COMPLEX SPECTRAL MAPPING WITH A CONVOLUTIONAL RECURRENT NETWORK FOR MONAURAL SPEECH ENHANCEMENT

Ke Tan[1] *and DeLiang Wang*[1,2]

[1]Department of Computer Science and Engineering, The Ohio State University, USA
[2]Center for Cognitive and Brain Sciences, The Ohio State University, USA
{tan.650, wang.77}@osu.edu

ABSTRACT

Phase is important for perceptual quality in speech enhancement. However, it seems intractable to directly estimate phase spectrogram through supervised learning due to lack of clear structure in phase spectrogram. Complex spectral mapping aims to estimate the real and imaginary spectrograms of clean speech from those of noisy speech, which simultaneously enhances magnitude and phase responses of noisy speech. In this paper, we propose a new convolutional recurrent network (CRN) for complex spectral mapping, which leads to a causal system for noise- and speaker-independent speech enhancement. In terms of objective intelligibility and perceptual quality, the proposed CRN significantly outperforms an existing convolutional neural network (CNN) for complex spectral mapping, as well as a strong CRN for magnitude spectral mapping. We additionally incorporate a newly-developed group strategy to substantially reduce the number of trainable parameters and the computational cost without sacrificing performance.

***Index Terms*—** complex spectral mapping, convolutional recurrent network, causal system, monaural speech enhancement

1. INTRODUCTION

Monaural speech enhancement is the task of separating target speech from background noise given a single microphone recording. It has been extensively studied in speech signal processing in the last decades. Inspired by the concept of time-frequency (T-F) masking in computational auditory scene analysis (CASA), speech enhancement has been formulated as a supervised learning problem in recent years [1]. For supervised speech enhancement, a proper selection of the training target is important for both learning and generalization [2]. Typically, training targets are defined on the T-F representations of speech signals, such as a spectrogram that is computed from a short-time Fourier transform (STFT). These training targets mainly fall into two groups. One group is masking-based targets like the ideal ratio mask (IRM) [3], which describe the time-frequency relationships between clean speech and background noise. Another is mapping-based targets such as log-power spectrum (LPS) [4], which correspond to the spectral representations of clean speech.

Most of these training targets operate on the magnitude spectrogram of noisy speech. In other words, typical speech enhancement systems enhance only the magnitude spectrogram and use the noisy phase spectrogram to reconstruct the time-domain waveform. The reason for not enhancing the phase spectrogram is two-fold. First, no clear structure exists in the phase spectrogram, which makes it intractable to directly estimate the phase spectrogram of clean speech [5]. Second, it was believed that an enhanced phase spectrogram does not lead to a significant improvement in speech quality [6]. A more recent study [7], however, shows that considerable improvements in both objective and subjective speech quality can be achieved by accurate phase spectrum estimation. Based on this observation, various phase enhancement algorithms for speech separation have been developed [8] [9] [10]. However, these algorithms do not address the magnitude spectrum. Williamson *et al.* [5] found that both real and imaginary components of the clean speech spectrogram show clear structure and are thus amenable to supervised learning. In [5], they developed the complex ideal ratio mask (cIRM) and employed a deep neural network (DNN) to jointly estimate real and imaginary components. Unlike the algorithms in [8], [9] and [10], the cIRM can simultaneously enhance both the magnitude and phase spectra of noisy speech. Their experimental results show that the estimated complex ratio mask (cRM) yields better speech quality over IRM estimation while achieving slight or no improvements in objective intelligibility.

Subsequently, Fu *et al.* [11] proposed a CNN to estimate clean real and imaginary spectrograms from the noisy ones. The estimated real and imaginary spectrograms are then utilized to reconstruct the time-domain waveform. The experimental results show that the CNN leads to better objective intelligibility and perceptual quality over a DNN. Additionally, they trained a DNN to map from the noisy LPS features to the clean ones. They found that complex spectral mapping using a DNN yields a 2.4% improvement in short-time objective intelligibility (STOI) [12] and a 0.21 improvement in perceptual evaluation of speech quality (PESQ) [13] over LPS spectral mapping using a DNN.

Motivated by our recent work [14] on CRNs, we propose a CRN architecture to perform complex spectral mapping for noise- and speaker-independent speech enhancement. In our proposed CRN, we incorporate a newly-developed technique to reduce the number of trainable parameters and the computational cost. Moreover, our enhancement system is causal, which is necessary for real-time speech enhancement in many real-world applications. In our experiments, we find that the proposed CRN substantially outperforms the CNN in [11] in terms of STOI and PESQ. In addition, the results show that complex spectral mapping using the CRN leads to significant STOI and PESQ improvements over magnitude spectral mapping using a CRN in [14]. We also find that complex spectral mapping consistently outperforms complex ratio masking and cRM-based signal approximation with the same CRN architecture.

The rest of this paper is organized as follows. We provide a detailed description of our proposed model in Section 2. The ex-

This research was supported in part by an NIDCD grant (R01 DC012048), an NSF grant (IIS-1409431), and the Ohio Supercomputer Center.

perimental setup and results are presented in Section 3. Section 4 concludes this paper.

2. ALGORITHM DESCRIPTION

2.1. Training targets

2.1.1. Target magnitude spectrum

The target magnitude spectrum (TMS) of clean speech is a standard training target in mapping-based approaches [15] [16]. In this case, a mapping from the magnitude spectrogram of noisy speech to that of clean speech is learned. The estimated magnitude spectrum is then combined with the noisy phase spectrum to resynthesize the enhanced speech waveform.

2.1.2. Target complex spectrum

In our proposed method, we use the real and imaginary spectrograms of clean speech as the training target. This training target is referred to as the target complex spectrum (TCS). In contrast to the TMS, the TCS can perfectly reconstruct clean speech.

2.1.3. Complex ideal ratio mask

The complex ideal ratio mask is an ideal mask defined in the complex domain [5]:

$$cIRM = \frac{Y_r S_r + Y_i S_i}{Y_r^2 + Y_i^2} + i\frac{Y_r S_i - Y_i S_r}{Y_r^2 + Y_i^2} \tag{1}$$

where Y_r and Y_i denote real and imaginary components of noisy speech spectrogram, respectively, and S_r and S_i real and imaginary components of clean speech spectrogram, respectively. The imaginary unit is represented by 'i'.

2.1.4. Complex ratio mask based signal approximation

In typical signal approximation, a ratio mask estimator is trained to minimize the difference between the spectral magnitude of clean speech and that of estimated speech [17]. Analogously, we can train a complex ratio mask estimator to minimize the difference between the complex spectrum of clean speech and that of estimated speech:

$$SA = |cRM * Y - S|^2 \tag{2}$$

where S and Y denote the spectrograms of clean speech and noisy speech, respectively, and '$*$' denotes complex multiplication. The complex modulus is represented by $|\cdot|$. We call the resulting training target cRM-based signal approximation (cRM-SA).

2.2. Convolutional recurrent network

In [14], we have recently developed a convolutional recurrent network, which benefits from the feature extraction capability of CNNs and the temporal modeling capability of recurrent neural networks (RNNs), by combining the two topologies together. The CRN is essentially an encoder-decoder architecture. Specifically, the encoder comprises five convolutional layers, and the decoder five deconvolutional layers. Between the encoder and the decoder, two long short-term memory (LSTM) layers are inserted to model the temporal dependencies. Additionally, skip connections are utilized to concatenate the output of each encoder layer to the input of the corresponding decoder layer. In the CRN, all convolutions and deconvolutions are causal, so that the enhancement system does not use future information. Fig. 1 depicts the CRN in [14] for magnitude spectral mapping.

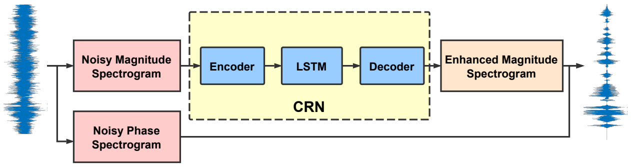


Fig. 1. (Color Online). Illustration of the CRN for magnitude spectral mapping in [14]. The CRN comprises three modules: an encoder module, an LSTM module and a decoder module.

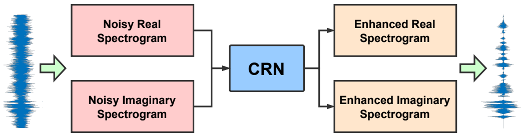
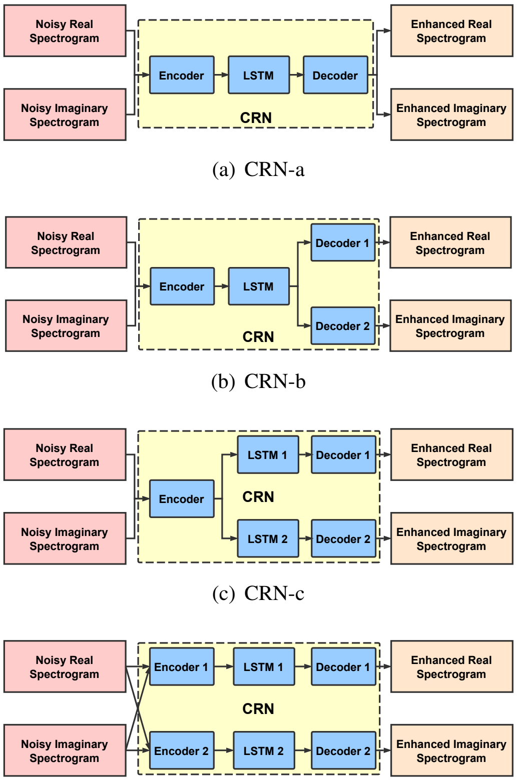

Noisy Real Spectrogram
Noisy Imaginary Spectrogram
CRN
Enhanced Real Spectrogram
Enhanced Imaginary Spectrogram

Fig. 2. (Color Online). Overview of complex spectral mapping using a CRN for speech enhancement.

In this study, we extend this CRN architecture to perform complex spectral mapping. An overview of complex spectral mapping using a CRN is illustrated in Fig. 2. Note that the real and imaginary spectrograms of noisy speech are treated as two different input channels as in [11]. To investigate the extent of correlation between real spectrogram estimation and imaginary spectrogram estimation, we consider four candidate CRN architectures using different parameter sharing approaches. These architectures are illustrated in Fig. 3. In the first architecture (Fig. 3(a)), the encoder module, the LSTM module and the decoder module are shared for the estimation of real and imaginary components. The real and imaginary spectrograms of enhanced speech are treated as two different output channels in the last deconvolutional layer of the decoder. In the second architecture (Fig. 3(b)), the encoder module and the LSTM module are shared, while two distinct decoder modules are employed to estimate real and imaginary components, respectively. In the third architecture (Fig. 3(c)), only the encoder module is shared, and two LSTM modules and two decoder modules are used for the estimation of real and imaginary components, respectively. In the fourth architecture (Fig. 3(d)), two distinct CRNs are utilized to estimate the real and imaginary spectrograms of enhanced speech, respectively. Both CRNs take the real and imaginary spectrograms of noisy speech as input features. We denote the four CRN architectures as *CRN-a*, *CRN-b*, *CRN-c* and *CRN-d*, respectively.

The selection of the parameter sharing mechanism may be important for both learning and generalization. Note that the real spectrogram estimation and the imaginary spectrogram estimation can be considered as two distinct subtasks. On one hand, parameter sharing can achieve a regularization effect between the subtasks, which may lead to better generalization. Moreover, the learning may be encouraged by parameter sharing, particularly when two subtasks are highly correlated. On the other hand, excessive parameter sharing between the subtasks could discourage the learning, especially when the two subtasks are weakly correlated.

In our experiments, we find that *CRN-b* and *CRN-c* achieve better performance over *CRN-a* and *CRN-d* in both STOI and PESQ metrics, while *CRN-b* and *CRN-c* yield similar STOI and PESQ scores. Moreover, the different parameter sharing mechanisms in the four architectures amount to different model sizes: *CRN-a* $<$ *CRN-b* $<$ *CRN-c* $<$ *CRN-d*. Therefore, we propose to use *CRN-b* due to



(a) CRN-a

(b) CRN-b

(c) CRN-c

(d) CRN-d

Fig. 3. (Color Online). Illustration of the four candidate CRN architectures with different parameter sharing mechanisms.

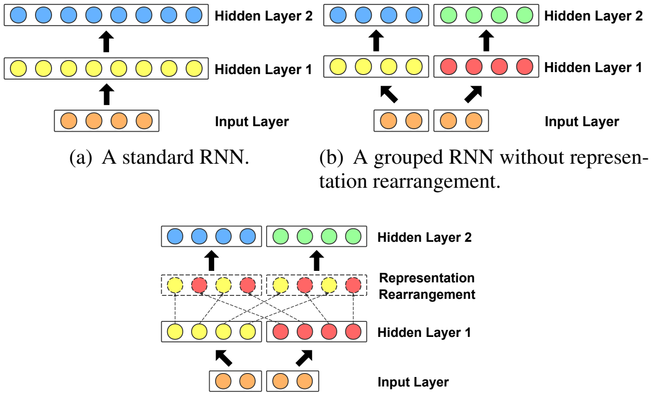
its higher model efficiency over *CRN-c*. Note that all our subsequent extensions are based on *CRN-b*.

2.3. Model capacity reduction via grouped LSTM

Model efficiency is important for many real-world applications. Gao *et al.* [18] have recently proposed a group strategy to improve the efficiency of recurrent layers while maintaining their performance. Fig. 4 depicts the group strategy developed in [18]. In a recurrent layer, both the input features and the hidden states are split into disjoint groups, and intra-group features are learned separately within each group, as shown in Fig. 4(b). Obviously, the model complexity is substantially reduced by the grouping operation. The inter-group dependency, however, cannot be captured. In other words, an output only depends on the input in the corresponding feature group, which significantly degrades the representation power. To mitigate this problem, a parameter-free representation rearrangement layer between two consecutive recurrent layers is used to repermute the features and hidden states, so that the inter-group correlations are recovered (Fig.4(c)). In order to achieve an efficient model, we adopt this group strategy for the two LSTM layers in our CRN architecture (*i.e. CRN-b* in Section 2.2).

2.4. Network architecture

In this study, we assume that all signals are sampled at 16 kHz. A 20-ms Hamming window is used to segment a signal into a set of time frames, where adjacent time frames are overlapped by 50%. We use 161-dimensional spectra that are calculated from a 320-point STFT.



(a) A standard RNN.

(b) A grouped RNN without representation rearrangement.

(c) A grouped RNN with representation rearrangement.

Fig. 4. (Color Online). Illustration of the group strategy for RNNs.

Table 1. Our proposed CRN architecture. Here T denotes the number of time frames in the spectrogram.

layer name	input size	hyperparameters	output size
conv2d_1	$2 \times T \times 161$	$1 \times 3, (1, 2), 16$	$16 \times T \times 80$
conv2d_2	$16 \times T \times 80$	$1 \times 3, (1, 2), 32$	$32 \times T \times 39$
conv2d_3	$32 \times T \times 39$	$1 \times 3, (1, 2), 64$	$64 \times T \times 19$
conv2d_4	$64 \times T \times 19$	$1 \times 3, (1, 2), 128$	$128 \times T \times 9$
conv2d_5	$128 \times T \times 9$	$1 \times 3, (1, 2), 256$	$256 \times T \times 4$
reshape_1	$256 \times T \times 4$	-	$T \times 1024$
grouped_lstm_1	$T \times 1024$	1024	$T \times 1024$
grouped_lstm_2	$T \times 1024$	1024	$T \times 1024$
reshape_2	$T \times 1024$	-	$256 \times T \times 4$
deconv2d_5 ($\times$ 2)	$512 \times T \times 4$	$1 \times 3, (1, 2), 128$	$128 \times T \times 9$
deconv2d_4 ($\times$ 2)	$256 \times T \times 9$	$1 \times 3, (1, 2), 64$	$64 \times T \times 19$
deconv2d_3 ($\times$ 2)	$128 \times T \times 19$	$1 \times 3, (1, 2), 32$	$32 \times T \times 39$
deconv2d_2 ($\times$ 2)	$64 \times T \times 39$	$1 \times 3, (1, 2), 16$	$16 \times T \times 80$
deconv2d_1 ($\times$ 2)	$32 \times T \times 80$	$1 \times 3, (1, 2), 1$	$1 \times T \times 161$
concat	$1 \times T \times 161$ ($\times$ 2)	-	$2 \times T \times 161$

A detailed description of our proposed network architecture is provided in Table 1. The input size and the output size of each layer are given in *featureMaps* $\times$ *timeSteps* $\times$ *frequencyChannels* format. In addition, the layer hyperparameters are specified in (*kernelSize*, *strides*, *outChannels*) format. Note that the number of feature maps in each decoder layer is doubled by the skip connections. Unlike the 2×3 (*time* $\times$ *frequency*) kernels in [14], we use a kernel size of 1×3, without degrading the performance. We employ exponential linear units (ELUs) [19] in all convolutional and deconvolutional layers except the output layer. In the output layer, we use linear activation for spectrogram estimation. Moreover, batch normalization [20] is adopted right after each convolution (or deconvolution) and before activation.

3. EXPERIMENTS

3.1. Experimental setup

In our experiments, we use the WSJ0 SI-84 training set [21] which includes 7138 utterances from 83 speakers (42 males and 41 females). We set aside six (3 males and 3 females) of these speakers as untrained speakers for test. In other words, we train the models with the 77 remaining speakers. For training, we use 10,000 noises from a sound effect library (available at https://www.sound-ideas.com), and the duration is about 126 hours. For test, we use two highly nonstationary noises (babble and cafeteria) from an Auditec CD (available at http://www.auditec.com).

Our training set includes 320,000 mixtures with a total duration

Table 2. Comparisons of different parameter sharing mechanisms in terms of STOI and PESQ on untrained noises and untrained speakers. The numbers represent the averages over the two test noises (the same as Tables 3 and 4).

metrics	STOI (in %)			PESQ		
SNR	-5 dB	0 dB	5 dB	-5 dB	0 dB	5 dB
unprocessed	57.94	70.01	81.20	1.51	1.80	2.12
CRN-a	78.71	89.06	93.75	2.14	2.67	3.06
CRN-b	**80.12**	**89.68**	**94.03**	**2.19**	**2.70**	**3.07**
CRN-c	**80.36**	**89.70**	**94.08**	**2.19**	**2.70**	**3.07**
CRN-d	79.38	89.28	93.80	2.16	2.67	3.04

Table 3. Comparisons of different models and training targets in STOI and PESQ metrics on untrained noises and untrained speakers. Note that K denotes the group number in the grouped LSTM layers. $K = 1$ means that grouping is not performed.

metrics	STOI (in %)			PESQ		
SNR	-5 dB	0 dB	5 dB	-5 dB	0 dB	5 dB
unprocessed	57.94	70.01	81.20	1.51	1.80	2.12
LSTM + TMS	74.84	85.66	91.57	1.97	2.43	2.81
CRN + TMS [14]	76.28	86.15	91.92	2.02	2.46	2.83
CRN + cIRM (K=2)	74.83	86.05	91.99	1.94	2.44	2.85
CRN + cRM-SA (K=2)	77.73	88.44	93.56	2.03	2.56	2.96
CNN + TCS [11]	66.42	80.39	87.91	1.64	2.11	2.47
CRN + TCS (K=1)	80.12	89.68	94.03	**2.19**	**2.70**	**3.07**
CRN + TCS (K=2)	**80.14**	**89.84**	94.15	2.17	2.68	3.05
CRN + TCS (K=4)	80.01	89.78	**94.21**	2.18	2.69	**3.07**
CRN + TCS (K=8)	78.63	89.06	93.83	2.15	2.67	3.05

of about 500 hours. To create a training mixture, we mix a randomly drawn training utterance with a random cut from the 10,000 training noises. The signal-to-noise ratio (SNR) is randomly sampled from $\{$-5, -4, -3, -2, -1, 0$\}$ dB. Our test set comprises 150 mixtures created from 25 × 6 utterances of 6 untrained speakers. We use three SNRs for the test set, *i.e.* -5, 0 and 5 dB.

We train the models using the AMSGrad optimizer [22] with a learning rate of 0.001. The mean squared error (MSE) is used as the objective function. The minibatch size is set to 16 at the utterance level. Within a minibatch, all training samples are zero-padded to have the same number of time steps as the longest sample.

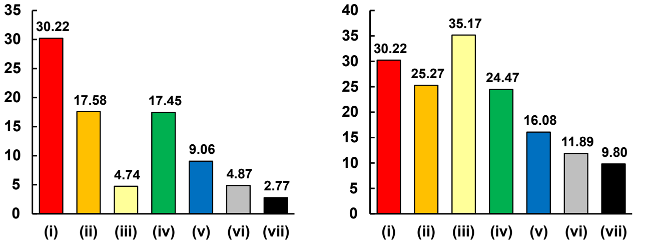
3.2. Experimental results

3.2.1. Comparisons of different parameter sharing mechanisms

We investigate the four candidate architectures discussed in Section 2.2, *i.e. CRN-a*, *CRN-b*, *CRN-c* and *CRN-d*. Table 2 lists STOI and PESQ scores yielded by the four architectures. Note that we do not use the group strategy in these architectures. We can observe that *CRN-b* and *CRN-c* consistently outperform *CRN-a* and *CRN-d* in all conditions. For example, a 1.41% STOI improvement and a 0.05 PESQ improvement is achieved by going from *CRN-a* to *CRN-b* at -5 dB SNR. Going from *CRN-b* to *CRN-d* degrades the performance in both metrics, which reveals the advantage of parameter sharing in a proper way.

3.2.2. Comparisons of different models and training targets

Table 3 presents comprehensive evaluations for different models and training targets on untrained noises and untrained speakers. The best results in each case are highlighted by boldface. We first compare our proposed CRN architecture with different group numbers using the TCS as the training target, as shown in the last four rows of Table 3. We can see that $K = 2$ and $K = 4$ produce similar results to $K = 1$ (*i.e.* no grouping). Further increasing the group number degrades the enhancement performance (*e.g.* $K = 8$). Moreover, our proposed CRN model significantly outperforms the CNN in [11]. Take, for example the -5 dB SNR case. The proposed CRN with $K = 2$ improves STOI by 13.72% and PESQ by 0.53 over the CNN.



(a) # trainable parameters.

(b) # floating-point fused multiply-adds.

Fig. 5. (Color Online). The numbers of trainable parameters and floating-point fused multiply-adds (per time frame) in different models. The unit in both figures is million. The models are (i) LSTM, (ii) CRN [14], (iii) CNN [11], (iv) CRN (K=1), (v) CRN (K=2), (vi) CRN (K=4) and (vii) CRN (K=8), respectively.

Table 4. Evaluation of phase estimation provided by complex spectral mapping.

metrics	STOI (in %)			PESQ		
SNR	-5 dB	0 dB	5 dB	-5 dB	0 dB	5 dB
unprocessed	57.94	70.01	81.20	1.51	1.80	2.12
noisy phase	76.28	86.15	91.92	2.02	2.46	2.83
estimated phase	78.49	88.72	93.73	2.14	2.65	3.02
clean phase	80.83	90.34	94.86	2.35	2.85	3.22

As Table 3 shows, our proposed CRN with the TCS achieves better performance than the same CRN with the cIRM and cRM-SA, as well as an LSTM and the CRN in [14] with the TMS. For example, the proposed CRN ($K = 2$) with the TCS yields a 2.41% STOI improvement and a 0.14 PESQ improvement compared with the cRM-SA. In our experiment, the LSTM model has the same architecture as [23], except that we do not use the feature window in order to achieve a causal system. Additionally, it should be noted that the cRM-SA leads to higher STOI and PESQ scores over the cIRM. Fig. 5(a) shows the numbers of trainable parameters in different models, and Fig. 5(b) the numbers of floating-point fused multiply-adds that are needed to process one time frame. We find that our proposed model achieves high efficiency in both metrics.

3.2.3. Evaluation of phase estimation via complex spectral mapping

An estimate of the TCS can provide a phase estimate simply by recovering the phase response from the estimated real and imaginary spectrograms. Now we quantify the effectiveness of the estimated phase by taking the magnitude spectrogram estimated by the CRN in [14] and reconstructing the waveform with three different phases: the noisy phase, the estimated phase and the clean phase. As shown in Table 4, our proposed approach produces an estimated phase that is better than the noisy phase.

4. CONCLUSION

In this study, we have proposed a new framework for complex spectral mapping using a convolutional recurrent network. The enhancement system is causal, and noise- and speaker-independent. In terms of STOI and PESQ, it significantly outperforms an existing CNN for complex spectral mapping, as well as a strong CRN for magnitude spectral mapping. Moreover, we incorporate a newly-developed group strategy to substantially elevate the model efficiency while maintaining the performance.

5. REFERENCES

[1] D. L. Wang and G. J. Brown, Eds., *Computational auditory scene analysis: Principles, algorithms, and applications*, Wiley-IEEE press, 2006.

[2] D. L. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2018.

[3] Y. Wang, A. Narayanan, and D. L. Wang, "On training targets for supervised speech separation," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 12, pp. 1849–1858, 2014.

[4] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal processing letters*, vol. 21, no. 1, pp. 65–68, 2014.

[5] D. S. Williamson, Y. Wang, and D. L. Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 3, pp. 483–492, 2016.

[6] D. L. Wang and J. Lim, "The unimportance of phase in speech enhancement," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 30, no. 4, pp. 679–681, 1982.

[7] K. Paliwal, K. Wójcicki, and B. Shannon, "The importance of phase in speech enhancement," *speech communication*, vol. 53, no. 4, pp. 465–494, 2011.

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

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

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

[11] S.-W. Fu, T.-y. Hu, Y. Tsao, and X. Lu, "Complex spectrogram enhancement by convolutional neural network with multi-metrics learning," in *IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2017, pp. 1–6.

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

[13] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2001, vol. 2, pp. 749–752.

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

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

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

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

[18] F. Gao, L. Wu, L. Zhao, T. Qin, X. Cheng, and T.-Y. Liu, "Efficient sequence learning with group recurrent networks," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, vol. 1, pp. 799–808.

[19] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," in *International Conference on Learning Representations*, 2016.

[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, 2015, pp. 448–456.

[21] D. B. Paul and J. M. Baker, "The design for the wall street journal-based csr corpus," in *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1992, pp. 357–362.

[22] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," in *International Conference on Learning Representations*, 2018.

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