Note that, in speech separation, there are usually multiple pairs of estimated outputs and target signals, so the final loss is computed as the mean of the loss values across these pairs.

Mean squared error (MSE): The Mean Squared Error (MSE) loss, also known as L2 loss, is defined as the average of the sum of squared differences between the predicted values and the ground truth. In time-frequency domain models for speech separation, this loss function is frequently employed by researchers who directly minimize the MSE in the spectrogram format. Specifically, the loss is calculated between the estimated magnitude and the true magnitude (target) of the spectrogram, and can be mathematically expressed as follows [5]:

$$MSE_{loss} = \frac{1}{T \times F \times S} \sum_{s=1}^{S} \left\| |\tilde{X}_s| - |X_s| \right\|^2 \tag{6}$$

where T and F indicate the number of time frames and frequency bins, respectively. S represents the number of speakers in the input speech mixture. $\tilde{X}_s$ denotes the estimated magnitude and X_s denotes the magnitude target.

Scale invariant speech to noise ratio (SI-SNR) [51]: SI-SNR aimed at producing high-quality

and intelligible speech signals directly working on waveforms, by measuring the ratio of the signal power to the noise power in decibels, which is formulated as [20]:

$$\begin{cases} x_{\text{target}} = \dfrac{\langle \hat{x}, x \rangle x}{\|x\|^2}, \\ e_{\text{noise}} = \hat{x} - x_{\text{target}}, \\ \text{SI-SNR}_{loss} = 10 \log_{10} \dfrac{\|x_{\text{target}}\|^2}{\|e_{\text{noise}}\|^2}. \end{cases} \tag{7}$$

Where x and $\hat{x}$ denote the clean speech waveform and the estimated speech waveform. The SI-SNR loss computed using the scaled target source x_{target} and the noise e_{noise}, which is the difference between the estimated source signal and the target source signal. This formulation makes SI-SNR scale-invariant, meaning that its evaluation is independent of the amplitude of the signals.

1.4 Objective and organization of thesis

The objective of this thesis is to propose resource-efficient neural networks for single-channel speech separation. The first contribution introduces a Conformer-based two-stage model with an integrated redundancy approach, referred to as the Redundant Conformer Neural Network (RC-Former). This model comprises an encoder, separator, and masking decoder, following a structure commonly used in both time-domain and time-frequency (T-F) domain separation models. The encoder utilizes a dilated dense convolutional neural network (CNN) that extracts more information compared to a single-layer convolution. The separator is designed to successively learn intra-frame and sub-band features. To address the issue of sparse information in the spectrogram, a redundancy approach is employed, stacking the input spectrogram into a denser representation. The outputs from the transformer blocks are passed through a masking decoder to generate a mask, which is then element-wise multiplied with the encoder outputs to obtain the separated speech signals. Our model achieves impressive performance compared to existing systems, with fewer trainable parameters and faster computation speed.

Although the first contribution, RCFormer, achieves impressive performance while maintaining a small model size, it still requires a relatively large computational workload, resulting in a

high GMACs/s. In the second contribution, we introduce a novel neural network that integrates a sub-band module and a full-band module to address this issue. In the sub-band module, a compact latent summaries method is applied to extract information within each sub-band and across different sub-bands. The full-band module incorporates a novel self-attention-based mechanism that operates on the entire spectrogram. To further reduce computational complexity, we estimate the complex spectrogram mapping directly rather than estimating the magnitude spectrogram mapping. We demonstrate that this novel full-band and sub-band architecture achieves effectiveness comparable to most speech separation models while significantly reducing model size and achieving the lowest GMACs/s.

The rest of this thesis is organized as follows:

Chapter 2: This chapter starts with an introduction to the Conformer architecture, followed by a detailed description of the proposed two-stage redundant Conformer model for speech separation in the time-frequency domain. It concludes with the experimental results, covering performance outcomes, model efficiency, and an ablation study of the proposed models across different configurations.

Chapter 3: This chapter first describes the proposed novel sub-band and full-band modules, with a particular focus on how the network is designed to minimize computational resource usage. It then explains the integration of the full-band and sub-band modules. Finally, the chapter presents experimental results demonstrating that the proposed approach achieves competitive performance in speech separation compared to larger transformer-based and dual-path models, while significantly reducing computational complexity.

Chapter 4: This chapter concludes the thesis and suggests some directions for future work.

Chapter 2

Conformer-based neural network for speech separation

In this chapter, we propose a two-stage Conformer neural network integrating redundant units, abbreviated as RCFormer. This chapter is organized as follows. In Section 2.1, we overview the Conformer, which is the architecture that integrates the multi-head self-attention mechanism and convolution neural networks. In Section 2.2, we first introduce the two-stage structure for speech separation, and then describe the proposed RCFormer for single-channel speech separation in the time-frequency domain. In Section 2.3, we provide the experiment results and the evaluation of our proposed RCFormer.

2.1 Convolution-augmented transformer

The convolution-augmented transformer (Conformer) [59] was originally designed for automatic speech recognition. It efficiently captures both local and global dependencies in an audio sequence by integrating Transformer and convolutional neural networks. While the general transformer structure includes both encoder and decoder, our network uses only the Conformer encoder since the input mixtures and output sequences are of the same length in the separation task. Each Conformer block comprises four modules: a feed-forward module, a multi-head self-attention module, a convolution module, and a second feed-forward module, with residual connections at each

sub-layer to enhance network robustness.

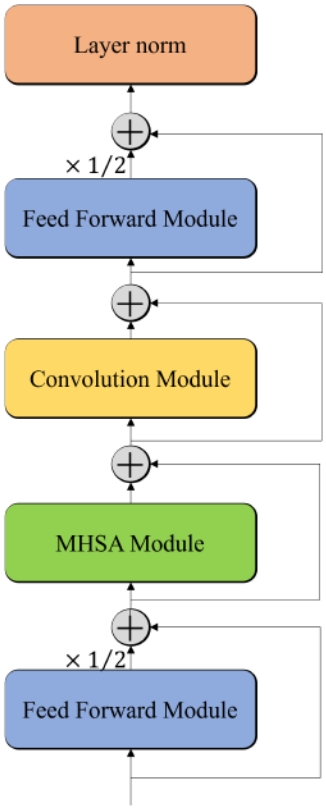


Figure 2.1: Illustration of Conformer block

2.1.1 Multi-head Self-attention Module (MHSA)

In the multi-head self-attention module (MHSA) [29], we employ a pre-norm residual unit with dropout and a multi-head attention module with relative positional encoding. As illustrated in Fig. 2.2, the pre-norm residual unit applies layer normalization before the sub-layer and the residual connection. The pre-norm method allows the subsequent layer to receive inputs with a consistent mean and variance and keeps the inputs to each layer in a moderate range. It stabilizes the training of our network and avoids the gradient vanishing problem.

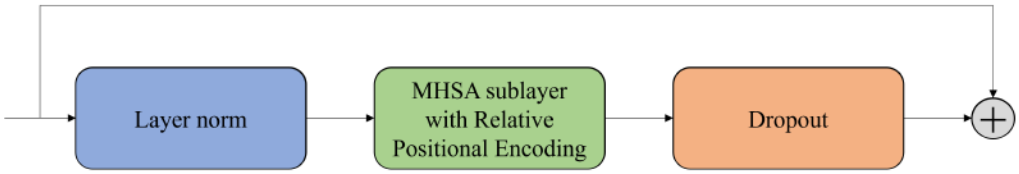


Figure 2.2: Illustration of multi-head self-attention module (MHSA)

The relative positional encoding method enhances the original self-attention mechanism by efficiently incorporating the relative positions or distances between sequence elements. Specifically, the MHSA sub-layer utilizes h attention heads. Each attention head operates on an input sequence $x = (x_1, x_2, \ldots, x_n)$ where $x_i \in R^{d_x}$, and the output is a new sequence with the same length $z = (z_1, z_2, \ldots, z_n)$ where $z_i \in R^{d_z}$. First, each input element x_i is computed with different, learnable linear transformations to get queries (Q), keys (K), and values (V), which are $x_i W^Q$, $x_i W^K$, and $x_i W^V$ respectively. $W^Q, W^K, W^V \in R^{d_x \times d_z}$ are parameter matrices. We also consider the edge between the input elements x_i and x_j, which are represented by corresponding keys and values vectors $a_{ij}^V, a_{ij}^K \in R^{d_a}$, where we set $d_a = d_z$. Then, a single element attention output z_i is computed as a weighted sum of linearly transformed input elements and propagates edge information as shown in Eq. (8) [29].

$$z_i = \sum_{j=1}^{n} \alpha_{ij}(x_j W^V + a_{ij}^V) \tag{8}$$

where $\alpha_i j$ is the weight coefficient, which is a softmax of a compatibility function that compares two input elements. After considering the edge information, $\alpha_i j$ is shown as Eq. (9) [29].

$$\alpha_{ij} = softmax(\frac{x_i W^Q (x_j W^K + a_{ij}^K)^T}{\sqrt{d_z}}) \tag{9}$$

2.1.2 Convolution module

The convolution module in Conformer integrates a gating mechanism, which is consisted of a pointwise convolution and gated linear unit (GLU). It is followed by a 1-D depthwise convolution layer [59]. Batch normalization [60] and a drop-out [57] is employed next to help the network

training and prevent overfitting, as shown in Fig.2.3.

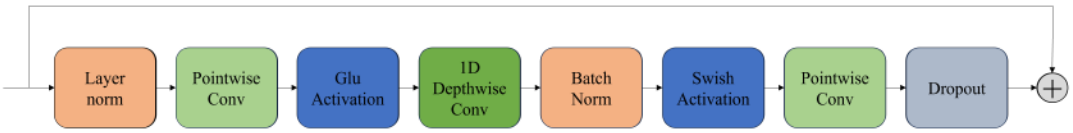


Figure 2.3: Illustration of convolution module

2.1.3 Feed forward module

The feed-forward module [61] consists of two linear transformations with a nonlinear activation in between, similar to the feed-forward layer in the vanilla Transformer. It also employs a pre-norm residual unit with dropout to enhance training stability, and uses the Swish activation function to regularize the neural network. Fig.2.4 below illustrates the feed forward module.

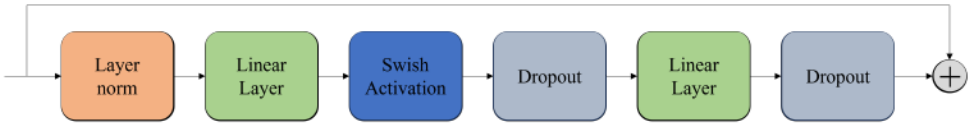


Figure 2.4: Illustration of feed forward module

2.1.4 Conformer block

The Conformer block features a 'sandwich' structure inspired by Macaron-Net [62]. It includes two half-step feed-forward layers surrounding the multi-head self-attention (MHSA) and convolution modules, as illustrated in Fig. 2.1. We replace the single feed-forward layer in the vanilla Transformer with two half-step feed-forward layers, positioned before and after the attention layer. We also employ half-step residual weights which is default $\frac{1}{2}$ in our feed forward module. A layer normalization is employed after the second half-step feed forward layer to normalize the output. Mathematically, for an input x_i to a Conformer block i, the output y_i would be [59]:

$$\tilde{x_i} = x_i + \frac{1}{2}FFN(x_i) \tag{10}$$

$$x_i' = \tilde{x_i} + MHSA(\tilde{x_i}) \tag{11}$$

$$x_i'' = x_i' + Conv(x_i') \tag{12}$$

$$y_i = LN(x_i'' + \frac{1}{2}FFN(x_i'')) \tag{13}$$

where FFN refers to the feed forward layer, $MHSA$ refers to the multi-head self-attention layer, $Conv$ refers to the convolution layer, and the LN refers to the layer normalization.

2.2 Proposed redundant Conformer neural network

In this section, we propose a Conformer-based speech separation model named redundant Conformer neural network (RCFormer). The RCFormer is a novel method that adopts the proposed two-stage conformer blocks containing intra-frame spectral module and sub-band temporal module to extract the frequency and time information of the spectrogram.

2.2.1 Two-stage structure

The two-stage structure, initially proposed to enhance RNN performance [23] in modeling long sequences for time-domain speech separation, is also used in the Transformer-based Sepformer [14]. This approach divides the long input sequence into smaller chunks, applying an intra-chunk network to capture dependencies within each chunk and an inter-chunk network to capture dependencies across chunks. We adapt this two-stage structure to the time-frequency domain by introducing a two-stage Conformer block. This block includes an intra-frame spectral module and a sub-band temporal module to capture frequency and time dependencies, respectively.

As shown in Fig.2.5, in two stage Conformer blocks, the input features Y have a shape of $[B, D, T, F]$, where B is the batch size, D is the channel size, and T and F represent for the number of time frames and frequency bins of the spectrogram respectively. In the intra-frame spectral module, we view the input as T separate sequences and each with length F. The Conformer block is applied to model the inter-frequency information within each frame. Then, a sub-band temporal module is applied to model the temporal information within each sub-band, where the input is viewed as F separate sequences and each with length T. Besides, the residual connection [58] is utilized into both two modules to avoid overfitting and gradient vanishing.

$$R_{intra-frame} = Conformer(Reshape(X)[:, t, :]) + Reshape(X) \tag{14}$$

$$R_{sub-band} = Conformer(Reshape(R_{intra-frame})[:, :, f]) + Reshape(R_{intra-frame}) \tag{15}$$

where $t = 1, 2, 3, \ldots, T$ denotes the index of time step, and $f = 1, 2, 3, \ldots, F$ denotes the index of frequency bin. $Reshape(\cdot)$ denotes the operation of dimension permutation.

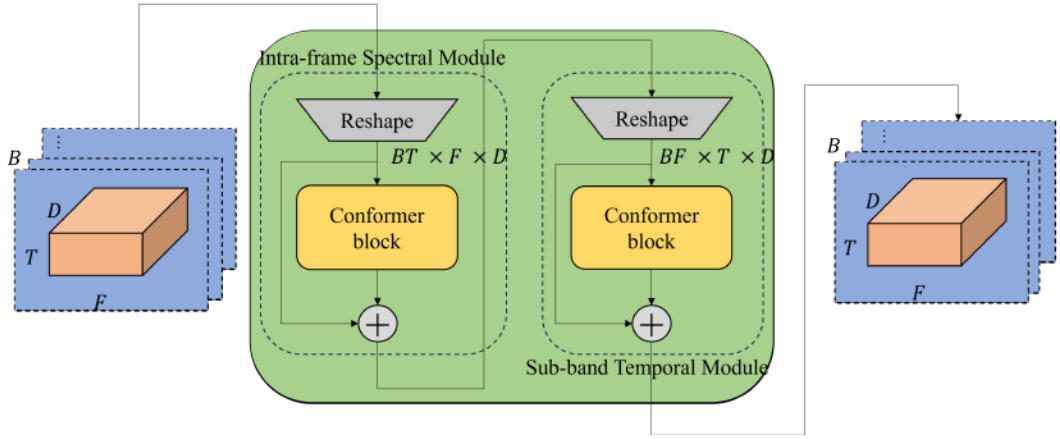


Figure 2.5: Illustration of Two stage structure

2.2.2 Proposed RCFormer

In general, the proposed RCFormer has three main components, including the encoder, conformer blocks, and the mask decoder. As shown in Fig.2.6, the noisy speech waveform is first processed by a short-time Fourier transformer (STFT) to obtain the input spectrogram. Then, the input spectrograms are first processed by a dilated densely convolutional encoder, which serves two primary functions. Firstly, it effectively extracts both low-level and global information from the spectrogram, and secondly, it significantly increases the receptive field, enhancing the learning of compressed features. These encoded inputs are then passed through a series of conformer blocks. The RCFormer employs two-stage conformer blocks to efficiently exploit local features and capture long-range dependencies. This approach includes both intra-frame conformer blocks and sub-band conformer blocks. Additionally, a redundant reconstruction method is applied to generate overlapping units, which improves performance while maintaining a relatively low parameter count.

Subsequently, the output from the conformer blocks is fed into the mask decoder, which is tasked with generating masks that estimate the speech features of different speakers. The mask decoder utilizes a densely convolutional neural network similar to that of the encoder and includes a masking module designed to predict masks that match the shape of the input spectrograms. Finally, the estimated masks are multiplied by the input spectrograms of mixed speech to obtain the separated spectrograms. These separated spectrograms are then processed by an Inverse Short-Time Fourier Transform (ISTFT), ultimately producing the separated speech waveforms.

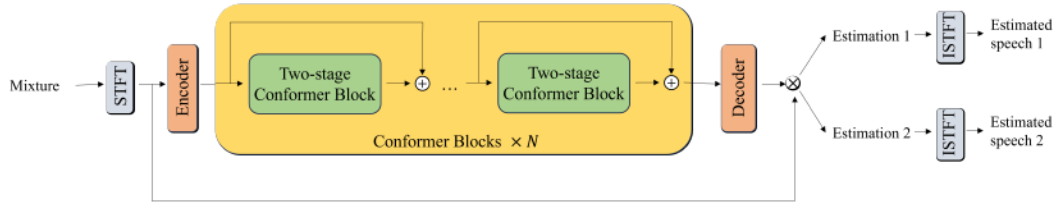


Figure 2.6: Overview of proposed RCFormer

Our RCFormer is proposed for time-frequency domain speech separation where the input is the spectrogram obtained by a short-time Fourier transformer (STFT). The original speech audio is obtained by a single-channel microphone and resampled at a certain sampling rate. For example, we can obtain 32000 samples by resampling a 4-second speech signal at $8kHz$. It is still difficult to process a long time-domain speech sequence although we can select small speech segments with a relatively small sample rate. As a result, we apply STFT to the original speech sequence. More specifically, given a S-speaker mixture with no noise $y[n]$, where n indexes N time samples. In our task, the number of speakers is 2. As a result, the row input mixture $Y \in R^N$. The physical model of a speaker separation task in the time domain is:

$$y[n] = \sum_{i=1}^{S} x^{(s)}[n] \tag{16}$$

In the time-frequency domain, the physical model is:

$$Y(t,f) = \sum_{i=1}^{S} X^{(s)}(t,f) \tag{17}$$

where t indexes T time frames and f indexes F frequency bins. After STFT, the row input is transformed into the 3-dimensional tensor $Y_f \in R^{2\times F\times T}$ and processed by the speech separation model. In the final stage, the Inverse short-time Fourier transformer (ISTFT) is used to inverse the estimated spectrogram of the two speeches into the two separated speech waveforms.

2.2.3 Encoder

The encoder is designed to extract the compressed features of inputs. As shown in Fig. 2.7, it is comprised of two convolution blocks and M dilated dense layers. Each convolution block consists of a convolution layer, an instance normalization, and a PReLU activation. In the first convolution block, the number of channels of inputs will first extend to D by the 2-dimensional convolution block with kernel size of $(1, 1)$. Then, M dilated dense layers are employed to extract both inside-band and cross-band features, which contain four convolution blocks with dense connections. Each convolution block consists of a constant padding, a 2-dimensional convolution layer, an instance normalization [63] and a PReLU [56] activation. Finally, the second convolution block is responsible for halving the frequency dimension to $F/2$ with D kernels of size $(1, 3)$ and a stride of $(1, 2)$, which reduces the computation complexity. The output tensor of the encoder is $X \in R^{D\times T\times(F/2)}$.

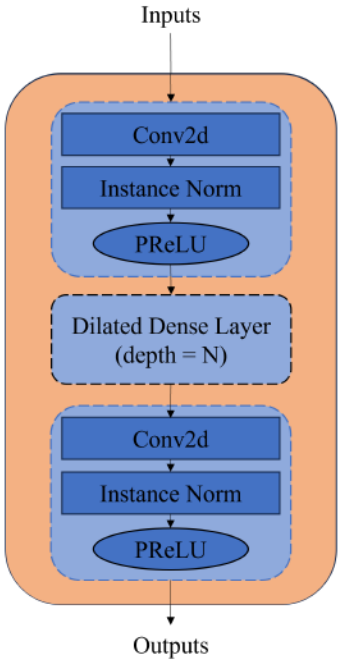


Figure 2.7: Illustration of Encoder

The dilated dense layer is inspired by the DenseNet [64], which is the densely connected convolution neural network. DenseNet is first applied in the music source separation task [65]. Different from conventional deep convolution neural networks, DenseNet connects each layer to every other layer in a feed-forward fashion. As shown in Fig.2.8 , the outputs of one block are passed to all subsequent blocks by concatenating the outputs of all previous blocks to final outputs. The conventional convolutions are also replaced by the dilated depth-wise separable convolutions, which expand the input by inserting gaps between its elements. The dilation rates of our proposed network are $1, 2, 4, 8$, which control how much the input is expanded. A constant padding is also added to maintain the spatial resolution of the feature maps and avoid cropping the input when applying dilated convolutions as shown in Fig.2.9. Each depth-wise convolution is followed by the Instance normalization [63] and PReLU [56] function.

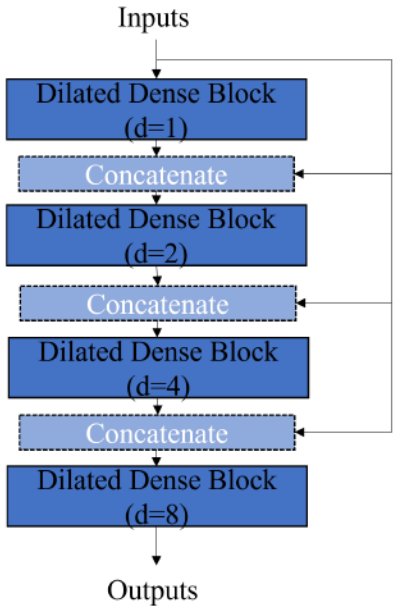


Figure 2.8: Illustration of dilated dense layer

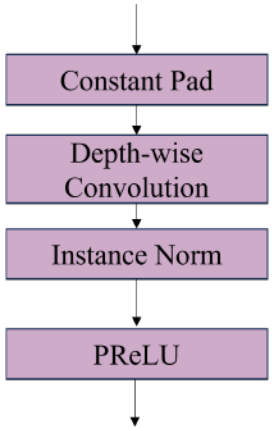


Figure 2.9: Illustration of dilated dense block

2.2.4 Redundant units

The redundant units are obtained from the encoded outputs of the dilated dense encoder through the "unfold" operation [66] and serve as the input to the conformer blocks. The "unfold" operation shares similarities with one-dimensional convolution, both containing parameters kernel size and

stride. However, the "unfold" operation does not perform any computation but reshapes the input tensor. Specifically, assume an input tensor X$\in$ $R^{D\times M\times N}$, for example, view X as N sequence vectors, each with length M. We take one sequence vector and extract the sliding blocks with kernel size I and stride size J, where the kernel size determines the size of the sliding window, and the stride size determines the steps the window slides. After zero padding, we can obtain the newly constructed redundant units $U \in R^{(I\times D)\times N\times(\frac{M-I}{J}+1)}$, where $(I \times D)$ is the total number of values within each block and $(\frac{M-I}{J} + 1))$ is the total number of such blocks. The output redundant units can be seen as collections of flattened blocks, where each block contains $(I \times D)$ values from the input. Note that the stride size J can be larger than one so that the sequence length of the redundant units and the amount of computation can be reduced. The "unfold" operation can be realized by torch.nn.Unfold [66]. As last, a one-dimension transpose convolution can be applied to transform the shape of the redundant units back to the shape of the input tensor.

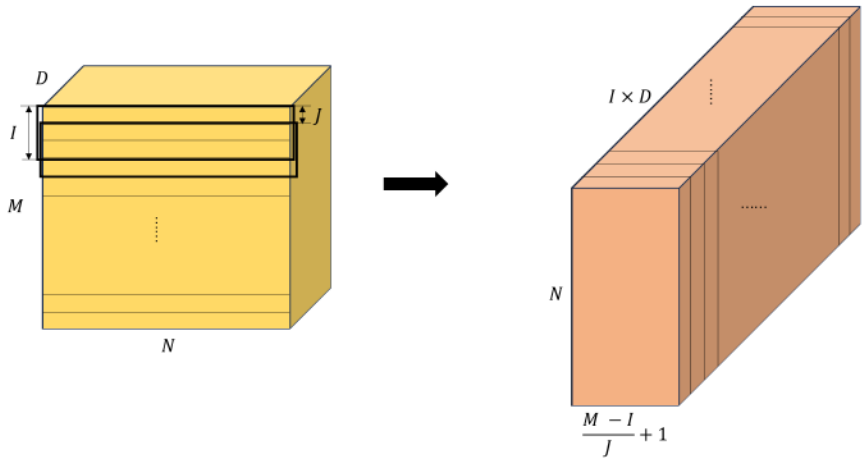


Figure 2.10: Illustration of constructing redundant units

2.2.5 Two-stage redundant Conformer blocks

Based on the two-stage conformer blocks, we combine the redundant units with the conformer blocks for our proposed two-stage redundant conformer blocks, which can capture both time and frequency information of speech features as illustrated in Fig.2.11. The proposed two-stage redundant conformer blocks share the same structure with two-stage conformer blocks as shown in Fig.2.5.

The intra-frame redundant conformer extract features within each frame while the sub-band redundant conformer extract features within each frequency sub bands. As demonstrated in Fig.2.12, each redundant conformer block consists of the redundant reconstruction, a layer normalization, a four-layer conformer and a one-dimensional transpose convolution layer. A residual connection is implemented to maintain the information of the input and avoid the gradient vanishing problem.

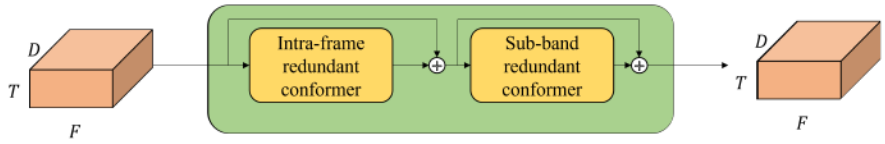


Figure 2.11: Illustration of two-stage redundant conformer blocks

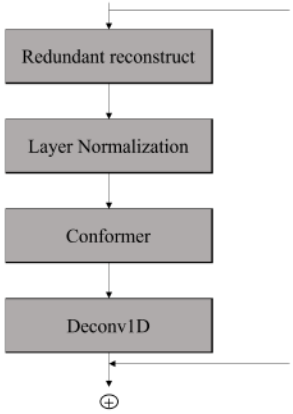


Figure 2.12: Illustration of redundant conformer block

2.2.6 Masking module

The masking module utilizes the feature output and generates masks to estimate separated speeches. As shown in Fig.2.13, the main part of the masking module is a gated convolution consists of two parallel convolution branches. More specifically, it is involved a 1-D convolution along with the sigmoid nonlinearity operation and a 1-dimensional convolution along with the Tanh operation. One convolution branch works for output generation and the other convolution branch is for the gate generation. The outputs of these branches are multiplied element wise to produce the gated

output. The two-branch gated convolution is proved to be effective to handle irregular masks. Then the outputs of the gated convolution will pass through a ReLU nonlinear function [55] for creating two masks for each speaker.

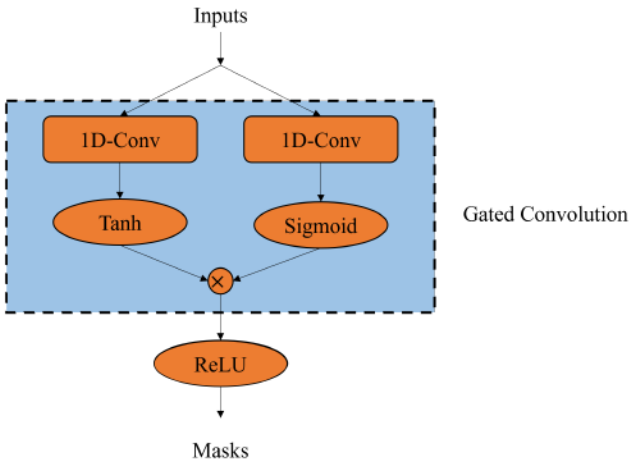


Figure 2.13: Illustration of masking module

2.2.7 Decoder

The decoder is adopted to reconstruct features from the two-stage conformer blocks to the speech features of the two target speakers. In our proposed model, we apply two different decoders, masking decoder and complex spectral mapping decoder. The masking decoder generates masks for each speaker and estimate separated speeches by element-wise multiplying the input mixture and the masks, while the complex spectral mapping decoder estimate the real and imaginary components of each speaker directly.

The masking decoder is comprised of a dilated dense decoder, a 2-dimensional convolution layer, and the masking module. As shown in Fig. 2.14, the dilated dense decoder shares a similar structure with the dilated dense encoder, consisting of a dilated dense block for up-sampling and a 2-dimensional convolution output layer. Then, the masking module is followed to generate masks of the target speakers. The dilated dense block for up-sampling employs M dilated dense layers and a sub-pixel convolution to double the dimension of the frequency bins size. It can be regarded

as the inverse procedure of the encoder. The sub-pixel convolution performs like a transposed convolution to enlarge the dimension size, which can be implemented more efficiently by rearranging the elements in the output tensor and avoids the checkerboard artifacts. Then, a 2-dimensional convolution is applied to resume the number of channels from D to 2, the same channel size with the input spectrogram, with a kernel size $(1, 2)$. Finally, the masking module is employed to generate masks of the target speakers. The generated masks will be multiplied with the input spectrogram in element-wise manner for giving final masked complex spectrogram.

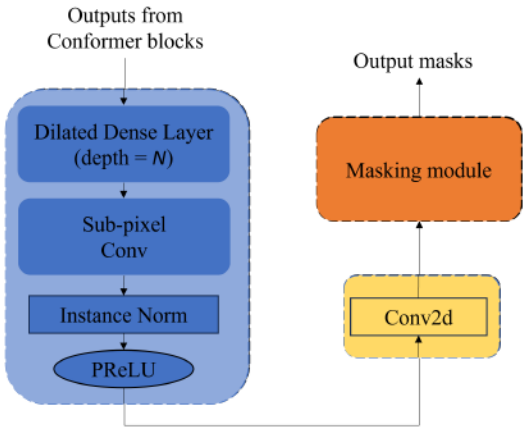


Figure 2.14: Illustration of masking decoder

The spectral mapping decoder is only consisted of a dilated dense decoder and a 2-dimensional convolution layer. As shown in Fig. 2.15, it outputs the complex spectrogram for each target speaker directly, rather than masks.

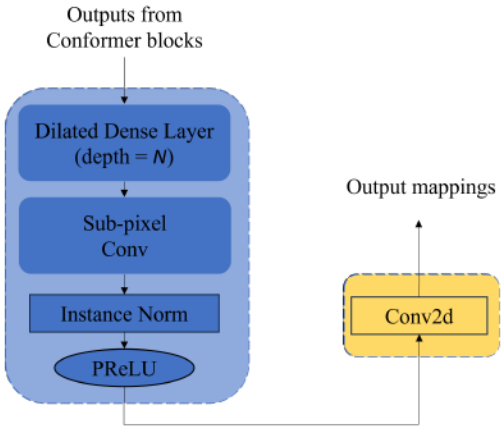


Figure 2.15: Illustration of spectral mapping decoder

2.2.8 Loss function

To train our proposed model, we adopt the loss function scale invariant speech to noise ratio (SI-SNR) [51] aimed at producing high-quality and intelligible speech signals in time domain, by measuring the ratio of the signal power to the noise power in decibels, which is formulated as [51]:

$$\begin{cases} x_{\text{target}} = \dfrac{\langle \hat{x}, x \rangle x}{\|x\|^2}, \\ e_{\text{noise}} = \hat{x} - x_{\text{target}}, \\ \text{SI-SNR} = 10 \log_{10} \dfrac{\|x_{\text{target}}\|^2}{\|e_{\text{noise}}\|^2}. \end{cases} \tag{18}$$

Where x and $\hat{x}$ denote the clean speech waveform and the estimated speech waveform. By calculating x_{target}, we define the target source signal based on the inner product and scaling it by the magnitude of the clean waveform. The SI-SNR loss is calculated by the scaled target source x_{target} and the noise e_{noise} between the estimated source signal and the target source signal. It makes the SI-SNR scale-invariant, which means that it does not depend on the amplitude of the signals.

In our two-speaker speech separation task, we calculate the SI-SNR value of the two separated speech signals separately and then computes the mean of the two SI-SNR values as the final loss function for training.

2.3 Experimental results

2.3.1 Experimental settings

We carry out our experimental studies based on a public speech data from Wall Street Journal corpus, named WSJ0-2mix [48], as introduced in 1.2.1. We adopt *SI-SNR* as evaluation metric with PIT method [5]. The mixture SI-SNR is 0 dB. We also adopt two parameters to evaluate the size and computation complexity of models, including number of parameters of the model *#Params* and multiply-accumulate operations per second *GMACs/s*. *#Params* stands for the size of a neural network and *GMACs/s* is a measure of the computational speed of a neural network.

For training speech separation models, two main types of inputs are used: speech waveforms, which lead to time-domain models, and speech spectrograms, which are used in time-frequency domain methods. Our experiments focus on spectrogram inputs, which are smaller and sparser than time-domain waveforms. We select a 4-second segment from each utterance longer than 4 seconds to reduce computational time. In each batch, shorter utterances are zero-padded to ensure uniform length. We use an STFT with a 64 ms window size and a 16 ms hop size, applying a square-root Hamming window for analysis. A 512-point discrete Fourier transform extracts 257-dimensional complex spectra for each frame. As shown in 2.1, the feature channel of spectrogram inputs is 2 and the encoding dimension of the dilated dense net encoder is D. The encoder and decoder settings have large impacts on performance. However, we do not discuss the performance of dense net here. To get a balance between the model size and the final performance, we utilize 4 down-sampling layers in encoder and 4 up-sampling layers in decoder, where each layer includes a dilated dense block using dilated convolution with dilation factor of 3. In redundant units, the kernel size and stride size are I and J respectively. We used 4 two-stage redundant Conformer blocks in proposed RCFormer. For proposed Conformer setting, the input feature dimension is set as D, the same with the encoding dimension, and the attention heads H in multi-head attention are set as 4. The kernel size of the convolution layer in Conformer is set as 31.

We train and optimize our proposed model for 200 epochs by using Adam optimizer [67]. To alleviate the gradient explosion problem in sequences and get more accurate training performance, we adopt a learning rate decay method during training. More specifically, In the first 50 epochs, the

Table 2.1: Summary of model hyper-parameters

Symbols	Description
D	Embedding dimension
N	Number of dilated dense layers
N	Number of two-stage Conformer blocks
I	Kernel size for Unfold and Deconv1D
J	Stride size for Unfold and Deconv1D
L	Kernel size in convolution layer in Conformer
H	Number of attention heads in multi-head attention of Conformer

learning rate is initially $8e^{-4}$. After that, the learning rate is decayed to half of the former learning rate for every 30 epochs. It can be defined below:

$$Lr = \begin{cases} k, & n \leq n\_initial \\ k \cdot 0.5^{\lfloor (n - n\_initial)/n\_decay \rfloor + 1}, & n > n\_initial \end{cases} \tag{19}$$

where n denotes the epochs, and $k = 8e^{-4}$ are hyper-parameters. $n\_initial$ denotes the number of epochs that using the initial learning rate, which is set as 50. n_decay denotes the number epochs that every time the learning rate delays half, which is set as 30.

2.3.2 Comparisons with baselines

Table 2.2 The comparison results for the proposed model and various existing time-domain and time-frequency domain methods, evaluated on the WSJ0-2mix dataset, are presented. No data augmentation methods were used during training. Despite its efficiency, our proposed RCFormer demonstrates impressive performance compared to existing models. While time-domain models generally perform better than time-frequency domain models, RCFormer achieves results comparable to the state-of-the-art time-domain model, Sepformer, but with significantly lower model complexity (4.8M parameters). RCFormer also outperforms popular time-domain models such as DPRNN, Conv-TasNet, and DPT-Net. Although the latest MossFormer and TF-GridNet outperform RCFormer, MossFormer has a much larger parameter size (42.1 million) compared to RCFormer (4.6 million), and TF-GridNet requires substantially more *GMACs/s* (231.1) than RCFormer (52.2)

Second, our proposed RCFormer achieves a state-of-the-art *SI-SNR* value compared to other

popular efficient models, such as Sepformer-light, SkiM, RE-Sepformer, and various efficient Transformer architectures. RCFormer excels in modeling long-range speech sequences while maintaining a relatively low number of parameters. This demonstrates that our redundant reconstruction and Conformer approach effectively captures both local and global information in spectrograms while reducing parameter count. Additionally, when comparing RCFormer to other efficient Transformer-based two-stage models like Reformer and Longformer, RCFormer delivers significantly better performance. This suggests that the combination of Conformer and redundant reconstruction is currently the most effective two-stage model, enhancing the ability to extract both intra-frame time dependencies and sub-band frequency dependencies.

Table 2.2: Comparison with other models on WSJ0-2mix

Models	Domain	Year	SI-SNR (dB)	#Params (M)	GMACs/s
Conv-TasNet [20]	Time	2019	15.3	5.1	3.2
FurcaNeXt [16]	Time	2020	18.2	51.4	-
DPRNN [23]	Time	2020	18.8	2.6	38.8
Gated DPRNN [25]	Time	2020	20.1	7.5	49.6
DPTNet [40]	Time	2020	20.2	2.6	-
SepFormer [14]	Time	2021	21.4	26.0	70
SepFormer Light [44]	Time	2022	19.8	6.4	17.5
Wavesplit [68]	Time	2021	21.1	29	-
TFPSNet [37]	T-F	2022	21.1	2.7	29.6
SFSRNet [69]	Time	2022	22	59.1	-
QDPN [70]	Time	2022	22.1	200	-
SkiM [43]	Time	2022	18.2	14.5	3.7
RE-SepFormer [44]	Time	2023	18.6	8.0	6.3
SepIt + DM [71]	Time	2023	22.4	4.6	-
Reformer	T-F	-	18.9	9.2	28.2
Longformer	T-F	-	14.2	15.1	12.3
MossFormer [72]	Time	2024	22.9	42.1	-
TF-GridNet [45]	T-F	2023	23.4	14.4	231.1
RCFormer	T-F	-	**21.8**	**4.8**	39

2.3.3 Ablation study

In this section, we conduct comparison experiments to evaluate the effectiveness of different components of our proposed model. For simplicity, these experiments are performed on the WSJ0-2mix dataset using the same loss function.

First, we compare our two-stage model with both time-domain and time-frequency-domain inputs. We conducted two sets of experiments to explore performance and efficiency. In the first set, we tune hyperparameters for both time-domain and time-frequency-domain inputs to achieve optimal results without considering model size. In the second set, we maintain a constant model size across various time-domain and time-frequency-domain configurations to compare their performance. For the time-domain model, we use a 1-dimensional convolution layer as an encoder to transform long speech sequences into latent representations.

As shown in Table 2.3, the model using time-domain inputs obtains a 20.2 dB performance, while the time-frequency-domain model obtains 18.9 dB performance. However, time-domain model requires much more computational resources than our time-frequency-domain model. More specifically, the time-domain model requires more than 7 times number of parameters comparing with the time-frequency-domain model and a *GMACs/s* increasing of 65%.

Table 2.3: Best results of time-domain and time-frequency-domain two-stage models

	SI-SNR (dB)	#Params (M)	GMACs/s
Time	20.2	14	60
T-F	18.6	2.3	23

From the Table 2.4, we also observe that the time-frequency-domain model outperforms the time-domain when the number of parameters is held constant range from 1.5M to 8M. However, if we increase the model size to more than 15M parameters, the time-domain model can achieve more than 20 dB performance while the performance of time-frequency-domain model only slightly improves. The possible reason is that the time-domain representation contains denser information while the information in the spectrogram is sparser. As a result, the time-frequency domain model requires significantly fewer computational resources than the time domain model.

Secondly, the two-stage Conformer block is designed to capture both temporal and frequency features through intra-frame and sub-band Conformers. This is followed by a two-stage model aimed at extracting local and global information from long sequences. The choice of core networks significantly impacts performance. To evaluate the effectiveness of our proposed Conformer core network, we conducted experiments comparing it against several popular core networks that have

Table 2.4: Evaluation results of time-domain and time-frequency-domain two-stage models

	SI-SNR (dB)	#Params (M)
Time	-	1.5M
	16.6	3M
	18.4	5M
	18.9	8M
	21.9	15M
	22.1	25M
T-F	18.4	1.5M
	18.9	3M
	19.8	5M
	20.1	8M
	20.9	15M
	-	25M

proven effective in prior research. For a fair comparison, we integrated each core network with our proposed two-stage redundant block, maintaining the same dilated dense encoder and decoder architecture. The inputs are complex spectrograms with no augmentation, with sizes consistent with those described in our experimental settings.

To further explore the superiority and efficiency of our proposed Conformer core, we designed two groups of comparative experiments. In the first group, we fixed the *SI-SNR* at 18.5 dB, a standard value empirically demonstrated as attainable by our reference models, to focus on comparing the model complexity of different core networks. In the second group, we held the number of model parameters constant at 3 million and compared the resulting *SI-SNR* performance across the networks.

Table 2.5: Evaluation results of different core networks with two-stage backbone (constant *SI-SNR* value)

	SI-SNR (dB)	#Params (M)	GMACs/s
RNN	18.4	8	48
LSTM	18.4	6.6	60
Vanilla Transformer	18.4	9.2	76
Reformer	18.4	5	30
Conformer (ours)	18.5	**1.3**	**19**

As shown in Table 2.5, our proposed Conformer-core network is distinguished by both its considerable size and the lowest *GMACs/s*. Especially, for the Transformer-based core networks

Table 2.6: Evaluation results of different core networks with two-stage backbone (constant model size)

	SI-SNR (dB)	#Params (M)	GMACs/s
RNN	17.9	4.9	26
LSTM	18.2	5	33
Vanilla Transformer	17.6	5	55
Reformer	15.4	4.9	30
Conformer (ours)	**21.8**	5	39

(Vanilla Transformer, Reformer, and Conformer), The vanilla Transformer core requires 7 times number of parameters and 3.5 times of *GMACs/s* to achieve the 18.5 dB. The Reformer core is superior to the vanilla Transformer which requires smaller model size and lower computational complexity. However, Conformer-core network achieves the overall best performance with the smallest model size and relatively low computational complexity. Table 2.6 also emphasize the Conformer core's superior performance in both performance and computational efficiency. It achieves a 21.8 dB SI-SNR value, which is about 3 dB better than reference models. We can find that attention-free core networks like RNN and LSTM perform better than the self-attention-based Transformer networks. However, Transformer networks achieve much better results with time-domain inputs. The possible reason is that vanilla Transformer architecture is more suitable for processing large long-sequences because its self-attention mechanism indeed excel in capturing global context and handling long-range dependencies, while it has limitation in capturing temporal dependencies like sparse spectrogram inputs. It proves that Conformer, the combination of convolution layer and Transformer, has great advantage in processing smaller and sparser input like spectrogram. Although the Conformer core requires more *GMACs/s* than RNN, LSTM, and Reformer, we demonstrate that it shares similar training and processing times compared to other reference models. Specifically, we measured the inference time in seconds using an NVIDIA A100 GPU across different input lengths. As shown in Fig.2.16, the inference time of the Conformer core is only marginally slower (by 0.2 seconds) than other reference models when the input length is 32 seconds. Given that our typical input signal length is between 4 to 6 seconds, this indicates that our proposed Conformer core network can operate efficiently and quickly in real-world applications.

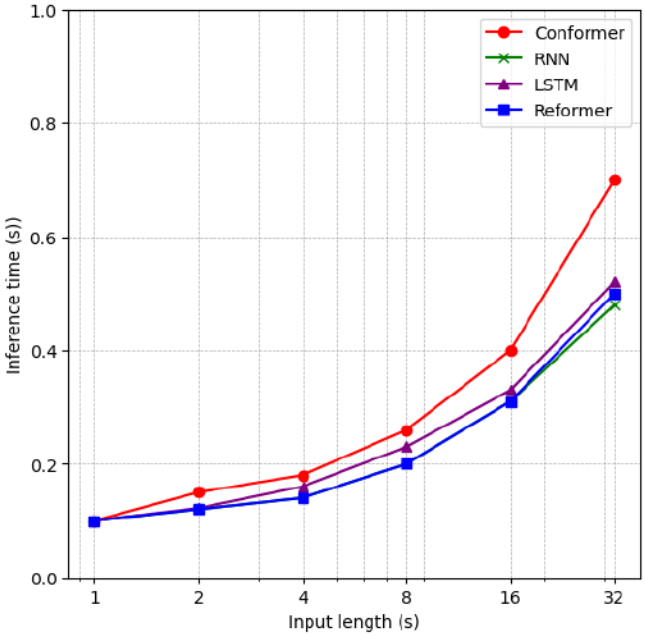


Figure 2.16: Inference time in seconds comparison of Conformer, RNN, LSTM, and Reformer cores

Finally, we further analyze the redundant approach which stacks the nearby embedding to reshape the encoded inputs. In order to explore the effectiveness and generalizability of our proposed redundant approach, we conduct experiments using the two-stage architecture, simply adding or removing our proposed redundant units. Apart from testing the performance of our proposed Conformer-based model, we also integrate other reference core networks including RNN, LSTM and Vanilla Transformer. To ensure a fair comparison and accurate evaluation of the improvement from using a redundant approach, we compare different core networks with and without redundant units. We ensure that the model parameter numbers are kept at the same level for these comparisons. This allows us to evaluate the performance and efficiency improvement by comparing the *SI-SNR* and *GMACs/s* values. Among all the previously mentioned core networks, we allocated 2M, 4M, 12M, and 5M parameters for RNN, LSTM, Vanilla Transformer, and Conformer core networks respectively. These values were chosen based on our experimental results and previous research, as they were reported to deliver the best *GMACs/s* performance.

As demonstrated in Table 2.7, we can see that utilizing a redundant approach can enhance the

Table 2.7: Evaluation results of our proposed redundant approach, (R) means applying redundant approach

	SI-SNR (dB)	#Params (M)	GMACs/s
RNN	17.1	2	22
RNN (R)	18	2	38
LSTM	17.6	4	33
LSTM (R)	18.7	4	46
Transformer	18.5	12	66
Transformer (R)	21.5	12.2	82
Conformer	20.1	5	20
Conformer (R)	21.8	4.8	39

SI-SNR performance by at least 1 dB while maintaining a similar *GMACs/s* value. Notably, our proposed Conformer-based model achieves the optimal performance of 21.8 dB, demonstrating that our RCFormer configuration is the most suitable for our task. Furthermore, we observe that the Vanilla Transformer-based model shows a 2.8 dB improvement when redundant units are added, reinforcing the suitability of a Transformer-based two-stage model for processing long-range sequences with substantial information.

2.4 Summary

In this chapter, we introduce the RCFormer, a Conformer-based neural network specifically designed for mono-channel two-speaker speech separation with a focus on resource efficiency. The RCFormer achieves its objective by leveraging a compact model architecture and using a sparse spectrogram as input. The model is built on a mask network framework comprising an encoder, two-stage Conformer blocks, a masking module, and a decoder. The two-stage Conformer block, the core component of the model, captures both frame-level and sub-band-level information from the input spectrogram. The intra-frame Conformer block models local spectral information within each frame, while the sub-band Conformer block captures temporal information across sub-bands. To effectively manage the sparse information in the spectrogram, we incorporate redundancy to transform the input spectrogram embedding into a denser representation. Finally, the output from the two-stage model passes through a masking decoder to generate two masks, which are applied to the input spectrogram to facilitate the reconstruction of the separated speech signals.

Experimental results demonstrate that the proposed RCFormer achieves outstanding performance in speech separation compared to existing methodologies. Moreover, the RCFormer strikes a balance between performance and efficiency, highlighting the effectiveness of our approach in integrating redundant units and employing a two-stage Conformer structure. This design minimizes the number of trainable parameters and reduces memory usage and inference time, making the RCFormer both powerful and resource-efficient.

Chapter 3

Frequency band level neural network for speech separation

In this chapter, we propose a novel and frequency band-level architecture based on our proposed sub-band module and full-band module, named Full-band and sub-band neural network (FSBNet). This chapter is organized as follows. We first introduce the sub-band module and full-band module in Section 3.1 and Section 3.2 respectively. In Section 3.3, we describe the proposed FSBNet and its application in speech separation. In Section 3.4, we evaluate the performance and efficiency of our proposed FSBNet.

3.1 Sub-band module

3.1.1 Compact latent summaries

The two-stage (dual-path) architecture has proven effective in tasks such as speech separation and speech enhancement. However, these models tend to be computationally intensive and require a large number of learnable parameters. This is likely due to the architecture's approach of splitting the input sequence into small chunks, which necessitates separate processing to capture both local and global dependencies effectively.

Recently, some researchers have proposed ideas called compact latent summaries [44], which compute a lower-dimensional representation capturing essential information. To illustrate, as shown

in Fig.3.1, assume we have a 3-dimensional latent representation $x \in \mathbf{R}^{D \times M \times N}$, where D represents the channel size and M and N represent the size of chunk and number of chunks, respectively. Then, an average pooling is applied on the chunk dimension (M dimension) to produce a summary representation $x' \in \mathbf{R}^{D \times N}$. The rationale behind this operation is that averaging over the chunk dimension of a latent representation can provide enough high-level contextual information and can save a significant amount of computations.

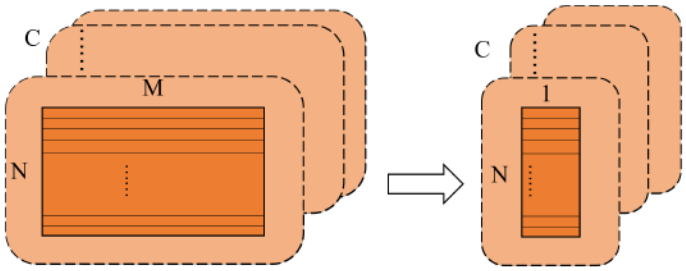


Figure 3.1: Illustration of compact latent summaries

3.1.2 Proposed sub-band module

Figure 3.2 illustrates the structure of our proposed sub-band module, which comprises three main components: $SubbandNet1$, $CrossbandNet$, and $SubbandNet2$. All of the mentioned Net can be set as any type of neural network, such as RNN, LSTM [22], or Transformer [29]. It is essential to ensure that the input and output tensor shapes of the neural network remain the same.

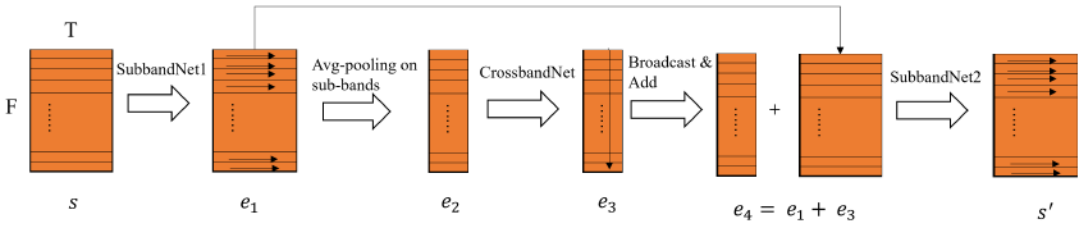


Figure 3.2: Illustration of proposed sub-band module

The sub-band module takes a 3-dimensional tensor as input. In our scenario, we use an encoded spectrogram representation $s \in \mathbf{R}^{D \times T \times F}$, where T represents the number of time frames, and F represents the number of frequency bins. Initially, we treat the input tensor e as F separate

sequences, each with a length of T, dividing e into F sub-bands. Next, we apply $SubbandNet1$ to the time frame axis of all the sub-bands, resulting in e_1. The purpose of this step is to model temporal information within each sub-band.

$$e_1 = [SubbandNet1(s[:,:,f]), for f = 1, ..., F] \in \mathbf{R}^{D \times T \times F} \tag{20}$$

Then, we compute a summary representation following the compact latent summaries method by average pooling e_1 over the time axis on every sub-bands. This operation allows us to obtain high-level contextual information to be used for the following cross-band dependency embedding:

$$e_2 = [AveragePooling(e_1[:,1,f]), for f = 1, ..., F] \in \mathbf{R}^{D \times F} \tag{21}$$

Afterwards, the latent summary e_2 is processed through the $CrossbandNet$, which operates along the frequency axis and models cross sub-bands dependencies. The resulting e_3 will first broadcast over the time axis and then add element-wise to e_1, generating a new tensor e_4:

$$e_3 = [CrossbandNet(e_2[:,:])] \in \mathbf{R}^{D \times F} \tag{22}$$

$$e_4 = [e_1 + Braodcasting(e_3)] \in \mathbf{R}^{D \times T \times F} \tag{23}$$

The resulting e_4 incorporates both sub-bands and cross-bands dependencies to extract sufficient features from the input spectrogram. The above steps can effectively reduce the number of trainable parameters and save computational resources. This approach differs from the two-stage model as it avoids operating on full tensor e_1 in two stages.

Finally, we also design another sub-band neural networks ($SubbandNet2$), which further enhances the separation quality by integrating the context provided by both sub-band and cross-band networks. The $SubbandNet2$ step ensures thorough consideration of both local and global features. The output s' maintains the same shape as the input s, which is convenient for further processing:

$$s' = SubbandNet2(e_4) \in \mathbf{R}^{D \times T \times F} \tag{24}$$

3.2 Full-band module

The full-band module is inspired by the self-attention convolutional neural networks used in music separation [68] and speech enhancement [28]. Similar to these networks, we employ a whole-sequence self-attention module to capture long-range global information. However, unlike previous approaches that operate in the time domain, we integrate this module with the encoded spectrogram input, which contains both time and frequency information. Figure 3.3 shows the structure of the full-band self-attention module, which consists of 2-dimensional convolution neural networks, L heads self-attention module, and another 2-dimensional convolution neural network.

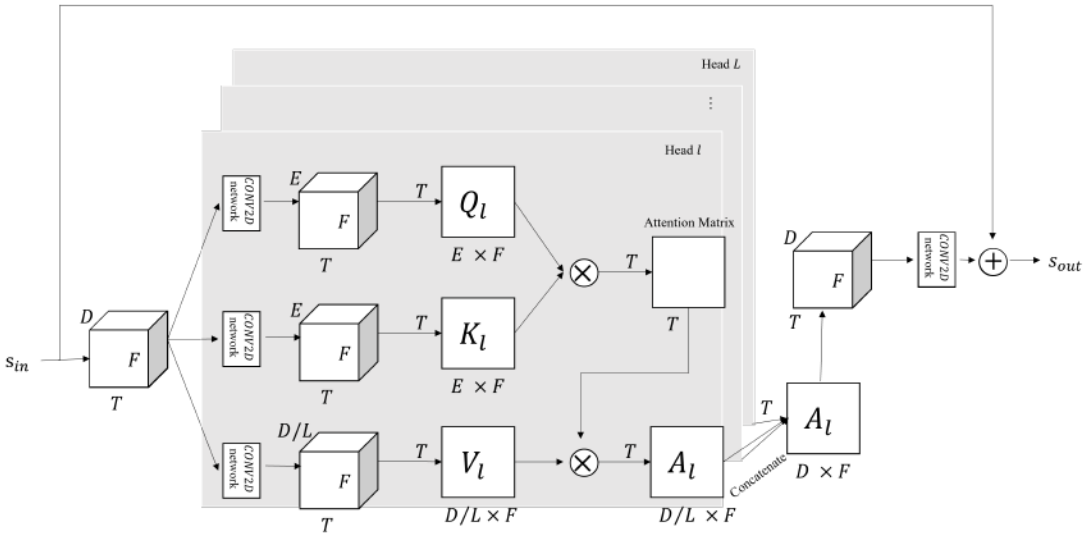


Figure 3.3: Illustration of proposed full-band module

Our input is the same with it in 3.1, a 3-dimensional encoded spectrogram $s_{in} \in \mathbf{R}^{D \times T \times F}$. We view the input spectrogram as a frame-level representation, which is T separate sequences and each with length F. The 2-dimensional convolution neural networks in this module consists of a 2-dimensional convolution with kernel size equal to 1, a PReLU function [56], and layer normalization [73]. We first compute frame-level embeddings within each frame and then use the whole-sequence self-attention on these frame-level embeddings. In detail, in each head l in the self-attention module, we first applies 2-dimensional convolution neural network, which transforms the channel size from D to E and normalize along the channel and frequency dimensions. Then

we reshape the normalized tensor and obtain the 2-dimensional query $Q_l \in \mathbf{R}^{T\times(F\times E)}$ and key $K_l \in \mathbf{R}^{T\times(F\times E)}$, leading to $F \times E$-dimensional query and key vectors at each frame. Similarly, we obtain the $F \times D/L$-dimensional value vector $V_l \in \mathbf{R}^{T\times(F\times D/L)}$. The attention output $A_l \in \mathbf{R}^{T\times(F\times D/L)}$ is computed as:

$$A_l = softmax(\frac{Q_l K_l^T}{\sqrt{F \times E}})V_l \tag{25}$$

Afterwards, we concatenate the attention outputs of all L heads along the T dimension, reshape it back to $D \times T \times F$. We also apply another 2-dimensional convolution neural network to learn the information across all heads. Finally, we use a residual connection to add the attention output to the input tensor, obtaining the final output.

3.3 Proposed full-band and sub-band neural network (FSBNet)

3.3.1 Discussion on full-band and sub-band modeling

In the realm of speech separation, effective modeling techniques are essential for achieving high-quality results, particularly when processing spectrograms that encapsulate both time and frequency information. Sub-band and full-band modeling are two prevalent approaches that offer distinct advantages.

Sub-band modeling involves decomposing the spectrogram along the frequency axis into several narrower sub-bands. Each sub-band is then analyzed and processed independently, allowing for targeted augmentations or modifications tailored to the specific characteristics of each frequency range. Many traditional algorithms, such as beamforming [74], dereverberation, and spatial clustering [75], are applied separately within each frequency band. This separation is based on the physical phenomenon that certain spatial patterns, like phase differences and reverberation, vary with frequency but remain stable within each time frame. Consequently, it is reasonable to employ DNNs for sub-band modeling, as the consistent pattern information within each time frame lends itself well to supervised learning using DNNs.

Full-band modeling, on the other hand, considers the complete range of frequencies simultaneously, utilizing all available spectral information to capture the overall characteristics of the speech signal. Full-band models can preserve the integrity of the original spectral features. In the case of FullsubNet [34], a DNN-based full-band model was developed to capture global spectral context and long-distance cross-band dependencies. The advantage of using a full-band DNN lies in its ability to extract spectral patterns, such as harmonic structures and gradual changes along the frequency axis.

Several early studies have employed DNNs to perform both full-band and sub-band modeling in speech separation and enhancement tasks. These studies can be categorized as follows:

(1) Only perform sub-band modeling, without full-band modeling, which is typically applied in early frequency domain speech enhancement tasks.

(2) Perform both sub-band and full-band modeling, but without iterative information flow from sub-band to full-band modules and from full-band to sub-band.

(3) Perform both sub-band and full-band modeling, but the so-called "full-band modeling" only captures the local spectral information within each time frame.

The strategies for developing our full-band and sub-band neural network are as follows: First, we integrate both sub-band and full-band modeling to comprehensively capture the necessary information from the input spectrogram. Second, we stack multiple full-band and sub-band module blocks to facilitate effective information flow between the two modules. Finally, as detailed in Section 3.2, our proposed full-band module enables each time frame to attend to any other frame of interest, thereby exploiting long-range dependencies within the spectrogram.

3.3.2 Proposed full-band and sub-band neural network (FSBNet) architecture

Figure 3.4 illustrates the entire structure of our proposed full-band and sub-band neural network (FSBNet), consists of an encoder, multiple full-band and sub-band (FSB) blocks, and a decoder. Note that, different from the masking module we proposed in 2.2.6, FSBNet is trained to perform the complex spectral mapping directly, where the real and imaginary (RI) components of the input mixture are concatenated as input features to predict the RI components of each speaker.

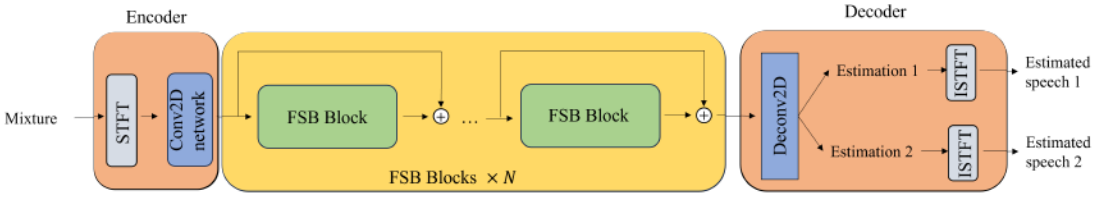


Figure 3.4: Overview of full-band and sub-band neural network (FSBNet) architecture

The encoder consists of a STFT feature extractor and a 2-D convolution neural network to obtain the representation of the input speech mixture. The 2-D convolution neural network includes a 2-D convolution with a kernel size of 3, a global layer normalization, and a PReLU function. Firstly, the complex spectrogram with real and imaginary (RI) components $M \in \mathbf{R}^{2\times T\times F}$ is extracted by using STFT, where T denotes the number of time frames and F denotes the number of frequency bins. Subsequently, the 2-dimensional convolution neural network is employed to compute a D-dimensional embedding, resulting in a new tensor $R_n \in \mathbf{R}^{D\times T\times F}$, which is then inputted into the subsequent FSB blocks.

The FSB blocks comprise a total of N blocks, each containing a sub-band module and a full-band module. As shown in Figure 3.5, The D-dimensional embedding R_n is treated as F sub-bands and initially processed by the sub-band module as proposed in 3.1.2. We choose Conformer as the component of $SubbandNet1$, $CrossbandNet$, and $SubbandNet2$, which has been proved to be most effective in Table 2.5 and Table 2.6. This operation extracts the local information within each sub-bands and high-level contextual information across different sub-bands. The output representation from sub-band module, Z_n, is then treated as T time frames and fed to the full-band module proposed in 3.2. This whole-sequence self-attention module captures long-range global information of the entire input representation. Finally, the output R_{n+1} is fed to the next FSB block. Moreover, the sum of the output of each FSB block and the residual connection from its input are passed to the subsequent FSB block, mitigating the gradient vanishing problem during training.

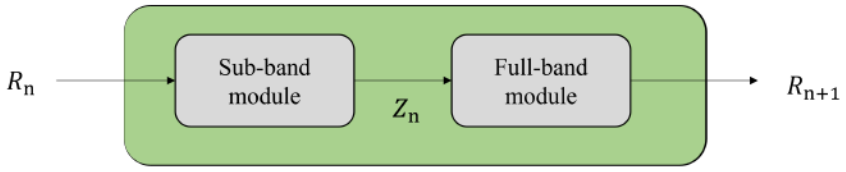


Figure 3.5: Illustration of full-band and sub-band (FSB) block

In the decoder stage, the output from FSB blocks is processed by a 2-dimensional deconvolution (Deconv2D) with a 3×3 kernel and $2S$ output channels, where S denotes the number of speakers in our task. The outputs of Deconv2D are the final predicted RI components, and inverse STFT (iSTFT) is applied for speech re-synthesis. As we mentioned earlier, the re-synthesis speeches of predicted RI components are our final outputs and these outputs will be used to calculate the loss function for training purposes.

3.3.3 Loss function

Our proposed models FSBNet also trained with PIT [5], which is the same with RCFormer in Chapter 2. The loss function follows the *SI-SDR* loss [51], with some minor improvements incorporated. First, in *SI-SDR*, we scale the estimation to equalize its energy level with that of the target. The formula is shown as follows [51]:

$$\mathcal{L}_{\text{SI-SDR-SE}} = -\sum_{c=1}^{C} 10 \log_{10} \frac{\|s^{(c)}\|_2^2}{\|\hat{\alpha}^{(c)}\hat{s}^{(c)} - s^{(c)}\|_2^2}, \tag{26}$$

where "SE" means "scaling estimation", $\hat{s}^{(c)}$ denotes the reconstructed signal based on the predicted RI components for speaker c, and $\hat{\alpha}^{(c)}$ indicates the scaled factor for estimation normalizing, which is:

$$\hat{\alpha}^{(c)} = \arg\min_{\alpha} \left\| \alpha\hat{s}^{(c)} - s^{(c)} \right\|_2^2 = \frac{(\hat{s}^{(c)})^\top s^{(c)}}{(\hat{s}^{(c)})^\top \hat{s}^{(c)}}. \tag{27}$$

Second, inspired from the MAE loss which is applied in PIT network, we add a mixture constraint (MC) loss between the mixture and the summation of the scaled estimated sources, defined as:

$$\mathcal{L}_{\text{SI-SDR-SE+MC}} = \mathcal{L}_{\text{SI-SDR-SE}} + \frac{1}{N}\left\|\sum_{c=1}^{C}\hat{\alpha}^{(c)}\hat{s}^{(c)} - y\right\|_1, \tag{28}$$

where N indicates the time samples, y indicates the waveform of input mixture and the sample variance of y has been normalized to one before hand.It is motivated by a trigonometric perspective in source separation, which suggested that constraining the separated sources to sum up to the mixture can lead to better phase estimation.

3.4 Experimental results

3.4.1 Experimental setups

To compare the proposed FSBNet with other existing models, we conduct our experiments on WSJ0-2mix dataset [48], which is the same with Chapter 2.

All utterances are trimmed to 4 seconds and sampled at 8 kHz. We use a square-root Hanning window for analysis, with an STFT window size of 32 ms and a hop size of 8 ms. A 256-point discrete Fourier transform is applied to extract 129-dimensional complex spectra at each frame. For embedding, we set $D = 64$, use $N = 8$ FSB blocks, and configure the number of channels for the query and key tensors in the full-band self-attention module E to 4. Additional hyperparameters, which may vary across experiments, are detailed in Table 3.1. During training, we use Adam as the optimizer. The FBSNet is trained for 200 epochs, starting with a learning rate of 0.001, which is halved after epoch 80 if validation performance does not improve for 3 consecutive epochs.

Table 3.1: Summary of model hyper-parameters in FSBNet

Symbols	Description
D	Embedding dimension for FSB block
N	Number of FSB blocks
B	Number of layers in Conformer-based $SubbandNet$ and $CrossbandNet$
H	Number of attention heads in Conformer
E	Number of channels to obtain the query and key tensors in the full-band module
L	Number of heads in self-attention in the full-band module

We evaluate our model in both performance and efficiency. We adopt *SI-SNR* as evaluation metric. The mixture *SI-SNR* is 0 dB. We also adopt two parameters to evaluate the size and computation complexity of models, including number of parameters of the model *#Params* and multiply-accumulate operations per second *GMACs/s*. We also evaluate the model efficiency in real-world application by comparing the inference time in seconds.

3.4.2 Comparisons with baselines

Table 3.2 summaries the comparison results with some state-of-the-art speech separation models on the same dataset. First, FSBNet achieves a superior performance compared to most of existing models, with a 20.6 dB *SI-SNR*. Especially, despite the conventional belief that time-domain models have better performance than T-F-domain models in terms of input feature representation, FSBNet shows remarkable performance, outperforming popular time-domain models like Conv-Tasnet, DPTNet, and DPRNN. While some models achieve more than 21 dB SI-SNR performance, most of them apply complex model structures and do not consider complexity during the training stage. In contrast, our proposed FSBNet achieve the best overall computational efficiency, with only 2.4M parameters and 9.8 GMACs/s, which is at least 10 times fewer parameters than the existing time-domain models with *SI-SNR* value over 21 dB. FSBNet obtains a inferior *SI-SNR* performance than the state-of-the-art T-F domian model, TFPSNet and TF-GridNet, but it only requires 9.8 GMACs/s, which is approximately 3 times fewer than TFPSNet and 23 times fewer than TF-GridNet.

Second, we further compare the proposed FSBNet with other models that prioritize resource efficiency. From Table 3.2, we observe that our full-band and sub-band based architecture model achieves the best performance regarding to *SI-SNR* value. Additionally, FSBNet has fewer trainable parameters compared to most resource efficiency models. While it requires slightly more parameters than S4M-tiny and DTCN + SW + DM, it delivers significantly better *SI-SNR* performance. In terms of *GMACs/s*, FSBNet also demonstrates impressive performance. It only requires 4 times fewer GMACs/s comparing to S4M, which obtains almost the same *SI-SNR* value. Furthermore, FSBNet outperforms SkiM and RE-SepFormer by 2 dB performance, with only a slight increase in *GMACs/s*.

Table 3.2: Experimental results of the proposed and existing models on WSJ0-2mix

Models	Domain	Year	SI-SNR (dB)	#Params (M)	GMACs/s
Conv-TasNet [20]	Time	2019	15.3	5.1	3.2
FurcaNeXt [16]	Time	2020	18.2	51.4	-
DPRNN [23]	Time	2020	18.8	2.6	38.8
DPTNet [40]	Time	2020	20.2	2.6	-
SepFormer [14]	Time	2021	21.4	26.0	70
Wavesplit [68]	Time	2021	21.1	29	-
SFSRNet [69]	Time	2022	22	59.1	-
SepIt + DM [71]	Time	2023	22.4	4.6	-
MossFormer [72]	Time	2024	22.9	42.1	-
Reformer	T-F	-	18.9	5.2	28.2
Longformer	T-F	-	16.2	7.1	12.3
TFPSNet [37]	T-F	2022	21.1	2.7	29.6
TF-GridNet [45]	T-F	2023	23.4	14.4	231.1
DTCN + SW + DM [76]	Time	2022	16.2	1.3	3.7
SkiM [43]	Time	2022	18.2	14.5	3.7
SepFormer Light [44]	Time	2022	19.8	6.4	17.5
RE-SepFormer [44]	Time	2023	18.6	8.0	6.3
S4M-tiny [38]	Time	2024	19.3	1.8	8.0
S4M [38]	Time	2024	20.5	3.6	38.7
FSBNet	T-F	-	**20.6**	**2.4**	**9.8**

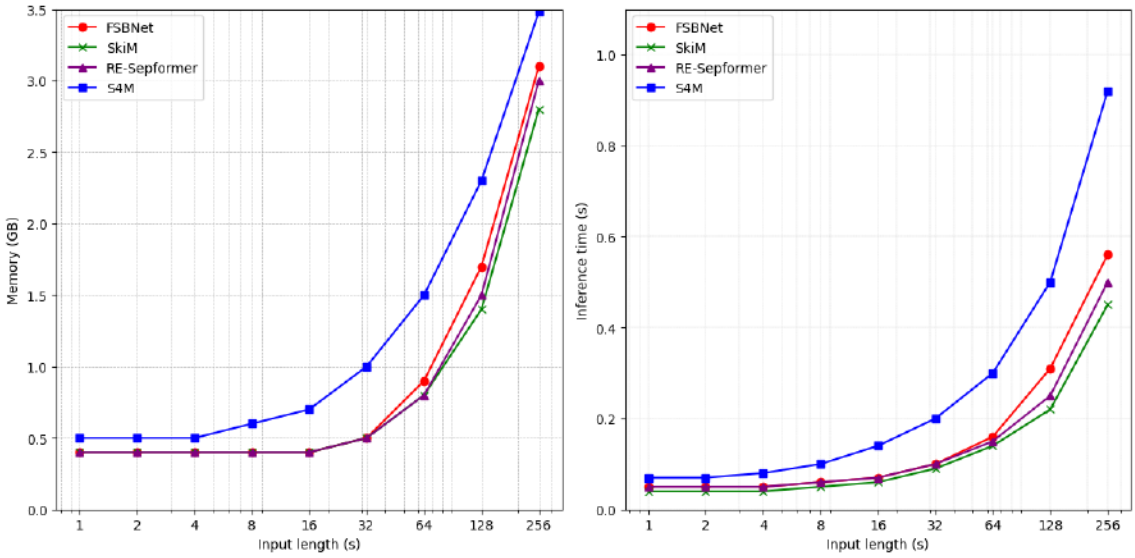


Figure 3.6: Memory in GB (left panel) and Inference time in seconds (right panel) comparison of FSBNet, SkiM, RE-SepFormer, and S4M

Finally, we compare the memory usage (Figure 3.6 (left panel)) and inference time (Figure

3.6 (right panel)) of FSBNet and some representative resource efficiency models,including SkiM, RE-SepFormer, and S4M. This experiment was conducted on an NVIDIA A100 GPU, considering various input lengths. We observed that the memory cost and inference time of FSBNet are slightly higher than those of the state-of-the-art RE-Sepformer model when the input signal length exceeds 64 seconds. However, such lengthy inputs represent an extreme situation, even in real-world applications.

3.4.3 Ablation study

In this section, we explore the performance of our proposed FSBNet with different network configurations. First, we applies complex spectral mapping for training, rather than the masking module applied in 2.2.6. We believe that predicting the RI component directly can be more effective when the input representation is complex spectrogram which contains sparse and small values. We set up experiments to evaluate the performance of complex spectral mapping when applied in our proposed FSBNet and some other popular time domain and T-F domain models. Specifically, we compare the $SI-SNR$ performance of time domain DPRNN, T-F domain two-stage Conformer, and our proposed T-F domain FSBNet, applying complex spectral mapping and masking module respectively. The two-stage Conformer can be viewed as a simplified RCFormer without applying redundant approach. For a fair comparison, all the models are trained with the loss 28 in WSJ0-2mix dataset. For DPRNN, the parameter setting follows the best configuration. For two-stage Conformer and FSBNet, the input spectrogram shares the same size, where window size is set to 32 ms with an 8 ms hop size.

Table 3.3: Comparison between complex spectral mapping and masking module

Row	Models	Masking or Mapping?	#params (M)	SI-SNR (dB)
1	DPRNN	Masking module	2.6	18.8
2	DPRNN	Complex spectral mapping	2.4	18.4
3	Two-stage Conformer	Masking module	2.5	19.5
4	Two-stage Conformer	Complex spectral mapping	2.4	19.8
5	FSBNet	Masking module	2.6	20.1
6	FSBNet	Complex spectral mapping	2.4	**20.6**

As indicated in Table 3.3, each model has nearly the same number of parameters. The slightly

differences are caused by the parameter increase when applying masking module. We observe that T-F domain two-stage Conformer and FSBNet model both perform better when using complex spectral mapping. However, we do not observe any improvement in time domain DPRNN model. It indicates that complex spectral mapping is effective for our proposed FSBNet as well as another T-F domain model, but its effect is not satisfactory when applied in the time domain model.

Additionally, our proposed FSBNet contains sub-band module and full-band module. The sub-band module is applied to extract temporal information within each band and overall spectral information with a summary vector. The full-band module is applied to exploit the long-range information from the entire frame. We further analyze the relative importance of each module to the overall performance.

In the sub-band module part, we apply $SubbandNet$, which consists of $SubbandNet1$ and $SubbandNet2$, to process on each decomposed sub-band, and $CrossbandNet$ to process on fame-level. We conduct experiments to explore the importance of $SubbandNet$ and $CrossbandNet$ along with their effectiveness for the entire sub-band module. To ensure more accurate experimental results, we remove the full-band module and solely conduct experiments on sub-band module with the same training loss as FSBNet.

Table 3.4: Ablation study on the number of Conformer-based $SubbandNet$/$CrossbandNet$ layers (#SB/ #CB) and Conformer feed-forward layer dimension (d_{ff}) in sub-band module

Row	#SB	SB d_{ff}	#CB	CB d_{ff}	SI-SNR (dB)	#Params (M)	GMACs/s
1	8	512	8	512	19.7	2.3	8.8
2	4	512	8	512	17.2	1.4	4.1
3	8	256	8	512	18.8	1.7	5.7
4	8	512	4	512	19.1	2.1	8.3
5	8	512	8	256	19.3	2.1	8
6	4	256	4	256	17	0.7	3.3

As shown in Table 3.4, we identify two crucial hyper-parameters, number of Conformer layers and the feed-forward layer dimension, that significantly influence the performance of Conformer. Row 1 represents our final model configuration. We can observe that reducing the hyper-parameters in $SubbandNet$ leads a significant performance decrease and a drop in $GMACs/s$. However, the impact on $CrossbandNet$ is not as pronounced. More specifically, when comparing Row 1 and

Row 2, we observe a 2.5 dB reduction in SI-SNR and a 50% decrease in $GMACs/s$ when only halving the number of $SubbandNet$ layers. Moreover, comparing Row 1, 4, and 5, as well as Row 2 and 6, reveals that reducing the number of $Crossband$ layers and the corresponding feed-forward layer dimension results in minimal drops in performance and $GMACs/s$. Consequently, we posit that the $SubbandNet$, which operates on individual sub-bands, has a more significant impact on the performance of the sub-band module.

In the full-band module part, we assess the effectiveness of the self-attention module. We design experiments by removing the entire full-band self-attention module and modifying the number of attention heads. Specifically, we carry out the experiments on the entire FSBNet, modifying only the number of attention heads and whether to use the full-band module. All other hyper-parameters remain the same as our final model configuration.

Table 3.5: Ablation study on full-band self-attention module

Row	Models	Use full-band module?	L	SI-SNR (dB)	#params (M)	GMACs/s
1	FSBNet	NO	-	19.7	2.3	8.8
2	FSBNet	YES	1	19.9	2.3	9.1
3	FSBNet	YES	2	20.2	2.4	9.5
4	FSBNet	YES	4	20.6	2.4	9.8
4	FSBNet	YES	6	20.6	2.6	10.5

As shown in Table 3.5, we can observe that our proposed full-band self-attention module is beneficial. Using 4 attention heads works better than using only one. However, increasing the number of attention heads leads to over-fitting during training. The SI-SNR value in training loss increase to 21 dB, but the validation loss remains 20.6. We believe that 20.6 dB SI-SNR value represents the best performance we can achieve. In terms of the model efficiency, we observe that the proposed full-band self-attention module only slightly increase the $GMACs/s$ and does not change to the number of model parameters. The reasons for this are twofold. First, we apply Conv2D layers to compress the input representation before computing the attention output, which introduces only a negligible number of parameters. Second, we operate on the entire frame-level rather than on the decomposed sub bands or time frames. The attention metrics only require $\mathcal{O}(B \times L \times T^2)$ memory cost, rather than $\mathcal{O}(B \times L \times F \times T^2)$ and $\mathcal{O}(B \times L \times T \times F^2)$ in two-stage models. As a result, our

proposed FSBNet is an overall better model than two-stage models, especially in terms of efficiency.

3.5 Summary

In this chapter, we introduce a novel architecture that integrates sub-band module and full-band module to address the 2-speaker speech separation task. The new architecture consists of an encoder, multiple full-band and sub-band blocks (FSB blocks), and a decoder. The encoder extracts STFT information from the raw data and encodes the speech spectrogram to obtain a high-dimensional feature representation via a Conv2D networks. The proposed FSB block is proposed to extract local temporal information, cross band and long-range information of the entire frame-level. Each FSB block consists of a sub-band module and a full-band module. The sub-band module extracts temporal information within each sub bands, and computes the high-level cross-band dependencies by applying compact latent summaries. The full-band module extracts the long-range information by applying self-attention module. The contextual information obtained from FSB blocks will further processed by a Deconv2D to reshape into 2 complex spectrograms, representing for 2 separated speech and re-synthesised into separated speeches via ISTFT.

The experimental results show that our proposed model achieves competitive performance among existing methods including time domain and T-F domain, while demonstrating significantly improved efficiency. Moreover, we show that the architecture can outperform most time domain efficient models for the first time since 2019.

Chapter 4

Conclusions and Future work

4.1 Summary of the work

In this thesis, novel resource-efficient deep learning approaches have been proposed for single-channel 2-speaker speech separation, with a focus on time-frequency domain methods. To address the challenges of long-range dependency in speech signals and the sparse nature of spectrograms, we have designed specific model structures and neural network modules. These include a redundancy-augmented two-stage Conformer, an attention-based full-band module, and a compact latent summaries-based sub-band module. These innovations enable the extraction of sufficient information to achieve competitive performance compared to both time-frequency domain models and time-domain models.

In Chapter 2, we introduce RCFormer, a Conformer-based neural network designed for mono-channel two-speaker speech separation with a focus on resource efficiency. RCFormer utilizes a compact model and processes sparse spectrograms to achieve its goals. The architecture is built on a mask net framework, which includes an encoder, two-stage Conformer blocks, a masking module, and a decoder. The core component, the two-stage Conformer block, captures both frame-level and sub-band-level information from the input spectrogram. The intra-frame Conformer block models local spectral information within each frame, while the sub-band Conformer block captures temporal information across each sub-band. To address the issue of sparse information in the spectrogram,

we use a redundancy approach to stack the input spectrogram embeddings into a denser representation. The output from the two-stage model is then processed through a masking decoder to generate two masks, which are applied to filter the input spectrogram and facilitate the reconstruction of the separated speech signals. Experimental results show that RCFormer delivers exceptional performance in speech separation compared to existing methods. Additionally, RCFormer effectively balances performance and efficiency.

In Chapter 3, we propose a novel architecture that integrates sub-band and full-band modules to address the two-speaker speech separation task. The architecture consists of an encoder, multiple full-band and sub-band blocks (FSB blocks), and a decoder. The encoder extracts STFT information from the raw audio and encodes the speech spectrogram into a high-dimensional feature representation. The proposed FSB block is designed to capture local temporal information, cross-band interactions, and long-range dependencies at the frame level. Each FSB block comprises a sub-band module and a full-band module. The sub-band module extracts temporal information within each sub-band and computes high-level cross-band dependencies through compact latent summaries. The full-band module, on the other hand, captures long-range dependencies using a self-attention mechanism. The contextual information obtained from the FSB blocks is then processed to reshape it into two complex spectrograms, representing the separated speech signals, which are then re-synthesized into separated audio using ISTFT. Experimental results demonstrate that our proposed model achieves competitive performance compared to existing methods, including both time-domain and time-frequency domain approaches, while showing significant improvements in efficiency. Furthermore, we highlight that this architecture outperforms most efficient time-domain models for the first time since 2019.

4.2 Future work

This thesis primarily addresses deep learning methods for single-channel speech separation. However, real-world scenarios frequently involve multi-channel speech separation tasks. Our proposed neural networks are adaptable for multi-channel scenarios with minimal modifications. Additionally, our models are non-causal, meaning the separation process begins after analyzing the

entire utterance. For practical applications like real-time online meetings, where real-time operation is essential, future work could explore adapting our models for causal or real-time settings to enhance their applicability.

Furthermore, our proposed FSBNet has proven to be an effective architecture and can be integrated with other advanced deep learning models. For instance, incorporating recent feature learning models such as Mamba-Net and Mossformer with FSBNet could potentially enhance overall performance.

In this thesis, we primarily conduct experiments using clean datasets, where the input mixtures to our models are free from noise. However, achieving a noise-free environment is nearly impossible in real-world applications. Consequently, we plan to extend our evaluation by training and testing our proposed models on noisy datasets, where input mixtures are contaminated with environmental noise. Datasets such as LibriMix and WHAM! will be used for this purpose.

Bibliography

[1] S. Venkataramani, J. Casebeer, and P. Smaragdis, "End-to-end source separation with adaptive front-ends," *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pp. 684–688, 2018.

[2] S. Choi, A. Cichocki, H.-M. Park, and S.-Y. Lee, "Blind source separation and independent component analysis: A review," *Neural Inf. Process. Lett. Rev*, vol. 6, 11 2004.

[3] M. Kolbæk, D. Yu, Z.-H. Tan, and J. Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1901–1913, 2017.

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

[5] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 241–245, 2017.

[6] M. Cooke, J. R. Hershey, and S. J. Rennie, "Monaural speech separation and recognition challenge," *Computer Speech and Language*, vol. 24, no. 1, pp. 1–15, 2010.

[7] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1979.

[8] Y. Shao, S. Srinivasan, Z. Jin, and D. Wang, "A computational auditory scene analysis system for speech segregation and robust speech recognition," *Comput. Speech Lang.*, vol. 24, no. 1, p. 77–93, jan 2010. [Online]. Available: https://doi.org/10.1016/j.csl.2008.03.004

[9] S. Srinivasan, N. Roman, and D. Wang, "Binary and ratio time-frequency masks for robust speech recognition," *Speech Communication*, vol. 48, no. 11, pp. 1486–1501, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167639306001129

[10] C. K. A. Reddy, A. Ganguly, and I. Panahi, "Ica based single microphone blind speech separation technique using non-linear estimation of speech," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5570–5574, 2017.

[11] M. Schmidt and R. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," *INTERSPEECH 2006 - ICSLP, Ninth International Conference on Spoken Language Processing*, 09 2006.

[12] T. Virtanen, "Speech recognition using factorial hidden markov models for separation in the feature space," *INTERSPEECH 2006 - ICSLP, Ninth International Conference on Spoken Language Processing*, 09 2006.

[13] K. Wang, "Novel deep learning approaches for single-channel speech enhancement," Master's thesis, Concordia University, July 2022, unpublished. [Online]. Available: https://spectrum.library.concordia.ca/id/eprint/990859/

[14] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 21–25, 2021.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, p. 1097–1105, 2012.

[16] L. Zhang, Z. Shi, J. Han, A. Shi, and D. Ma, "Furcanext: End-to-end monaural speech separation with dynamic gated dilated temporal convolutional networks," *MultiMedia*

Modeling: 26th International Conference, MMM 2020, Daejeon, South Korea, January 5–8, 2020, Proceedings, Part I, p. 653–665, 2020. [Online]. Available: https://doi.org/10.1007/978-3-030-37731-1_53

[17] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1003–1012, 2017.

[18] D. Stoller, S. Ewert, and S. Dixon, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation," *ArXiv*, vol. abs/1806.03185, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:47015908

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *arXiv preprint arXiv:1505.04597*, 2015. [Online]. Available: https://arxiv.org/abs/1505.04597

[20] Y. Luo and N. Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[21] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, D. Wu, M. Carpuat, X. Carreras, and E. M. Vecchi, Eds. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 103–111. [Online]. Available: https://aclanthology.org/W14-4012

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, p. 1735–1780, nov 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[23] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path rnn: Efficient long sequence modeling for time-domain single-channel speech separation," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 46–50.

[24] Z. Shi, R. Liu, and J. Han, "La furca: Iterative context-aware end-to-end monaural speech separation based on dual-path deep parallel inter-intra bi-lstm with attention," *ArXiv*, vol. abs/2001.08998, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:210911710

[25] E. Nachmani, Y. Adi, and L. Wolf, "Voice separation with an unknown number of multiple speakers," in *Proceedings of the 37th International Conference on Machine Learning*, ser. ICML'20. JMLR.org, 2020.

[26] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *ArXiv*, vol. abs/1508.01991, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:12740621

[27] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:11212020

[28] 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)*, 2019, pp. 249–253.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6000–6010.

[30] Z.-Q. Wang, J. L. Roux, and J. R. Hershey, "Alternative objective functions for deep clustering," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 686–690.

[31] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 246–250.

[32] 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)*, 2015, pp. 708–712.

[33] Z. Wang, J. L. Roux, D. Wang, and J. R. Hershey, "End-to-end speech separation with unfolded iterative phase reconstruction," *ArXiv*, 2018, accessed August 15, 2024. [Online]. Available: https://arxiv.org/abs/1804.10204

[34] J. Chen, Z. Wang, D. Tuo, Z. Wu, S. Kang, and H. Meng, "Fullsubnet+: Channel attention fullsubnet with complex spectrograms for speech enhancement," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7857–7861.

[35] 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, 2016.

[36] F. Bahmaninezhad, S. Zhang, Y. Xu, M. Yu, J. H. Hansen, and D. Yu, "A unified framework for speech separation," *ArXiv*, 2019, accessed August 15, 2024. [Online]. Available: https://arxiv.org/abs/1912.07814

[37] L. Yang, W. Liu, and W. Wang, "Tfpsnet: Time-frequency domain path scanning network for speech separation," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6842–6846.

[38] C. Chen, C.-H. H. Yang, K. Li, Y. Hu, P.-J. Ku, and E. Chng, "A neural state-space modeling approach to efficient speech separation," in *INTERSPEECH 2023*, 08 2023, pp. 3784–3788.

[39] K. Goel, A. Gu, C. Donahue, and C. R'e, "It's raw! audio generation with state-space models," in *International Conference on Machine Learning*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:247011489

[40] T. Zhao, C. Bao, X. Yang, and X. Zhang, "Dptnet-based beamforming for speech separation," in *2022 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, 2022, pp. 1–5.

[41] M. Sperber, J. Niehues, G. Neubig, S. Stüker, and A. H. Waibel, "Self-attentional acoustic models," in *Interspeech*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:4427800

[42] A. Gu, K. Goel, and C. R'e, "Efficiently modeling long sequences with structured state spaces," *ArXiv*, vol. abs/2111.00396, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:240354066

[43] C. Li, L. Yang, W. Wang, and Y. Qian, "Skim: Skipping memory lstm for low-latency real-time continuous speech separation," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 681–685.

[44] L. Libera, C. Subakan, M. Ravanelli, S. Cornell, F. Lepoutre, and F. Grondin, "Resource-efficient separation transformer," in *2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 04 2024, pp. 761–765.

[45] Z.-Q. Wang, S. Cornell, S. Choi, Y. Lee, B.-Y. Kim, and S. Watanabe, "Tf-gridnet: Integrating full- and sub-band modeling for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 3221–3236, 2023.

[46] Y. Liu, B. Thoshkahna, A. A. Milani, and T. T. Kristjansson, "Voice and accompaniment separation in music using self-attention convolutional neural network," *ArXiv*, vol. abs/2003.08954, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:214606014

[47] 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.

[48] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 31–35.

[49] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. L. Roux, "Wham!: Extending speech separation to noisy environments," *ArXiv*, 2019, accessed August 15, 2024. [Online]. Available: https://arxiv.org/abs/1907.01160

[50] J. Cosentino, M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, "Librimix: An open-source dataset for generalizable speech separation," 2020.

[51] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "Sdr – half-baked or well done?" in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 626–630.

[52] P. C. Loizou, *Speech Enhancement: Theory and Practice*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2013.

[53] 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, 2008.

[54] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *ArXiv*, vol. abs/1905.11946, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:167217261

[55] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML'10. Madison, WI, USA: Omnipress, 2010, p. 807–814.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1026–1034.

[57] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, p. 1929–1958, jan 2014.

[58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:206594692

[59] A. Gulati, J. Qin, C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *ArXiv*, 2020, accessed August 15, 2024. [Online]. Available: https://arxiv.org/abs/2005.08100

[60] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, 2015, p. 448–456.

[61] G. Bebis and M. Georgiopoulos, "Feed-forward neural networks," *IEEE Potentials*, vol. 13, no. 4, pp. 27–31, 1994.

[62] Y. Lu, Z. Li, D. He, Z. Sun, B. Dong, T. Qin, L. Wang, and T. Liu, "Understanding and improving transformer from a multi-particle dynamic system point of view," *ArXiv*, 2019, accessed August 15, 2024. [Online]. Available: https://arxiv.org/abs/1906.02762

[63] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *ArXiv*, vol. abs/1607.08022, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:16516553

[64] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:9433631

[65] N. Takahashi and Y. Mitsufuji, "D3net: Densely connected multidilated densenet for music source separation," *ArXiv*, 2020, accessed August 15, 2024. [Online]. Available: https://arxiv.org/abs/2010.01733

[66] A. Paszke, S. Gross, F. Massa, A. Lerer *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS)*, vol. 32, 2019.

[67] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:6628106

[68] N. Zeghidour and D. Grangier, "Wavesplit: End-to-end speech separation by speaker clustering," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2840–2849, 2021.

[69] J. Rixen and M. Renz, "Sfsrnet: Super-resolution for single-channel audio source separation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, pp. 11 220–11 228, Jun. 2022. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/21372

[70] ——, "QDPN - Quasi-dual-path Network for single-channel Speech Separation," in *Proc. Interspeech 2022*, 2022, pp. 5353–5357.

[71] S. Lutati, E. Nachmani, and L. Wolf, "Sepit: Approaching a single channel speech separation bound," in *Interspeech*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:249017815

[72] S. Zhao and B. Ma, "Mossformer: Pushing the performance limit of monaural speech separation using gated single-head transformer with convolution-augmented joint self-attentions," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[73] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016, accessed August 15, 2024.

[74] B. Van Veen and K. Buckley, "Beamforming: a versatile approach to spatial filtering," *IEEE ASSP Magazine*, vol. 5, no. 2, pp. 4–24, 1988.

[75] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96. AAAI Press, 1996, p. 226–231.

[76] W. Ravenscroft, S. Goetze, and T. Hain, "Deformable temporal convolutional networks for monaural noisy reverberant speech separation," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.