APSIPA Transactions on Signal and Information Processing, 2023, 12, e19

Original Paper

A Dual-branch Convolutional Network Architecture Processing on both Frequency and Time Domain for Single-channel Speech Enhancement

Kanghao Zhang[1], Shulin He[1], Hao Li[2] and Xueliang Zhang[1*]

[1] *College of Computer Science, Inner Mongolia University, China*
[2] *Department of Electrical and Electronic Engineering, Southern University of Science and Technology, China*

ABSTRACT

Single-channel speech enhancement aims to remove the interfering noise and reverberation in real environments by a single microphone, which is a very challenging task in the speech signal processing field. Over the past years, deep learning has shown great potential for speech enhancement. In this paper, we propose a novel real-time framework, called DBCN, which is a dual-branch architecture. One branch takes waveform as its input for time-domain modeling and the other one takes shift real spectrum as input for frequency-domain modeling. The two branches have the same network structure, which is the representative convolutional recurrent network. To exchange information sufficiently, a bridge module is added between the two branches. Furthermore, we propose a novel feature normalization approach that enables each band to complete the normalization independently by counting the root mean square of each band and obtaining the inter-frame relationship for each band. The proposed approach allows the network to ignore the magnitude during processing, reducing learning difficulty and improving performance. Systematical evaluation and comparison are conducted. Experimental

*Corresponding author: Xueliang Zhang, cszxl@imu.edu.cn

Received 08 December 2022; Revised 20 March 2023
ISSN 2048-7703; DOI 10.1561/116.00000083

results show that the proposed system substantially outperforms related algorithms for causal and non-causal speech enhancement under very challenging environments.

Keywords: Deep learning, speech enhancement, time-domain processing, frequency-domain processing, feature normalization.

1 Introduction

Teleconferencing is becoming more and more popular at present, particularly during the COVID-19 pandemic. However, various noises in real environments can severely interfere with normal speech communication. To recover clean speech from the noise-contaminated mixture, speech enhancement is indispensable in almost all speech communication devices. Although the study of speech enhancement has a long history and many methods are invented in the literature [20], it is still a challenging task in practice, particularly for the single-channel scenario where only one microphone is available.

Spectral subtraction [2] is a classic method for single-channel speech enhancement, which subtracts the estimated spectrum of noises from the mixture. However, it is very difficult to estimate the spectrum of non-stationary noise which varies dramatically with time, and spectral subtraction becomes invalid in this most common scene. To overcome the shortcomings of traditional speech enhancement methods [2, 5, 37], Wang *et al.* first introduced a deep neural network for speech separation [48], which decomposes the input mixture into time-frequency representation and estimate the time-frequency (T-F) binary mask to recover the clean speech. The experimental results showed the great potential of DNN for speech enhancement task.

Since then, a large number of methods for speech enhancement using deep learning have been extensively studied, most of which exploit the T-F structure. Their training objectives can be divided into two main streams, one is masking-based and the other is mapping-based. For the masking-based targets, such as ideal binary mask (IBM) [44] and ideal ratio mask (IRM) [47], the network models the relationship between noisy speech and clean speech, and the mask is learned to cover the noisy speech to remove noise. For the mapping-based targets [21], such as spectral magnitude and complex spectral, the network learns a mapping and directly outputs clean speech. The training target together with the input features trains a DNN and obtains the enhanced waveform by reconstructing the estimated target.

Since phase is difficult to estimate, the early enhancement methods [18, 40] only enhance the magnitude and use the noisy phase for reconstruction. One example is the convolutional recurrent network (CRN) proposed by Tan

et al. [40], which incorporated convolutional encoder-decoder (CED) and long short-term memory (LSTM) into the CRN architecture, and finally trained a magnitude-based mapping network. However, subsequent research has shown that incorporating phase into the supervised learning step can be effective in enhancing the listener's subjective perception [27]. At present, many approaches combining magnitude enhancement and phase perception have been proposed [13, 46, 52, 54, 55], which can be roughly classified into time-domain speech enhancement and complex frequency-domain speech enhancement. In complex spectrogram enhancement, the real and imaginary components of the complex-valued noise STFT (Short Time Fourier Transform) are simultaneously enhanced to recover the complex spectrogram of clean speech, and also indirectly recover the phase information by learning the relationship of the RI component information [11, 15, 16, 25, 41, 50]. Compared with complex frequency-domain approaches, time-domain methods [10, 17, 22, 26, 29, 30, 34] have several obvious advantages. First, STFT for frequency domain processing usually requires multiple vibration periods in one frame to analyze the frequency characteristics, while time domain processing has no requirement for frame length. This means that temporal networks can model data at a finer scale. Second, the time-domain approach avoids the computations associated with converting between the two domains. That is, the raw speech is used directly for regenerated speech enhancement without going through the STFT process. Luo *et al.* used time-domain models for speech separation, and successively proposed TasNet, Conv-TasNet and DPRNN [22–24]. All of them used extremely small frame lengths to replace the STFT, which greatly improved the performance of the model. Subsequently, many similar methods have been widely explored in the field of speech enhancement [7, 31, 33, 51]. However, time-domain methods usually require more trainable parameters and greater model complexity, which is difficult to apply in practical scenarios. Therefore, making full use of their respective advantages is still a problem worth exploring.

Most mainstream enhancement models are still trained either in frequency or in time domain, and some researchers use cross-domain constraints or cross-domain training for speech enhancement. Pandey *et al.* [30] combined the loss of the time-domain waveform and the loss of the frequency domain in a certain ratio to train the model, and Wang *et al.* [45] stringed together three different domain modules for cross-domain joint training. Both of them achieve great performance. In fact, the superiority of the cross-domain model is theoretically based. Models in different domains may play different roles in dealing with different types of noise. For impulse noise as shown in the left column of Figure 1, it is easy to eliminate in the time domain. Only a few samples need to be removed. When it comes to the frequency domain, the noise pollutes the entire frequency band, and it is difficult to be eliminated on the frequency domain. In contrast, for a narrow band noise like a pure tone as shown in the right column of Figure 1, the noise is distributed on the

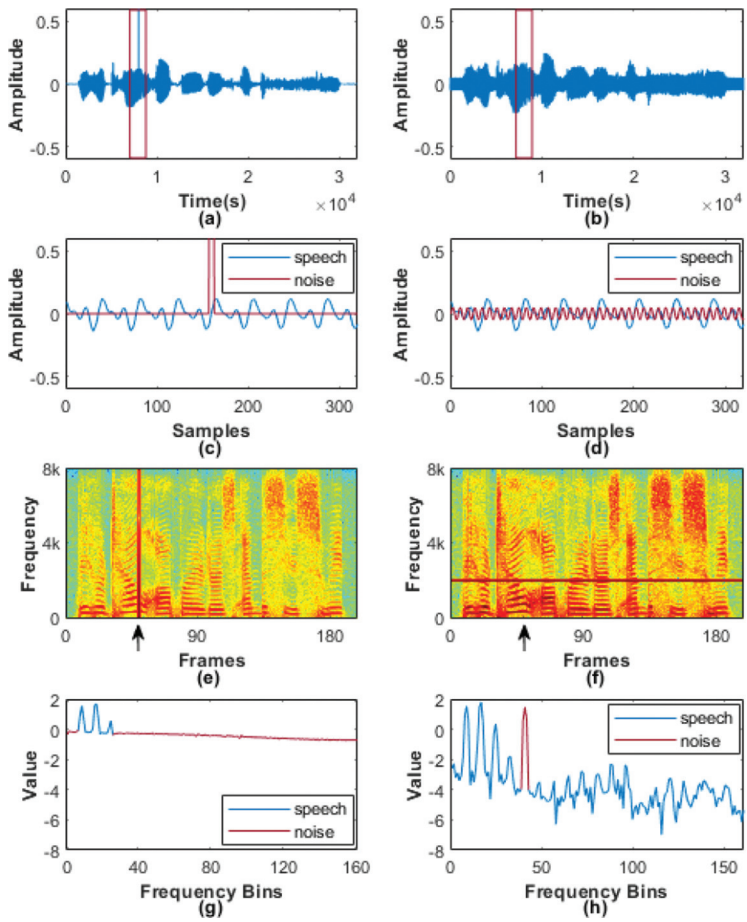


Figure 1: Here is a schematic diagram illustrating the difference between impulse noise and narrow-band noise. The left column contains: (a) the waveform of a mixed speech with impulse noise, (c) the waveform diagram of the frame inside the red box in (a), (e) the log power spectrum corresponding to the waveform in (a), (f) the frequency diagram of the frame which is pointed out in (e) by an arrow. The right column contains corresponding schematic representations of mixed speech with narrow-band noise.

narrow band and the frequency domain-based method covers well. In the time domain, the noise and the speech are coupled together at every sample, and it is hard to decouple on time domain. Therefore, we proposed a dual-branch network architecture [53] processing on both frequency- and time-domain, and the experimental results show impressive enhancement performance.

The dual-branch network consists of two convolutional recurrent networks (CRN) to process the time and the frequency representations respectively. At the same time, two branches exchange information after each convolution except

the output layer. The key idea is to learn feature representations from two different perspectives in the time domain and the frequency domain and provide the learned features for each other as a reference to improve performance. From the complementary perspective, the time domain branch directly enhances the original speech, which avoids the problem of signal distortion caused by invalid STFT [8]. In addition, the time domain calculates all the discrete sample points, resulting in more attention to detail and prompting the network to eliminate some time-sensitive noises. In contrast, the frequency domain branch is responsible for suppressing the main noise components and eliminating frequency-sensitive noise. Note that we convert the complex spectrum to shift real spectrum (SRS) [38] losslessly to ensure that all operations of the network are in the real-valued domain and make the interaction of the two branches smooth. The contributions of this work are as follows,

1. We propose a dual-branch convolutional network structure that combines time-domain and frequency-domain processing. We use a bridge layer to facilitate the flow and fusion of cross-domain information. This allows our model to effectively exploit the complementary nature of these two domains and improve performance. By fully comparing our approach with multiple baseline methods, we demonstrate the effectiveness of our proposed network structure.

2. Based on the dual-branch convolutional network, we propose a novel feature normalization method. This method normalizes each frequency band independently by calculating the root mean square of each frequency band. It makes the network independent of the input amplitude and leads to better generalization ability for real-time scenarios.

3. We conducted a preliminary exploration of the information fusion layer and investigated the effects of different initialization methods on the performance of the fusion layer. This research forms the basis for further development and optimization of the fusion layer in our proposed network structure.

4. We used three different combinations of loss functions to constrain the time-domain branch and the frequency-domain branch separately. This allowed us to investigate the behavior of our network under different loss function configurations and validated the superiority of the proposed cross-domain loss function combination. We also conduct ablation experiments to evaluate the contribution of the proposed modules.

The remainder of the paper is organized as follows. In Section 2, the time-domain speech enhancement problem and shift real spectrum transformation are described. In Section 3, we present the proposed dual-branch network in detail. In Section 4, the experimental setting and data setup are given.

Experimental results are displayed along with the analysis in Section 5. Finally in Section 6 we conclude the paper.

2 Problem Formulation

2.1 Monaural Speech Enhancement

Given a single-microphone noisy mixture m, monaural speech enhancement aims to separate target speech s from background noise n. A noisy mixture can be formulated as:

$$m[k] = s[k] + n[k], \tag{1}$$

where k denotes the time index.

In the time domain, the algorithm aims to get $\hat{s}$ directly from mixture y rather than a T-F representation of y. The process for time-domain speech enhancement using DNN can be expressed as:

$$\hat{s} = \phi_\theta(m), \tag{2}$$

where ϕ_θ represents a function defined by a DNN model parameterized by θ. Generally, a speech enhancement network is designed to process frames of the speech signal. Given a speech signal s, it is first chunked into overlapping frames before being processed by the DNN model ϕ_θ. Let M denote a matrix consisting of frames of the signal m, and M_t denote the t^{th} frame of the signal, then M_t can be formulated as:

$$M_t[i] = m[(t-1) \cdot J + i], i = 0, \ldots, K-1, \tag{3}$$

where K is the frame length and J is the frame shift. The number of frames T is calculated from the signal length L and the frame shift J and expressed as $\lfloor \frac{L}{J} \rfloor$, where $\lfloor\ \rfloor$ denotes the floor function. Then a target frame can be computed by:

$$\hat{s_t} = \phi_\theta(M_{t-T_1}, \ldots, M_{t-1}, M_t, M_{t+1}, \ldots, M_{t+T_2}), \tag{4}$$

where $\hat{s_t}$ is computed using M_t, T_1 past frames, and T_2 future frames.

2.2 Shift Real Spectrum (SRS)

In addition to the time domain, noisy speech is usually processed in the T-F domain. The two major frequency-domain methods are to enhance speech on the magnitude spectrum and the complex spectrum, respectively. The former ignores the phase and couples the predicted magnitude with the noisy phase in the recovery phase, causing a deviation from the target speech. The latter

models the real and imaginary parts at the same time, which requires further exploration of the relationship between the real and imaginary parts and also brings large complexity. Soni *et al.* proposed an alternative to complex spectrum in their study [38], called shift real spectrum (SRS). By applying SRS, frequency domain information can be learned in the real field [19].

Given a time-domain speech s, we can use discrete-time Fourier transform (DTFT) to obtain a complex spectrum S which consists of a real part (denoted as S_R) and an imaginary part (denoted as S_I). As such, $S = S_R + jS_I$ where j is an imaginary unit, and the speech s can be expressed as (S_R, S_I) losslessly. More specifically, the real part S_R is an even function that consists of a series of cosine basic functions, while the imaginary part S_I is an odd function that stands for a superposition of a series of sine functions. That is, any signal s in the time domain can be described as:

$$s = s_{even} + s_{odd}, \tag{5}$$

where s_{even} and s_{odd} denote the IDTFT(S_R) and IDTFT($j \cdot S_I$), respectively. When the signal s is an odd function, the even part can be ignored because $s_{odd} = 0$. Likewise, the odd part can be ignored when the signal is an even function. By padding the signal with zeros of appropriate length as Figure 2 shown, we can decompose the signal into an even and an odd function, whose amplitude is half of the signal s. Thus, the original signal can be described as $s = 2 \cdot IDTFT(S_R)$. In practice, we first separate the signal into windowed frames. Then we apply STFT to these windowed frames. Finally, the real part is taken as the representation. In the remainder of this article, all the mentioned STFT processes are performed in this way.

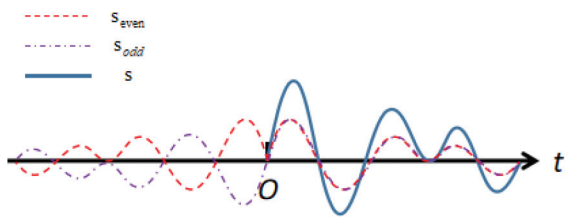


Figure 2: An example of zero-padded singal. The blue line is the original signal. The red and purple dashed lines denote the odd and even signals, respectively.

3 Dual-Branch Architecture

In this section, the details of the proposed dual-branch network are introduced. As shown in Figure 3, the proposed dual-branch network consists of two main modules: the time-domain module CRN-Time and the frequency-domain

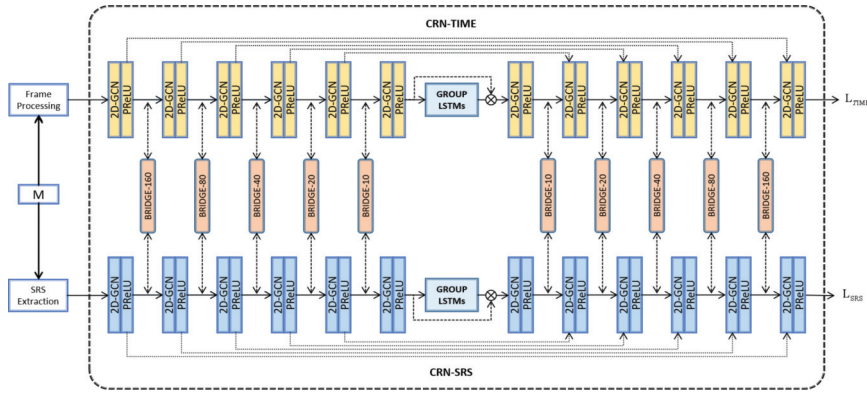


Figure 3: A schematic diagram illustrating the dual-branch architecture. The upper module CRN-Time employs a CRN to predict the temporal representation, the lower module CRN-SRS employs the same structure to predict the frequency-domain representation. After each layer of convolution, a bridge layer is used for feature conversion and fusion. Each module get an output and the output of CRN-SRS represents the outcome of the proposed network because of its better performance in the evaluation stage.

module CRN-SRS. The input of the two-branch network is a noisy mixture. CRN-Time module takes 320-point frames with 160-point overlapping as input and generates a time-domain estimate of the clean frames which is then converted back to speech. The CRN-SRS module takes the SRS obtained by the ISRS operation as input and estimates the clean SRS representation. Note that each branch has an independent output and the output of CRN-SRS is obtained by ISRS. We adopt the same structure for the time domain branch and the frequency domain branch. This makes the features of the two branches have the same dimension, which ensures the smooth flow of information in the two domains. Secondly, since the best feature conversion structure has not been found for the time being, we hope to achieve the SRS-liked transformation by using the trainable parameters, which also require the same dimension. In the following subsections, we will introduce several key modules, network configurations, and loss functions.

3.1 Gated Convolution And Grouping Strategy for RNN

A gating mechanism, which controls the information flows of the network, is first designed for long short-term memory (LSTM) [9] based on RNN. This could allow modeling more complicated interactions and enable RNNs to achieve better performance. Van den Oord *et al.* added a gating mechanism to the convolutional layer to model the image in their study [43], and it can

be described as:

$$\begin{aligned} y &= tanh(x * W_1 + b_1) \odot \sigma(x * W_2 + b_2) \\ &= tanh(v_1) \odot \sigma(v_2), \end{aligned} \tag{6}$$

where Ws and bs denote kernels and bias, respectively, and σ represents sigmoid function. Convolution and element-wise multiplication are denoted by the symbols $*$ and $\odot$, respectively. However, the gradient gradually disappears as the network deepens. To alleviate this phenomenon, a similar structure was proposed by Dauphin *et al.* [4] which can be expressed as:

$$\begin{aligned} y &= x * W_1 + b_1 \odot \sigma(x * W_2 + b_2) \\ &= v_1 \odot \sigma(v_2). \end{aligned} \tag{7}$$

The removal of the *tanh* function makes the entire unit treated as a multiplicative skip connection, allowing the gradient to pass through the layers smoothly. An example of a gated convolutional unit (denoted as "GCN") as well as a gated deconvolutional unit (denoted as "DeGCN") is shown in Figure 4. In our network, the feature normalization is introduced in the convolution, which is explained in the latter subsection. A feature normalization layer is placed before the sigmoid of the gated convolution. In addition, the bias of the convolution is set to zero to prevent it from destroying the linear structure of the feature.

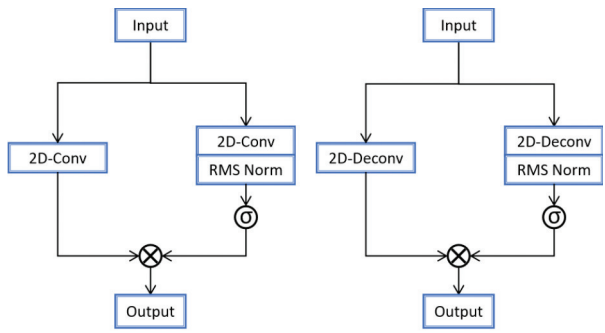


Figure 4: Diagrams of a gated convolutional unit and a gated deconvolutional unit, where σ denotes a sigmoid function.

The application of LSTM allows modeling long-term dependencies and using deeper layers, but also introduces enormous complexity. However, the efficiency of the model is quite significant for practical applications, for example, the microphone noise reduction program requires efficient computation and a small memory footprint. To mitigate this problem, a grouping strategy is proposed by Gao *et al.* [6] to improve the model efficiency. The layers of a

typical LSTM are fully connected, and the number of inter-layer connections increases dramatically with the number of nodes. As illustrated in Figure 5, the grouping strategy divides the input features and hidden states of each recurrent layer into two groups evenly, and each group processes the internal features independently. In addition, a frame-level rearrangement layer is applied between consecutive recurrent layers to alleviate the problem of the inability to build mutual dependencies between separate groups. And this rearrangement layer brings together distantly-located frames into one group, which potentially allows for learning long-span features. We employ this grouping strategy for the LSTM layers in the proposed dual-branch network and set the number of groups to 2, which is proven to be the most efficient and effective setting in [41].

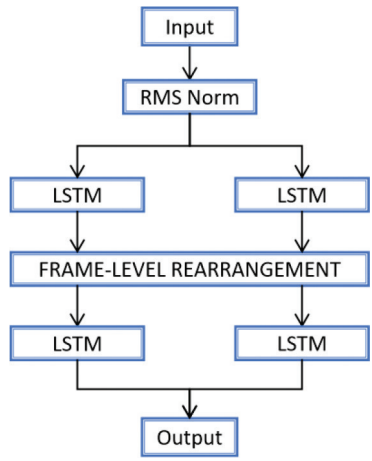
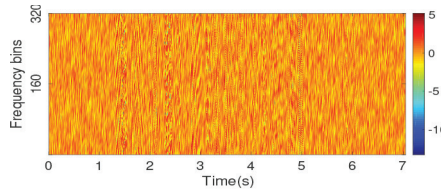


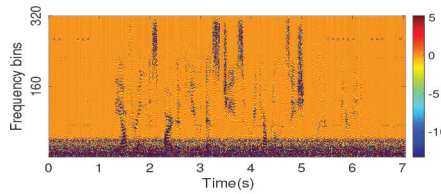
Figure 5: Illustration of a grouped long-short time memory (GLSTM) module.

3.2 Bridge Module

Figure 6a and 6b show a feature map of the CRN-Time module and the CRN-SRS module, respectively. It can be easily found that their feature maps exhibit distinct patterns, so coarse connections may introduce unnecessary modeling difficulties. Therefore, a bridge module is applied between each convolutional layer from the two branches. The bridge module can be regarded as an information transfer module between different domains, which is responsible for converting information from one branch to another. More specifically, the bridge module is composed of two $K \times K$ independent matrices, where K denotes the frame length. Each transformation matrix is trainable and used for a unidirectional transformation process (from the time domain to frequency domain or vice versa). In practical applications, we determine the initial value



(a) A feature map of the CRN-Time branch.



(b) A feature map of the CRN-SRS branch.

Figure 6: Example feature maps from CRN-TIME and CRN-SRS, they exhibit quite different feature patterns and the values vary widely.

of the matrix in three ways: random initialization, specific initialization using the real part of fast Fourier transform (FFT) parameters or SRS parameters. We found that initializing the matrix with SRS parameters leads to better performance of the model.

3.3 Feature Normalization

Normalization is a technique to improve generalization ability and facilitate DNN training. Batch normalization [12], layer normalization [1], and instance normalization [42] are widely used for speech enhancement. For a given 4-dimensional input of $Batch \times Channel \times Timeframes \times Frequencybins$, the three normalization methods calculate the statistics of [B, T, F], [C, T, F] or [T, F] slices respectively, which makes them suitable for different application scenarios. However, we found that the augmentation ability of the network is deprived when the magnitude range of the input is inconsistent with the training samples. In this paper, we propose the feature normalization which can be described as:

$$y^{norm} = \frac{y}{R_y + \epsilon} \odot \gamma + \beta, \tag{8}$$

$$R_y = \sqrt{\frac{1}{C \cdot T} \sum_{c=0}^{C-1} \sum_{t=0}^{T-1} y_{c,t}^2}, \tag{9}$$

where $y^{norm} \in \mathbb{R}^{B \times C \times T \times F}$ and $R_y \in \mathbb{R}^{B \times 1 \times 1 \times F}$, respectively, are the normalized output and the root mean square of the input data. C and T represent

the channels and frames of input data. $\gamma \in \mathbb{R}^{1\times1\times1\times F}$ and $\beta \in \mathbb{R}^{1\times1\times1\times F}$ are trainable variables of the same size as frequency bins, and $+$ and $\odot$ denote element-wise addition and multiplication. ϵ is a small positive constant to avoid division by zero. As described in the formula, we independently compute the root mean square of the input data along the channel and time dimensions for each frequency bin and perform normalization on each frequency bin. In subsequent experiments, we found that feature normalization improves the enhancement ability and still performs well in noise suppression in the face of various untrained amplitude scenarios.

3.4 Time Module And SRS Module

The two branches of the model use the same structure which is similar to CRN. CRN is an encoder-decoder structure composed of convolutional layers and LSTMs, which combines the feature extraction capabilities of CNNs with the temporal modeling capabilities of RNNs. In the proposed network, CRN-Time is fed with time-domain frames of noisy utterances, and CRN-SRS is fed with frequency-domain frames. The encoder stacks 6 normal convolutions with a stride 2 to downsample along the frequency axis, while the decoder stacks 6 deconvolutions with the same stride for upsampling. A modified version of the gated convolutional unit elaborated in Section III. A is used as a convolutional layer, and each one is followed by a parametric ReLU (PReLU) nonlinearity except the output layer. Each layer of the encoder and decoder not only receives the output of the previous layer but also concatenates the corresponding information transformed from the other branch. The used architecture additionally incorporates skip connections to facilitate optimization, which connects each layer in the encoder to its corresponding layer in the decoder. Additionally, we apply the grouping strategy on the intermediate LSTM, which greatly reduces the model complexity.

Table 1 provides a more detailed description of the proposed network architecture. The input size and the output size of each layer are displayed in inChannels $\times$ timeFrames $\times$ frequencyBins format. The layer hyperparameters are specified in (kernelSize, strides, outChannels) format. For all the convolutions and the deconvolutions, We use a kernel size of 1×3 (time $\times$ frequency) for the causal system and 3×3 for the non-causal system. The number of input channels in each encoder layer is doubled due to the connections of the bridge module, while the number of input channels in the decoder is tripled due to the additional skip connections.

3.5 Loss Functions

The training objective of our dual-branch architecture consists of two parts, corresponding to the outputs generated by these two branches. We employ

Table 1: Architecture of our proposed one branch. Both branches contain the same structure and parameter settings. Here T denotes the number of time frames.

layer name	input size	hyperparameters	output size
GCN2d_1	$1 \times T \times 320$	(1×3), (1,2), 64	$64 \times T \times 160$
Bridge_1	$64 \times T \times 160$	–	$64 \times T \times 160$
GCN2d_2	$128 \times T \times 160$	(1×3), (1,2), 64	$64 \times T \times 80$
Bridge_2	$64 \times T \times 80$	–	$64 \times T \times 80$
GCN2d_3	$128 \times T \times 80$	(1×3), (1,2), 64	$64 \times T \times 40$
Bridge_3	$64 \times T \times 40$	–	$64 \times T \times 40$
GCN2d_4	$128 \times T \times 40$	(1×3), (1,2), 64	$64 \times T \times 20$
Bridge_4	$64 \times T \times 20$	–	$64 \times T \times 20$
GCN2d_5	$128 \times T \times 20$	(1×3), (1,2), 64	$64 \times T \times 10$
Bridge_5	$64 \times T \times 10$	–	$64 \times T \times 10$
GCN2d_6	$128 \times T \times 10$	(1×3), (1,2), 64	$64 \times T \times 5$
reshape_1	$64 \times T \times 5$	–	$T \times 320$
glstm_1	$T \times 320$	320	$T \times 320$
glstm_2	$T \times 320$	320	$T \times 320$
reshape_2	$T \times 320$	–	$64 \times T \times 5$
DeGCN2d_6	$64 \times T \times 5$	(1×3), (1,2), 64	$64 \times T \times 10$
Bridge_5	$64 \times T \times 10$	–	$64 \times T \times 10$
DeGCN2d_5	$192 \times T \times 10$	(1×3), (1,2), 64	$64 \times T \times 20$
Bridge_4	$64 \times T \times 20$	–	$64 \times T \times 20$
DeGCN2d_4	$192 \times T \times 20$	(1×3), (1,2), 64	$64 \times T \times 40$
Bridge_3	$64 \times T \times 40$	–	$64 \times T \times 40$
DeGCN2d_3	$192 \times T \times 40$	(1×3), (1,2), 64	$64 \times T \times 80$
Bridge_2	$64 \times T \times 80$	–	$64 \times T \times 80$
DeGCN2d_2	$192 \times T \times 80$	(1×3), (1,2), 64	$64 \times T \times 160$
Bridge_1	$64 \times T \times 160$	–	$64 \times T \times 160$
DeGCN2d_1	$192 \times T \times 160$	(1×3), (1,2), 64	$64 \times T \times 320$

various loss functions for two outputs to find the best combination. The first loss function is the utterance level mean squared error (MSE) in the time domain which can be defined as:

$$L_{Mse}(s, \hat{s}) = \frac{1}{L} \sum_{k=0}^{L-1} (s[k] - \hat{s}[k])^2. \tag{10}$$

The second one is the mean absolute error loss between the L1 norm of clean and estimated STFT coefficients [32] and can be described as:

$$L_{Mag}(s, \hat{s}) = \frac{1}{T \cdot F} \sum_{t=0}^{T-1} \sum_{f=0}^{F-1} (|S(t, f)| - |\hat{S}(t, f)|), \tag{11}$$

where T and F represent the number of time frames and frequency dimensions, S and $\hat{S}$ denote STFTs of s and $\hat{s}$, respectively. Although L_{Mag} obtains better objective scores, it introduces unnecessary artifacts. Therefore, a loss function based L_{Com} on the complex spectrum [49] is applied.

$$L_{RI}(s,\hat{s}) = \frac{1}{T \cdot F} \sum_{t=0}^{T-1} \sum_{f=0}^{F-1} (|S_r(t,f) - \hat{S}_r(t,f)| + |S_i(t,f) - \hat{S}_i(t,f)|), \tag{12}$$

$$L_{Com}(s,\hat{s}) = L_{Mag}(s,\hat{s}) + L_{RI}(s,\hat{s}), \tag{13}$$

where S_r and $\hat{S}_i$ represent the real and imaginary parts of S, respectively. Finally, we tried three combinations of loss functions: (1) L_{Mse} for the time branch, L_{Mag} for the frequency branch. (2) L_{Mag} for both time- and frequency branch. (3) L_{Com} for both time- and frequency branch. Comparative experiments show that training with the combination (1) can improve the performance of the model, which may indicate that multi-domain constraints are beneficial for model training.

4 Experimental Settings

4.1 Datasets

The proposed dual-branch architecture is evaluated on the WSJ0 SI-84 dataset [35] which includes 7138 utterances from 83 speakers (42 males and 41 females). We select the utterances of 77 speakers (42 males and 41 females) as the original corpus for training and validation sets, while the utterances of the rest 6 speakers (3 males and 3 females) are used to generate the test set. For training, 10000 non-speech sounds from a sound effect library (available at www.sound-ideas.com) [3] are used, and the duration is about 126 h. For testing, we choose two highly challenging noises (babble and cafeteria) from Auditec CD (available at http://www.auditec.com).

During the mixing process, different generative strategies are used for training and testing. For training, we first concatenate all training sentences into one large file and do the same with noises. Then we randomly intercept 7s segments from the speech file and noise file, respectively, and generate mixtures under an SNR uniformly sampled from −5, −4, −3, −2, −1, and −0dB. As a result, a total of 320,000 and 3000 noisy-clean pairs are generated for training and validation, respectively. For testing, 150 utterances from 6 speakers (25 each) are mixed with two selected non-stationary noises at three SNR levels −5, 0, 5 dB. Note that test mixtures are of variable length.

4.2 Experimental Setup

All the utterances are resampled to 16kHz. For SRS operations, Hamming window is used for smoothing. Frame length and frame shift are set to 20ms and 10ms, respectively. For the frame-level operations in the time domain, the 20 ms rectangle window is used to divide each utterance into segments of 320 samples with 50% overlap. All models are developed by PyTorch and trained with stochastic gradient descent optimization using the Adam optimizer [14]. We train the models for 20 epochs with a batch size of 8 at utterance level and keep the learning rate of 0.001. We develop the proposed dual-branch model[1] with one NVIDIA RTX 3090 for two weeks.

4.3 Baselines Models

We compared the proposed model with time-domain and frequency-domain methods. In the frequency domain, CRN and GCRN are very representative. As for the time-domain method, we chose AECNN [28], DDAEC [30], and DCN [29] as our baseline methods. The description of baselines is listed as follows:

- CRN: it is a convolutional recurrent network in the T-F domain. The network uses 5 convolution layers as the encoder and 5 deconvolution layers as the decoder. Two LSTM layers are used for sequence modeling. This network receives magnitude as input. We kept the best configuration in [40].
- GCRN: similar to CRN, it also consists of a convolutional layer and an LSTM with the same structure. The difference is that the GCRN process the signal in the complex domain, and there are two decoders used to recover the real and imaginary parts respectively. Furthermore, a gating mechanism is introduced in the convolutional layers to model more complex interactions, while a grouping strategy is applied to LSTMs to reduce the complexity of the model. We implemented both causal and non-causal versions.
- AECNN: it is an autoencoder-based fully convolutional neural network in the time domain. AECNN directly calculates the clean speech segments from the noisy waveform segments. The hyperparameter settings we adopted are consistent with those designed for the WSJ dataset in the original paper.
- DDAEC: the network is an encoder-decoder based architecture with skip connections. Layers in the encoder and decoder are followed by densely

[1]The code and examples: https://github.com/zhangkanghao/DBCN.

connected blocks, including dilation and causal convolution. Dilated convolutions are used to aggregate contexts of different resolutions. Subpixel convolution is used in the decoder.

- DCN: this is an encoder-decoder based architecture with skip connections. Each layer in the encoder and the decoder comprises a dense block and an attention module. Dense blocks and attention modules help in feature extraction using a combination of feature reuse, increased network depth, and maximum context aggregation. We implemented both causal and non-causal versions. The kernel is set to (1, 3) for the causal system.

4.4 Evaluation Metrics

In our experiments, we use perceptual evaluation of speech quality (PESQ) [36], and short-time objective intelligibility (STOI) [39] as the objective metrics to evaluate the enhancement performance of different models. PESQ is used to evaluate speech quality, with its values ranging from -0.5 to 4.5. STOI values range from 0 to 1, which is used to evaluate speech intelligibility. Note that higher scores for all metrics mean better speech quality.

5 Results, Comparisons and Analyses

5.1 Comparison Results

In this section, we compare the proposed architecture with several excellent baselines and show the results on the WSJ0 SI-84 dataset. Table 2a and Table 2b show the comparison results in terms of the objective metrics STOI and PESQ for two challenging noises at -5 dB, 0 dB, and 5 dB. All the best results are marked in bold in both tables. The proposed dual-branch network (DBCN) yields the best results under all conditions.

For objective metrics, the results in Table 2a and Table 2b show that our proposed models both greatly outperform time-domain DCN and frequency-domain GCRN. Among the frequency-domain baselines, GCRN performs the best. It can achieve an average STOI score of 89.1 and a PESQ score of 2.64 under challenging babble noise. However, the performance of GCRN is far from the proposed method. Compared with the frequency domain, the time-domain method generally has better performance. Among them, the DCN performance is the most prominent. The DBCN still maintains a sufficient advantage in all conditions, especially for the PESQ, which is 0.24 better than DCN on average under babble. In addition to the comparison of causal systems, we provide non-causal versions of GCRN, DCN, and DBCN, denoted GCRN-NC, DCN-NC, and DBCN-NC. In the non-causal version, causal convolutions are replaced by non-causal convolutions, and LSTMs are replaced by bidirectional

Table 2a: Evaluations and comparisons of different enhancement models in terms of STOI(%).

Metrix	STOI								Casual?
Test Noise	Babble				Cafeteria				
Test SNR (dB)	-5	0	5	Avg.	-5	0	5	Avg.	
Mixture	58.5	70.4	81.2	70.0	57.5	69.9	81.1	69.5	
CRN	77.9	88.0	93.2	86.4	75.7	86.6	92.7	85.0	✓
GCRN	81.5	90.8	94.9	89.1	78.5	89.0	94.3	87.3	✓
AECNN	80.5	90.6	94.2	88.4	80.0	89.4	94.0	87.8	✓
DDAEC	84.0	92.2	95.6	90.6	81.7	90.9	94.9	89.2	✓
DCN	83.9	91.8	95.2	90.3	81.0	90.3	94.5	88.6	✓
DBCN	**85.3**	**92.7**	**95.9**	**91.3**	**82.0**	**91.2**	**95.2**	**89.5**	✓
GCRN-NC	84.1	92.1	95.5	90.5	81.3	90.5	95.0	89.0	×
DCN-NC	87.9	93.5	96.1	92.5	85.0	92.1	95.3	90.8	×
DBCN-NC	**88.3**	**94.1**	**96.5**	**93.0**	**85.2**	**92.5**	**96.0**	**91.2**	×

Table 2b: Evaluations and comparisons of different enhancement models in terms of PESQ.

Metrix	PESQ								Casual?
Test Noise	Babble				Cafeteria				
Test SNR (dB)	-5	0	5	Avg.	-5	0	5	Avg.	
Mixture	1.54	1.82	2.12	1.83	1.46	1.77	2.12	1.78	
CRN	1.99	2.50	2.91	2.47	2.01	2.47	2.89	2.46	✓
GCRN	2.08	2.71	3.13	2.64	2.03	2.62	3.09	2.58	✓
AECNN	2.00	2.57	2.88	2.48	2.03	2.55	2.93	2.50	✓
DDAEC	2.33	2.88	3.26	2.82	2.30	2.81	3.19	2.77	✓
DCN	2.23	2.72	3.09	2.68	2.15	2.62	3.01	2.59	✓
DBCN	**2.45**	**2.98**	**3.33**	**2.92**	**2.29**	**2.84**	**3.24**	**2.79**	✓
GCRN-NC	2.22	2.81	3.17	2.73	2.08	2.72	3.07	2.62	×
DCN-NC	2.61	3.04	3.33	2.99	2.45	2.91	3.23	2.86	×
DBCN-NC	**2.63**	**3.12**	**3.38**	**3.05**	**2.48**	**3.01**	**3.35**	**2.95**	×

LSTMs to provide long-term contextual information. The performance of the non-causal system will be significantly improved compared to the causal system because future information is learned. However, we are pleasantly surprised to find that the proposed DBCN outperforms the non-causal system GCRN-NC. This suggests that the proposed model is a highly effective network for speech enhancement even without any context information. We compare DBCN-NC with DCN-NC, the best-performing non-causal baseline system. DBCN-NC obtaine an average improvement of 0.5% and 0.4% for babble and cafeteria in terms of STOI, respectively, and an average improvement of 0.06 and 0.09 for PESQ.

In conclusion, the proposed dual-branch model outperforms both DCN which is a time-domain based model, and GCRN which is a frequency-domain based model for complex spectrogram mapping, indicating that information transformation and fusion of two domains can significantly improve the performance of the model and improve parameter utilization.

Table 3: Ablation study on components of the dual-branch architecture.

Metric	STOI		PESQ		Param.(M)
Test Noise	Babble	Cafeteria	Babble	Cafeteria	
DBCN	**91.3**	**89.5**	**2.92**	**2.79**	2.85
− FN	90.3	88.2	2.73	2.63	2.85
− BG	90.9	88.7	2.80	2.68	2.85
CRN-Time	87.9	86.0	2.51	2.45	1.17
CRN-SRS	87.6	86.2	2.58	2.57	1.17

5.2 Ablation Study

In this section, we conduct ablation experiments on various constituent techniques of the causal system. As shown in Table 3, we evaluate the contributions of the components by taking the average of all test results. "- FN" means that we remove the feature normalization and use the batch normalization instead. "-BG" means to use random values to initialize the weight matrices in the bridge layer instead of the SRS coefficients. CRN-Time and CRN-SRS stands for single-branch on time and frequency domain respectively. For fair comparison, we adjusted all systems to have similar numbers of parameters.

We can find that all the variants underperforms the proposed dual-branch architecture, regardless of whether the feature normalization or bridge modules are replaced. Among them, feature normalization plays a more important role, which brings 1.0% STOI and 0.19 PESQ improvement under the babble condition. Note that replacing the bridge layer initialization method is not equivalent to deleting the bridge layer, but using a random initialization method. Under babble condition, STOI attenuates by 0.4% and PESQ attenuates by 0.12. Additionally, two independent single-branch variants were also evaluated. We can find that missing the reference information from the other branch leads to a sharp drop in performance. The independent time-domain branch and the frequency-domain branch reduce STOI by 3.4% and 3.7% under the babble, respectively. It further proves that the dual-branch network improves the enhancement performance by using alternate interconnection.

5.3 Comparison of Loss Function

To analyze the effectiveness of the loss function, we compare the models trained using different loss functions. First, we apply utterance-level MSE constraints to the time-branch outputs and magnitude-spectral constraints to the frequency-domain outputs, which are denoted as "L_{Mse}-L_{Mag}". Second, the magnitude spectral constraint is imposed on both outputs simultaneously, denoted as "L_{Mag}-L_{Mag}". These two combinations are used to prove that

Table 4a: Evaluations and comparisons of models using different loss function in terms of STOI(%). The best socre is marked in bold.

Metrix	STOI							
Test Noise	Babble				Cafeteria			
Test SNR (dB)	-5	0	5	Avg.	-5	0	5	Avg.
L_{Mse}-L_{Mag}	**85.3**	**92.7**	**95.9**	**91.3**	**82.0**	**91.2**	**95.2**	**89.5**
L_{Mag}-L_{Mag}	84.8	92.3	95.6	90.9	81.4	90.6	95.0	88.6
L_{Com}-L_{Com}	84.5	92.3	95.6	90.8	80.6	90.4	95.1	88.3
L_{Mse}-L_{Mag}-NC	**88.6**	94.1	**96.5**	**93.1**	**85.2**	**92.5**	**96.0**	**91.2**
L_{Mag}-L_{Mag}-NC	**88.6**	**94.2**	**96.5**	**93.1**	85.1	**92.5**	**96.0**	**91.2**
L_{Com}-L_{Com}-NC	88.3	94.1	**96.5**	93.0	84.6	92.3	96.0	91.0

Table 4b: Evaluations and comparisons of models using different loss function in terms of PESQ. The best socre is marked in bold.

Metrix	PESQ							
Test Noise	Babble				Cafeteria			
Test SNR (dB)	-5	0	5	Avg.	-5	0	5	Avg.
L_{Mse}-L_{Mag}	**2.45**	**2.98**	**3.33**	**2.92**	**2.29**	**2.84**	**3.24**	**2.79**
L_{Mag}-L_{Mag}	2.32	2.85	3.21	2.79	2.15	2.72	3.15	2.67
L_{Com}-L_{Com}	2.24	2.85	3.25	2.78	2.04	2.68	3.19	2.64
L_{Mse}-L_{Mag}-NC	2.63	3.12	3.38	3.05	**2.48**	3.01	3.35	2.95
L_{Mag}-L_{Mag}-NC	2.64	3.13	3.38	3.05	2.47	3.00	3.35	2.94
L_{Com}-L_{Com}-NC	**2.65**	**3.15**	**3.45**	**3.08**	2.45	**3.02**	**3.41**	**2.96**

the combination of different domain loss functions benefits the performance of the dual-branch network. However, the used L_{Mag} loss function brings unknown artifacts to the enhanced speech, which corrupts the subjective perception. Therefore, the complex loss, which constrains both real and imaginary components and magnitude, is used on both outputs to alleviate this problem. This is denoted as "L_{Com}-L_{Com}".

Table 4a and Table 4b gives the comparison results for the three combinations. For the causal system, the combination of L_{Mse}-L_{Mag} consistently outperforms the other loss functions in all conditions. It is the deviation of the two learning objectives that makes the features extracted by the two branches more differentiated, prompting each branch to focus on the noise it excels and reducing the learning difficulty of individual branches. Then, the transformation and fusion of the bridge layers allow the information to complement each other, giving a huge boost to the model performance. In contrast, in the non-causal system, the combination of L_{Mse} and L_{Mag} still maintains the advantage in STOI, but lags behind L_{Com} in PESQ, especially at high signal-to-noise ratios. It can be noticed that the frequency domain branch using L_{Mag} lacks the phase constraint compared to L_{Com}, which suggests that the extra phase constraint could bring improvement in speech quality when the contextual information is sufficient.

Table 4a and Table 4b shows the results of the three combinations, and it can be seen that the combination of constraints in different domains can bring performance improvements. In our records, however, the performance of the time branch drops considerably. We reckon that it is the deviation of the learning target of the two branches that makes the information from the other branch more instructive, resulting in better performance of the frequency domain branch. Additionally, we notice that L_{Mag} does not consistently improve PESQ at different SNRs. At low SNR, L_{Com} obtains better performances than the others. Consistent with the previously mentioned, we think this is because L_{Mag} introduces artifacts and PESQ is more sensitive to artifacts. Moreover, this problem is more pronounced in non-causal systems. According to the results, the combination of L_{Mse} and L_{Mag} may be the best choice.

5.4 Comparison of Model Complexities

The number of parameters and computational complexity are important indicators for evaluating a model and determining the scenarios in which the model can be applied. We evaluate the model complexity of the proposed model and other baseline systems using trainable parameters and memory access cost (MAC) for processing one second of speech. As shown in the Table 5, the dual-branch model has about 2.85M parameters and 6.12G MAC per second signal. Compared with the frequency domain model, the proposed model has a bit more computational cost and fewer trainable parameters, while the performance of the model is much better than the baseline in the frequency domain. Compared with the time-domain model, the proposed model is superior in terms of parameter quantity, computational complexity, and even performance. This undoubtedly proves the effectiveness of the dual branch network.

Table 5: Number of Trainable Parameters and MACS for Different Causal Systems, Where M Indicates Million.

Model	Parameters	MACs
CRN	17.6M	2.51G
GCRN	9.76M	2.37G
AECNN	6.44M	11.7G
DDAEC	4.80M	36.5G
DCN	4.63M	35.7G
CRN-Time	1.43M	3.06G
CRN-SRS	1.43M	3.06G
DBCN	2.85M	6.12G

6 Conclusion

In this study, we propose a novel dual-path convolutional network for cross-domain speech enhancement, where the time-domain branch pays more attention to local information, and the frequency-domain branch pays more attention to the relationship between frames. The bridge layer is introduced for the conversion and fusion of information between the two domains, thus the two branches can achieve the effect of mutual guidance. Furthermore, we propose a novel feature normalization method. The relative relationship between frames of each frequency point can be obtained by using statistical root mean square to regulate data. In the absence of bias, the model maintains complete noise reduction capability for input waveform with arbitrary amplitudes.

We have developed causal and non-causal DBCN, which are trained on the WSJ corpus and evaluated on untrained WSJ speakers. Systematic comparison experiments and ablation experiments of each module are conducted, which proved that DBCN outperforms existing noise and speaker-independent approaches for speech enhancement. In addition, the amount of parameters and computation of the proposed network is much smaller than other time-domain baselines, and it can denoise under fully causal conditions. However, due to equipment limitations, this algorithm cannot be applied to terminal equipment. In future studies, we will design more suitable structures for each domain and plan to further reduce the computational load of the model to better implement the algorithm in real time scenarios. Finally, we consider introducing a pre-training strategy to improve the practicality and performance of the model.

Acknowledgement

This research was supported by the China National Nature Science Foundation (No. 61876214). This work was also supported by The Project (KF-2022-07-009) Supported the Open Fund of Key Laboratory of Urban Land Resources Monitoring and Simulation, Ministry of Natural Resources, China.

References

[1] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," *stat*, 1050, 2016.

[2] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(2), 1979, 113–20.

[3] J. Chen, Y. Wang, S. E. Yoho, D. Wang, and E. W. Healy, "Large-scale training to increase speech intelligibility for hearing-impaired listeners in novel noises," *The Journal of the Acoustical Society of America*, 139(5), 2016, 2604–12.

[4] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*, 2017, 933–41.

[5] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(6), 1984, 1109–21.

[6] F. Gao, L. Wu, L. Zhao, T. Qin, X. Cheng, and T. Y. Liu, "Efficient sequence learning with group recurrent networks," in *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018, 799–808.

[7] M. Ge, C. Xu, L. Wang, E. S. Chng, J. Dang, and H. Li, "Spex+: A complete time domain speaker extraction network," in *Interspeech*, 2020, 1406–10.

[8] D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32(2), 1984, 236–43.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, 9(8), 1997, 1735–80.

[10] T. A. Hsieh, H. M. Wang, X. Lu, and Y. Tsao, "Wavecrn: An efficient convolutional recurrent neural network for end-to-end speech enhancement," *IEEE Signal Processing Letters*, 27, 2020, 2149–53.

[11] Y. Hu, Y. Liu, S. Lv, M. Xing, S. Zhang, Y. Fu, and L. Xie, "DCCRN: Deep complex convolution recurrent network for phase-aware speech enhancement," in *Interspeech*, 2020, 2472–6.

[12] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, 448–56.

[13] Y. Ju, W. Rao, X. Yan, Y. Fu, S. Lv, L. Cheng, and S. Shang, "TEA-PSE: Tencent-ethereal-audio-lab personalized speech enhancement system for ICASSP 2022 DNS CHALLENGE," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, 9291–5.

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

[15] A. Li, W. Liu, C. Zheng, C. Fan, and X. Li, "Two heads are better than one: A two-stage complex spectral mapping approach for monaural speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29, 2021, 1829–43.

[16] A. Li, C. Zheng, G. Yu, J. Cai, and X. Li, "Filtering and refining: A collaborative-style framework for single-channel speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30, 2022, 2156–72.

[17] J. Li, H. Zhang, X. Zhang, and C. Li, "DCCRN: Single channel speech enhancement using temporal convolutional recurrent neural networks," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 2019, 896–900.

[18] J. Liu and X. Zhang, "Inplace gated convolutional recurrent neural network for dual-channel speech enhancement," in *Interspeech*, 2021, 1852–6.

[19] Y. Liu, H. Zhang, and X. Zhang, "Using shifted real spectrum mask as training target for supervised speech separation," in *Interspeech*, 2018, 1151–5.

[20] P. C. Loizou, *Speech Enhancement: Theory and Practice*, Boca Raton: CRC Press, 2007.

[21] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder," in *Interspeech*, 2013, 436–40.

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

[23] 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*, 27(8), 2019, 1256–66.

[24] Y. Luo and N. Mesgarani, "Tasnet: time-domain audio separation network for real-time, single-channel speech separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, 696–700.

[25] S. Lv, Y. Hu, S. Zhang, and L. Xie, "DCCRN+: Channel-wise subband DCCRN with SNR estimation for speech enhancement," 2021, arXiv:2106.08672.

[26] C. Macartney and T. Weyde, "Improved speech enhancement with the wave-u-net," 2018, arXiv:1811.11307.

[27] K. Paliwal, K. Wójcicki, and B. Shannon, "The importance of phase in speech enhancement," *Speech Communication*, 53(4), 2011, 465–94.

[28] A. Pandey and D. Wang, "A new framework for CNN-based speech enhancement in the time domain," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(7), 2019, 1179–88.

[29] A. Pandey and D. Wang, "Dense CNN with self-attention for time-domain speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29, 2021, 1270–9.

[30] A. Pandey and D. Wang, "Densely connected neural network with dilated convolutions for real-time speech enhancement in the time domain," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, 6629–33.

[31] A. Pandey and D. Wang, "Learning complex spectral mapping for speech enhancement with improved cross-corpus generalization," in *Interspeech*, 2020, 4511–5.

[32] A. Pandey and D. Wang, "On adversarial training and loss functions for speech enhancement," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, 5414–8.

[33] A. Pandey and D. Wang, "Self-attending RNN for speech enhancement to improve cross-corpus generalization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30, 2022, 1374–85.

[34] A. Pandey and D. Wang, "TCNN: Temporal convolutional neural network for real-time speech enhancement in the time domain," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, 6875–9.

[35] D. B. Paul and J. Baker, "The design for the Wall Street Journal-based CSR corpus," in *Proceedings of the Workshop on Speech and Natural Language*, 1992, 357–62.

[36] A. W. Rix, M. P. Hollier, A. P. Hekstra, and J. G. Beerends, "Perceptual Evaluation of Speech Quality (PESQ) The New ITU Standard for End-to-End Speech Quality Assessment Part I–Time-Delay Compensation," *Journal of the Audio Engineering Society*, 50(10), 2002, 755–64.

[37] P. Scalart, "Speech enhancement based on a priori signal to noise estimation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, 1996, 629–32.

[38] M. H. Soni, R. Tak, and H. A. Patil, "Novel Shifted Real Spectrum for Exact Signal Reconstruction," in *Interspeech*, 2017, 3112–6.

[39] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 19(7), 2011, 2125–36.

[40] K. Tan and D. Wang, "A convolutional recurrent neural network for real-time speech enhancement," in *Interspeech*, 2018, 3229–33.

[41] K. Tan and D. Wang, "Learning complex spectral mapping with gated convolutional recurrent networks for monaural speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28, 2019, 380–90.

[42] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," 2017, arXiv:1607.08022.

[43] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, and A. Graves, "Conditional image generation with pixelcnn decoders," *Advances in neural information processing systems*, 29(8), 1151–5.

[44] D. Wang, *On ideal binary mask as the computational goal of auditory scene analysis, in Speech separation by humans and machines*, Boston, MA: Springer, 2005.

[45] H. Wang and D. Wang, "Neural cascade architecture with triple-domain loss for speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30, 2021, 734–43.

[46] T. Wang, W. Zhu, Y. Gao, Y. Chen, J. Feng, and S. Zhang, "Harmonic gated compensation network plus for ICASSP 2022 DNS challenge," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, 9286–90.

[47] Y. Wang, A. Narayanan, and D. Wang, "On training Targets for supervised speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(12), 2014, 1849–58.

[48] Y. Wang and D. Wang, "Towards Scaling Up Classification-Based Speech Separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 21(7), 2013, 1381–90.

[49] Z. Q. Wang, P. Wang, and D. Wang, "Complex spectral mapping for single-and multi-channel speech enhancement and robust ASR," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28, 2020, 1778–87.

[50] D. S. Williamson, Y. Wang, and D. Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(3), 2015, 483–92.

[51] C. Xu, W. Rao, E. S. Chng, and H. Li, "Spex: Multi-scale time domain speaker extraction network," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28, 2020, 1370–84.

[52] G. Zhang, L. Yu, C. Wang, and J. Wei, "Multi-scale temporal frequency convolutional network with axial attention for speech enhancement," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, 9122–6.

[53] K. Zhang, S. He, H. Li, and X. Zhang, "DBCN: A dual-branch network architecture processing on spectrum and waveform for single-channel speech enhancement," in *Interspeech*, 2021, 2821–5.

[54] Z. Zhang, L. Zhang, X. Zhuang, Y. Qian, H. Li, and M. Wang, "FB-MSTCN: A full-band single-channel speech enhancement method based on multi-scale temporal convolutional network," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, 9276–80.

[55] S. Zhao, B. Ma, K. N. Watcharasupat, and W. S. Gan, "FRCRN: Boosting feature representation using frequency recurrence for monaural speech enhancement," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, 9281–5.