DCCRN+: Channel-wise Subband DCCRN with SNR Estimation for Speech Enhancement

Shubo Lv*, *Yanxin Hu**, *Shimin Zhang, Lei Xie* †

Audio, Speech and Language Processing Group (ASLP@NPU), School of Computer Science, Northwestern Polytechnical University, Xi'an, China

{shblv, yxhu, shmzhang}@npu-aslp.org, lxie@nwpu.edu.cn

Abstract

Deep complex convolution recurrent network (DCCRN), which extends CRN with complex structure, has achieved superior performance in MOS evaluation in Interspeech 2020 deep noise suppression challenge (DNS2020). This paper further extends DCCRN with the following significant revisions. We first extend the model to sub-band processing where the bands are split and merged by learnable neural network filters instead of engineered FIR filters, leading to a faster noise suppressor trained in an end-to-end manner. Then the LSTM is further substituted with a complex TF-LSTM to better model temporal dependencies along both time and frequency axes. Moreover, instead of simply concatenating the output of each encoder layer to the input of the corresponding decoder layer, we use convolution blocks to first aggregate essential information from the encoder output before feeding it to the decoder layers. We specifically formulate the decoder with an extra *a priori* SNR estimation module to maintain good speech quality while removing noise. Finally a post-processing module is adopted to further suppress the unnatural residual noise. The new model, named DCCRN+, has surpassed the original DCCRN as well as several competitive models in terms of PESQ and DNSMOS, and has achieved superior performance in the new Interspeech 2021 DNS challenge.

Index Terms: speech enhancement, sub-band processing, deep complex convolution recurrent network

1. Introduction

Speech enhancement refers to the task of eliminating background noise and improving speech quality and intelligibility from noisy audio signals. A decent speech enhancer has significant applications aiming to improve human or machine interpretation of speech, including hearing aids, audio communication and automatic speech recognition. Traditional speech enhancement methods usually apply a spectral suppression gain (or filter) to the noisy signal under the statistical signal processing theory. With the help of deep learning (DL), speech enhancement has been formulated as a supervised learning problem. Such data-driven approaches have become the mainstream because of their strong noise reduction abilities (especially for non-stationary noise) learned from simulated clean-noisy speech pairs. The recent deep noise suppression challenge (DNS) series[1, 2] have benchmarked many state-of-the-art DL-based speech enhancers, especially for real-time ones for speech communications, through subjective listening test and promising performance has been reported.

A key challenge in developing a speech enhancer for human speech communication (and even for machine speech recognition) is how to preserve perceived subjective speech quality to the best extent while greatly suppressing the noise interference. Some recent studies, including the new DNS challenge initiative [2], have pointed out that many neural noise suppressors are very good at suppressing noise, but do not improve the quality of speech, even introducing apparent speech distortions. In this paper, we study the problem by revising our recently proposed deep complex convolution recurrent network (DCCRN) [3].

Mapping and masking are two commonly used DL-based strategies for speech enhancement. The mask-based approaches have gradually become the mainstream as they constraint dynamic range and usually converge faster. Various masks have been explored, including ideal binary mask (IBM) [4], ideal ratio mask (IRM) [5] and spectral magnitude mask (SMM) [6]. Most of these approaches ignore the phase information as it is difficult to model due to its unclear structure. But recent speech enhancement studies have shown the clear benefits of modeling phase through phase-sensitive mask (PSM) [7] and complex ratio mask (CRM) [8]. In the CRM-based approach, the mask is applied to both real and imaginary (RI) components, and both magnitude and phase can be reasonably estimated, leading to improved enhancement performance. Time-domain approaches have also shown impressive noise suppression ability, which bypass explicit phase modeling by direct waveform input and output [9].

Meanwhile, the speech enhancement performance has been boosted with the well-designed network structures [10, 11]. Recently, these network structures have gradually taken into account the phase information in the Time-Frequency (T-F) domain [12]. Convolution recurrent network (CRN) [13] is one of the most popular convolution encoder-decoder (CED) structures for speech enhancement. Originally it took a real spectrum as input and only estimated real mask through network. Later Tan *et al.* proposed an updated structure of one encoder and two decoders for complex spectrum mapping (CSM) [14] to estimate the real and imaginary parts of the spectrum, leading to improved performance. Very recently, we further updated the CRN network by introducing complex convolution and LSTM, resulting in the deep complex convolution recurrent network (DCCRN) [3]. The DCCRN models are very competitive over other networks, either on objective or subjective metric. With 3.7M parameters, the model has ranked first in MOS evaluation in the Interspeech2020 DNS challenge real-time-track.

In this paper, we further update DCCRN with the following important aspects, resulting in DCCRN+, aiming to improve perceived speech quality while greatly reducing the noise interference. 1) We enable the model with subband processing ability by learnable neural network filters. With smaller model size and speed-up inference, this update leads to 0.17 PESQ improvement as compared with the subband counterpart based on

---

* equal contribution
† corresponding author

engineered FIR-filters, while maintaining the PESQ at the same level with DCCRN. 2) We update the network structure with convolution based connections between encoder and decoder layers for better embedding and complex TF-LSTM for temporal dependency modeling. This update brings an extra PESQ gain of 0.05. 3) To maintain good speech quality, we formulate the decoder as a multi-task learning framework with an auxiliary task of *a priori* SNR estimation which has proven to be beneficial to the perceived speech quality [15]. This benefit is confirmed with PESQ gain of 0.03 according to our experiments. 4) Finally the new DCCRN+ model (with MMSE-LSA based post-processing for residual noise removal) has surpassed several state-of-the-art models and obtained superior performance with 3.51 MOS in the DNS2021 challenge.

2. DCCRN+

2.1. The new design

DCCRN [3] is the complex network version of CRN with a symmetrically-designed encoder-decoder structure and long short-term memory (LSTM) layers. The encoder and decoder are composed of complex convolution/deconvolution layers and LSTM or complex LSTM is inserted in between, which intends to model temporal dependencies in speech. The complex module models the correlation between magnitude and phase with the simulation of complex multiplication. Skip connections are used to concatenate the output of each encoder layer to the input of the corresponding decoder layer. With this model structure, DCCRN has achieved the best MOS in the subjective listening test in Interspeech 2020 deep noise suppression challenge.

We substantially extend DCCRN with several important revisions. The model structure of the so-called DCCRN+ is shown in Figure 1(a). We can see that the general structure is similar to DCCRN but with the following differences: 1) Subband processing with split/merge modules before/after encoder/decoder. 2) Complex TF-LSTM for both frequency and time scale temporal modeling. 3) Addition of convolution for better information aggregation from encoder output before feeding to decoder. 4) Addition of SNR estimation module to alleviate speech distortion during noise suppression. 5) Post-processing to further remove residual noise.

2.2. Subband Processing

Subband processing is a common approach in audio processing [16, 17, 18] to reduce model size and save computation cost. The audio in each frequency subband can be down-sampled by a factor of K (the number of frequency bands), and thus the total computational cost can be reduced. Previous work [19] also pointed out that local patterns in the spectrogram are often different in each frequency band: the lower frequency band tends to contain high energies, tonalities as well as long sustained sounds, while the higher frequency band is likely to have low energy components, noise and rapidly decaying sounds. Subbands are often split by engineered Finite Impulse Response (FIR) filters [20] and it is difficult to design perfect filters for different applications. Recent attempts on learnable frontend using neural networks [21] have shown promising results, which may avoid the troublesome manual design of specific filters. This inspires us to develop learnable subband split and merge modules automatically using neural layers.

The input of the band-splitting module is the T-F spectrum resulted from short time Fourier transform (STFT), denoted as $Y \in R^{T \times F}$, where T is the number of frames

and F is the number of frequency bins. We employ neural network based analysis filters $A_k(f_k)$ for band splitting, where $k \in 1, \cdots, K$ stands for the number of subbands and $f_k \in (F/K) \cdot (k-1), \cdots, (F/K) \cdot k$. The output after split analysis for band k is

$$Y_k(f,t) = \sum_{f_k=(F/K)\cdot(k-1)}^{(F/K)\cdot k} Y(f,t) \cdot A_k(f_k). \quad (1)$$

Then Y_k is processed by Instance Normalization to accelerate the model convergence. and maintain the independence between each band. Y_k for band k will go through the encoder-decoder structure, as shown in Fig. 1(a). Deepfilter [22] is adopted to process estimated mask and input subbands. Finally they are merged back using

$$\hat{X}(f,t) = \sum_f \text{CAT}(\hat{X}_1, \cdots, \hat{X}_k, \cdots, \hat{X}_K) \cdot S(f) \quad (2)$$

where $\hat{X}_k = \hat{X}_k(f_k,t)$ denotes the network output of k subband. After concatenating them together, we use another neural network based synthesis filters $S(f)$ to obtain the (merged) output spectrum $\hat{X}(f,t)$. The number of bands K can be pratically set, and $A_k(f_k)$ and $S(f)$ are learnable neural network layers.

2.3. Complex TF-LSTM Block

DCCRN employs LSTM layers to process the encoder output for temporal dependency modeling at time scale. Inspired by [23], we explicitly model the frequency-wise evolution of spectral patterns as well by another LSTM module Fig. 1(b). Specifically, we first use a complex frequency-LSTM (F-LSTM) to browse the frequency bands so that frequency-evolving information is summarized, and then the output layer activations are used as the input to the complex time-LSTM (T-LSTM) for time scale summarization. In details, the complex F-LSTM can be described as

$$U_f = [\text{CAT}(\text{BLSTM}_r(\Re(E)[i,:,:]),$$
$$\text{BLSTM}(\Im(E)[i,:,:])), i = 1 \cdots F] \quad (3)$$

where $E \in R^{F \times T \times C}$ denotes the encoder output. Then we send the real($\Re$)/imag($\Im$) parts of E to real/imag BLSTMs separately. After concatenating (CAT in Eq (3)) them together, the output U_f is processed by Complex Linear Projection (CLP) [24]) without modulo operation:

$$O_f = \text{CLP}(U_f). \quad (4)$$

Subsequently we take the outputs from the complex F-LSTM as the input to feed to the complex T-LSTM to do time scale analysis:

$$O_t = \text{CLP}([\text{CAT}(\text{LSTM}_r(\Re(O_f)[:,i,:]),$$
$$\text{LSTM}_i(\Im(O_f)[:,i,:])), i = 1 \cdots T]) \quad (5)$$

where O_t is the final output of the complex TF-LSTM block which is subsequently sent to the decoder. Note that we use unidirectional LSTM at time scale as real-time speech enhancement is time sensitive.

2.4. Convolution Pathway

In the original DCCRN, through skip-connections, the feature maps of the encoder are directly concatenated to the decoder. In DCCRN+, before feeding to the decoder layers, the encoder output in each layer undergoes a convolution-based information aggregation block. Specifically, the skip pathway between encoder and decoder consists of a complex convolution block and batch normalization.

2.5. SNR Estimator

Our aim is to maintain good speech quality while reducing noise. Previous experiences [25] show that directly training the
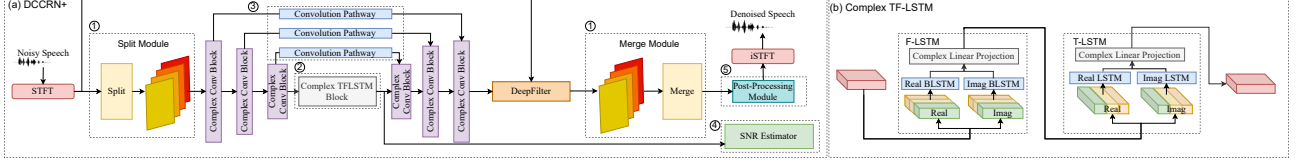
Figure 1: *Illustration of (a) Network structure of the proposed DCCRN+ and (b) Complex TF-LSTM Block. Main contributions include 1) subband processing, 2) complex TF-LSTM, 3) convolution pathway, 4) SNR estimation module and 5) post-processing.*

neural noise suppression module may inevitably lead to a certain amount of speech distortion. We migrate this problem by using an SNR estimator to estimate frame-level SNR under a multi-task learning framework. The input of SNR estimator is Complex TF-LSTM, then it is fed to one LSTM layer and a Conv1D layer with sigmoid to estimate frame-level SNR. This module is used as an auxiliary task in training and removed at inference stage. The SNR target label is given as follows:

$$\xi(t) = 20log_{10}(X(t)/N(t)) \tag{6}$$

where t is the frame index, $\xi(t)$ is the *priori* frame-level SNR, $X(t)$ is the clean speech spectrum and $N(t)$ is the noise spectrum. To avoid the influence of the fluctuation of SNR estimation, a simple processing is done on the *priori* frame-level SNR during training.

$$\begin{aligned} \hat{\mu} &= \hat{\mu} \cdot \alpha + \mu \cdot (1 - \alpha) \\ \hat{\sigma} &= \hat{\sigma} \cdot \alpha + \sigma \cdot (1 - \alpha) \end{aligned} \tag{7}$$

where $\hat{\mu}$ and $\hat{\sigma}$ denote the moving average of the SNR mean and std respectively. The current SNR mean and std of a training utterance is denoted as μ, σ. Then the SNR is normalized.

$$\xi(t) = (\xi(t) - \hat{\mu})/\hat{\sigma} \tag{8}$$

After that, we use ERF (Error Function) and compress SNR to 0 ~1 to avoid the too large range which may lead to poor network convergence [15]. In this paper, we empirically set α to 0.99.

$$\xi(t) = (\text{ERF}(\xi(t)) + 1)/2 \tag{9}$$

2.6. Post-Processing

With the SNR estimator, despite the notable improvement of speech quality, there may exist some residual noise. In order to remove such residual noise, we adopt a post-processing module. The estimated spectrum and noisy spectrum are used to calculate *priori* and *posterior* SNR as follows:

$$\begin{aligned} \xi'(t) &= \text{CDF}(\sigma_{\hat{x}}^2(t))/\text{CDF}(\sigma_n^2(t)) \\ \gamma(t) &= \text{CDF}(\sigma_y^2(t))/\text{CDF}(\sigma_n^2(t)) \end{aligned} \tag{10}$$

where $\xi'(t), \gamma(t)$ are the *priori* and *posterior* frame-level SNR, CDF is the cumulative distribution function, $\sigma_{\hat{X}}^2(t)$ is the variance of estimated spectrum, and $\sigma_N^2(t)$ and $\sigma_Y^2(t)$ are the variance of noise spectrum and noisy spectrum respectively. With the estimated SNR, an MMSE-LSA estimator [26] is introduced to compute the final frame-level gain G and then applied to estimated spectrum $\hat{X}$ to suppress the residual noise.

$$\begin{aligned} \text{G} &= \text{MMSE\_LSA}(\xi', \gamma) \\ \hat{X} &= \text{G} \cdot \hat{X} \end{aligned} \tag{11}$$

In practice, the noise suppression ability of the post-processing module gradually deteriorates because of cumulative variance. Thus we use the rate-of-change of SNR to reset the variance:

$$r(t) = (\xi'(t) - \xi'(t-1))/\xi'(t-1) \tag{12}$$

where $r(t)$ is the rate-of-change of SNR. Once $r(t)$ is larger than 1, the variance is reset. In this way, we can observe that the speech is undamaged and residual noise is largely removed.

2.7. Loss Function

The SI-SNR loss [9] is used in noise suppression. In addition, we also use MSE loss to guide the learning of SNR estimator.

$$L_{\text{SNR}} = \text{MSE}(\hat{\xi}, \xi) \tag{13}$$

where $\hat{\xi}$ is the output of the SNR estimation module and ξ is the label. The final loss is

$$L = L_{\text{SI-SNR}} + \delta \cdot L_{\text{SNR}}. \tag{14}$$

In this paper, as the values of the two losses are not in the same scale, we empirically set δ to 30.

3. Experiments

3.1. Datasets

We first take a comprehensive ablation study on the proposed model on the DNS-2020 dataset [1]. Then our model is trained, integrated with the post-processing module and evaluated with the Interspeech 2021 DNS challenge dataset (DNS-2021) [2] to show its performance on more complicated and real acoustic scenarios. We also compare other competitive models (such as PercepNet [27]) with our model on Voice Bank + DEMAND dataset [28] as these models have PESQ scores reported on this set. Specifically, DNS-2020 speech set includes a total of 500 hours of clean speech from 2,150 speakers. The 180-hour DNS-2020 noise set includes 65,000 noise clips from 150 noise classes. And there are 80,000 RIR clips, coming from the RIR_NOISES set [29] and the provided RIR set by the challenge. For DNS-2021 dataset, we drop the utterances with poor quality. Totally, we generate a large clean dataset with 760 hours duration. The training set and validation set are configured according to a ratio of 9:1 after shuffling. In the training stage, we perform dynamic mixing of speech and noise, while the SNR ranges from -5db to 20db, and there is 50% probability of convolving with RIR for simulating speech with reverberation. We also apply a biquad filter [30] on half of the training data. As for the Voice Bank + DEMAND dataset, there are 824 samples from 8 speakers for testing.

3.2. Training setup and baselines

For all of our models, the window length and frame shift are 20ms and 10ms respectively and the FFT length is 512. We use Adam optimizer and the initial learning rate is 1e-3. When the loss of the validation set increases, the learning rate will be decayed by a ratio of 0.5.

DCCRN: The number of channels for the DCCRN is {16,32,64,128,256,256}, and the convolution kernel and step size are set to (5,2) and (2,1). Two LSTM layers are adopted and the number of nodes is 256. There is a 1024*256 fully connected layer after the LSTM. Each encoder module handles the current frame and one previous frame. In the decoder, the last layer processes one extra future frame, and each previous layer utilizes the current frame and one historical frame.

DCCRN+: The number of channels for DCCRN+ is {32,64,128,256}. The split-band module is a group Conv1D layer with 4 groups. Correspondingly, the merge-module is a linear layer. The Complex TF-LSTM module is composed of one complex LSTM (the units of real and imag parts are 256

Table 1: *Various models' PESQ on DNS-2020 synthetic test set.*

Model	Para.(M)	look-ahead (ms)	RTF	no reverb	reverb	Ave.
Noisy	-	0	-	2.45	2.75	2.60
NSNet [32] (Baseline)	1.3	10	-	3.07	2.81	2.94
DCCRN	3.7	10	0.272	3.26	3.20	3.23
Subband DCCRN (FIR filter)	2.8	10	0.177	3.03	3.11	3.07
Subband DCCRN (NN filter)	2.8	10	**0.137**	3.25	3.22	3.24
+Complex TF-LSTM	3.3	10	0.244	3.29	3.27	3.28
+ Convolution Pathway	3.3	10	0.248	3.30	3.28	3.29
+ SNR Estimator	3.3	10	0.248	**3.33**	**3.30**	**3.32**
+ Post-processing	3.3	10	0.250	3.27	3.23	3.25

Table 2: *PESQ on Voice Bank + DEMAND*

Model	Para.(M)	Semi-Causal	External Data	PESQ-WB
Noisy	-	-	-	1.97
RNNoise [30]	0.06	√	√	2.29
PercepNet [27]	8	√	√	2.73
DCCRN	3.7	√	√	2.68
DCCRN+	3.3	√	√	**2.84**

separately) and one complex BLSTM (real and imag parts are 256 for each direction separately). The CLP module has 256 units for real and imag parts. The convolution pathway module is composed of a 1×1 complex Conv2D layer. The SNR Estimator is a 64 units LSTM layer followed by a Conv1D layer with 3 kernels. The rest configuration is the same as DCCRN.

Since only 10ms future speech is seen in the last layer of the decoder, the total processing time is $20 + 10 + 10 = 40ms$, which meets the challenge requirement: frame length + frame shift + future frame ≤ 40ms.

3.3. Experimental results and discussion

PESQ [31] is first reported for various models on the DNS-2020 synthetic test set, as shown in Table 1. Real Time Factor (RTF) is also tested on a machine with an Intel(R) Xeon(R) CPU E5-2640@2.50GHz in single thread. Note that NSNet is the baseline model provided by the challenge organizer.

The result in Table 1 shows that the subband operation can considerably improve speed and reduce model size. But the PESQ for the FIR-filter-based subband DCCRN model has a clear degradation with the orginal DCCRN. With the help of the proposed neural network filter, the PESQ for the subband model is restored to the same level with DCCRN and the inference speed is further boosted with RTF of 0.137. After substituting LSTM with complex TF-LSTM in subband DCCRN (NN filter), we obtain a noticeable PESQ improvement, while the model becomes larger and slower. The use of convolution pathway and SNR estimator brings further PESQ gain and the best PESQ is 3.32, which is a clear improvement as compared with original DCCRN. Interestingly, when comparing the feature map before and after using the convolution pathway (CP), we find that the use of CP results in more clear patterns on the feature map, leading to better noise reduction. A case study on one testing clip is shown in Fig.2, where more noise is suppressed when CP is applied. Finally, when post-processing is applied, PESQ is decreased due to some spectrum information lost [33]. However the use of post-processing is beneficial to subjective listening as shown in previous works [27, 33] because unnatural residual noise is further suppressed.

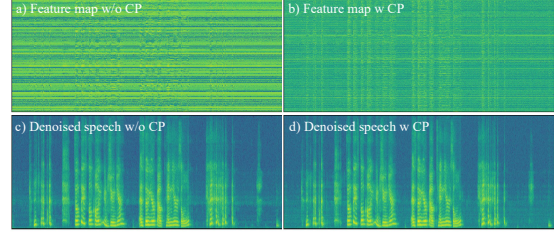We compare the PESQ performance of DCCRN+ with

Figure 2: *Comparison on the denoising result on a testing noisy clip for the cases with and without convolution pathway (CP).*

Table 3: *DNSMOS on DNS-2021 blind test set.*

Model	DNSMOS
Noisy	2.94
DCCRN	3.40
Subband DCCRN (FIR filter)	3.16
Subband DCCRN (NN filter)	3.37
+ complex TF-LSTM	3.40
+ CP	3.42
+ SNR Estimator	3.43
+ post-processing	**3.46**

other competitive models on the Voice Bank + DEMAND test set. The results in Table 2 show that the proposed DCCRN+ clearly outperforms other models, DCCRN+ surpasses PercepNet by a large margin with even fewer parameters.

We further test our models trained on the DNS-2021 dataset. This time we use DNSMOS [34] for evaluation – the new metric provided by the challenge organizer that is believed to be more correlated with subjective listening score. As can be seen from Table 3 that DNSMOS is increased with the addition of the updates on our model and the highest score 3.46 is achieved by the use of all updates, including the post-processor. We submitted the enhanced clips on the blind test set using the the final model. Table 4 shows the P.835 [35] DNS-2021 subjective evaluation results released by the challenge organizer. We can observe that our DCCRN+ model significantly outperforms NSNet2 baseline with large MOS improvement. Our system ranked top 4 in overall MOS in all 20 submissions in the wideband track in the challenge.

Table 4: *DMOS on DNS-2021 challenge blind test set*

Model	Stat.	Emot.	Non-En-T	Non-En	Mus	En.	Overall
Noisy	0(3.03)	0(2.28)	0(3)	0(3.04)	0(2.57)	0(2.52)	0(2.77)
NSNet2 [32]	0.25	0.47	0.31	0.21	0.21	0.41	0.30
DCCRN+	**0.69**	**1.13**	**0.59**	**0.64**	**0.76**	**0.75**	**0.74**

4. Conclusions

This paper presents substantial updates on our previously proposed DCCRN model for speech enhancement. The new model, called DCCRN+, is equipped with subband processing ability via learnable neural filters for band split and merge, leading to compact model size and speed-up inference. The new model has also been updated with TF-LSTM and convolution pathway. Importantly, an SNR estimator is adopted along with the decoder, under the multi-task learning framework, to maintain good speech quality while removing noise. Finally a post-processor is adopted to remove unnatural residual noise. Experiments have shown the effectiveness of these updates. The DCCRN+ model has achieved superior performance in the subjective listening test in the Interspeech2021 DNS challenge. Some of the enhanced audio clips on the blind test set can be found from `https://imybo.github.io/dccrn-plus/`.

5. References

[1] C. K. Reddy, V. Gopal, R. Cutler, E. Beyrami, R. Cheng, H. Dubey, S. Matusevych, R. Aichner, A. Aazami, S. Braun, P. Rana, S. Srinivasan, and J. Gehrke, "The INTERSPEECH 2020 Deep Noise Suppression Challenge: Datasets, Subjective Testing Framework, and Challenge Results," in *Proc. Interspeech*, 2020, pp. 2492–2496.

[2] C. K. Reddy, H. Dubey, K. Koishida, A. Nair, V. Gopal, R. Cutler, S. Braun, H. Gamper, R. Aichner, and S. Srinivasan, "Interspeech 2021 deep noise suppression challenge," *ArXiv*, vol. abs/2101.01902, 2021.

[3] Y. Hu, Y. Liu, S. Lv, M. Xing, S. Zhang, Y. Fu, J. Wu, B. Zhang, and L. Xie, "Dccrn: Deep complex convolution recurrent network for phase-aware speech enhancement," in *Proc. Interspeech*, 2020, pp. 2472–2476.

[4] D. Wang, "On ideal binary mask as the computational goal of auditory scene analysis," in *Speech separation by humans and machines*, 2005, pp. 181–197.

[5] A. Narayanan and D. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *Proc. ICASSP*. IEEE, 2013, pp. 7092–7096.

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

[7] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *Proc. ICASSP*. IEEE, 2015, pp. 708–712.

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

[9] 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.

[10] Y. Wang and D. Wang, "A deep neural network for time-domain signal reconstruction," in *Proc. ICASSP*. IEEE, 2015, pp. 4390–4394.

[11] S. Pascual, A. Bonafonte, and J. Serrà, "SEGAN: Speech Enhancement Generative Adversarial Network," in *Proc. Interspeech*, 2017, pp. 3642–3646.

[12] H.-S. Choi, J.-H. Kim, J. Huh, A. Kim, J.-W. Ha, and K. Lee, "Phase-aware speech enhancement with deep complex u-net," in *Proc. ICLR*, 2018.

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

[14] K. Tan and D. Wang, "Complex spectral mapping with a convolutional recurrent network for monaural speech enhancement," in *Proc. ICASSP*. IEEE, 2019, pp. 6865–6869.

[15] A. Nicolson and K. K. Paliwal, "Masked multi-head self-attention for causal speech enhancement," *Speech Communication*, vol. 125, pp. 80–96, 2020.

[16] X. Li and R. Horaud, "Online Monaural Speech Enhancement Using Delayed Subband LSTM," in *Proc. Interspeech*.

[17] H. Liu, L. Xie, J. Wu, and G. Yang, "Channel-wise subband input for better voice and accompaniment separation on high resolution music," in *Proc. Interspeech*, 2020, pp. 1241–1245.

[18] G. Yang, S. Yang, K. Liu, P. Fang, W. Chen, and L. Xie, "Multi-band MelGAN: Faster waveform generation for high-quality text-to-speech," in *Proc. SLT*. IEEE, 2021, pp. 492–498.

[19] N. Takahashi and Y. Mitsufuji, "Multi-scale multi-band densenets for audio source separation," in *Proc. ASPAA*. IEEE, 2017, pp. 21–25.

[20] C. Yu, H. Lu, N. Hu, M. Yu, C. Weng, K. Xu, P. Liu, D. Tuo, S. Kang, G. Lei, D. Su, and D. Yu, "DurIAN: Duration Informed Attention Network For Multimodal Synthesis," *ArXiv*, vol. abs/1909.01700, 2019.

[21] N. Zeghidour, O. Teboul, F. d. C. Quitry, and M. Tagliasacchi, "LEAF: A Learnable Frontend for Audio Classification," *ArXiv*, vol. abs/2101.08596, 2021.

[22] W. Mack and E. A. Habets, "Deep filtering: Signal extraction and reconstruction using complex time-frequency filters," *IEEE Signal Processing Letters*, vol. 27, pp. 61–65, 2019.

[23] J. Li, A. Mohamed, G. Zweig, and Y. Gong, "LSTM time and frequency recurrence for automatic speech recognition," in *Proc. ASRU*. IEEE, 2015, pp. 187–191.

[24] E. Variani, T. N. Sainath, I. Shafran, and M. Bacchiani, "Complex Linear Projection (CLP): A Discriminative Approach to Joint Feature Extraction and Acoustic Modeling," in *Proc. Interspeech*, 2016, pp. 808–812.

[25] C. Zheng, X. Peng, Y. Zhang, S. Srinivasan, and Y. Lu, "Interactive speech and noise modeling for speech enhancement," in *AAAI*, 2021.

[26] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE transactions on acoustics, speech, and signal processing*, vol. 33, no. 2, pp. 443–445, 1985.

[27] J.-M. Valin, U. Isik, N. Phansalkar, R. Giri, K. Helwani, and A. Krishnaswamy, "A Perceptually-Motivated Approach for Low-Complexity, Real-Time Enhancement of Fullband Speech," in *Proc. Interspeech*, 2020, pp. 2482–2486.

[28] C. Valentini-Botinhao, X. Wang, S. Takaki, and J. Yamagishi, "Investigating RNN-based speech enhancement methods for noise-robust Text-to-Speech." in *Proc. SSW*, 2016, pp. 146–152.

[29] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proc. ICASSP*. IEEE, 2017, pp. 5220–5224.

[30] J.-M. Valin, "A hybrid DSP/deep learning approach to real-time full-band speech enhancement," in *Proc. MMSP*. IEEE, 2018, pp. 1–5.

[31] I. Rec, "P. 862.2: Wideband extension to recommendation p. 862 for the assessment of wideband telephone networks and speech codecs," *International Telecommunication Union, CH–Geneva*, 2005.

[32] Y. Xia, S. Braun, C. K. Reddy, H. Dubey, R. Cutler, and I. Tashev, "Weighted speech distortion losses for neural-network-based real-time speech enhancement," in *Proc. ICASSP*. IEEE, 2020, pp. 871–875.

[33] A. Li, W. Liu, X. Luo, C. Zheng, and X. Li, "ICASSP 2021 deep noise suppression challenge: Decoupling magnitude and phase optimization with a two-stage deep network," in *Proc. ICASSP*. IEEE, 2021, pp. 6628–6632.

[34] C. K. Reddy, V. Gopal, and R. Cutler, "Dnsmos: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors," in *Proc. ICASSP*. IEEE, 2021, pp. 6493–6497.

[35] B. Naderi and R. Cutler, "A crowdsourcing extension of the itu-t recommendation p. 835 with validation," *ArXiv*, vol. abs/2010.13200, 2020.