Interspeech 2018
2-6 September 2018, Hyderabad

Real-time Single-channel Dereverberation and Separation with Time-domain Audio Separation Network

Yi Luo *Nima Mesgarani*

Department of Electrical Engineering, Columbia University, New York, NY

yl3364@columbia.edu nima@ee.columbia.edu

Abstract

We investigate the recently proposed Time-domain Audio Separation Network (TasNet) in the task of real-time single-channel speech dereverberation. Unlike systems that take time-frequency representation of the audio as input, TasNet learns an adaptive front-end in replacement of the time-frequency representation by a time-domain convolutional non-negative autoencoder. We show that by formulating the dereverberation problem as a denoising problem where the direct path is separated from the reverberations, a TasNet denoising autoencoder can outperform a deep LSTM baseline on log-power magnitude spectrogram input in both causal and non-causal settings. We further show that adjusting the stride size in the convolutional autoencoder helps both the dereverberation and separation performance.

Index Terms: speech dereverberation, speech separation, time-domain, deep learning

1. Introduction

Real-world speech communication often takes place in crowded or reverberant conditions where the speech signal is corrupted by other speakers, environmental noises, or room reverberations. A successful system in such conditions thus requires robust speech separation or speech dereverberation function. Moreover, in such applications where real-time processing is necessary, the latency of the system remains an important limiting issue.

In recent years, deep learning systems have shown to have better generalization ability and higher performance in various conditions for both separation and dereverberation [1, 2, 3, 4, 5, 6, 7, 8, 9]. In most of the systems, a time-frequency (T-F) representation is calculated from the audio waveform as the input by short-time Fourier transform (STFT). In speech separation tasks, a general method is to estimate a T-F mask for each of the speaker in the mixture. In dereverberation, the anechoic T-F representation is typically estimated from the reverberant signal. The reconstruction of the waveforms is then done by inverse STFT. However, there are several issues with the usage of T-F representations. First, performance of STFT-based systems is related to the choice of the window length in STFT, which directly affects the frequency resolution as well as the system latency. In many systems, a window size that is longer than 32 ms is required to achieve a good performance [1, 2, 10]. This limits the use of such systems in applications where a very short latency is required, such as hearing aids and telecommunication devices. Additionally, most of the systems for separation and dereverberation only modify the magnitude spectrogram or the mel-frequency cepstral coefficient (MFCC) while the phase spectrogram remains unchanged [3, 5, 6]. This limits the performance upper-bound due to the usage of the noisy phase during inverse STFT. Although there are methods such as phase-sensitive mask for separation [11] or complex ratio mask and iterative reconstruction for dereverberation [12, 7], the performance is still limited and model complexity might be much higher.

Modeling the signals directly in time-domain may remedy the issues mentioned above. A recently proposed neural network, the Time-domain Audio Separation Network (TasNet [4]), is a deep learning system that operates in the time-domain. TasNet models the input waveform with a 1-D convolutional encoder-decoder framework where the output of the encoder forms a non-negative adaptive front-end (representation) to replace the STFT. The target sources are estimated by calculating mask-like matrices that are applied to the non-negative representation of the input, which is similar to the typical mask estimation process in STFT-based systems. Because all of the operations in TasNet are in the time-domain, there is no upper-bound performance due to the noisy phase spectrogram, and the latency of the system can be controlled by the length of the 1-D filters in the convolutional autoencoder. Comparing with STFT-based systems, the latency of TasNet can be as low as 5ms [4], which makes it possible for real-time low latency applications.

It was shown that TasNet outperformed the state-of-the-art STFT-based systems on the separation task in both causal and non-causal configurations [4]. However, whether TasNet is effective in the problem of single-channel dereverberation is unknown. In this paper, we investigate the usage of TasNet as a denoising autoencoder (DAE) in the problem of speech dereverberation. We formulate the dereverberation problem as a separation problem, where the reverberant speech is treated as the summation of the direct path and the reverberant noise. A similar mask estimation process is designed to extract the direct path from the reverberant input. Based on the observation on the dereverberation problem, we further show that by adding overlap between the windows (i.e. adjusting the stride size in the convolutional autoencoder), the performance of dereverberation and separation can both be improved.

The rest of the paper is organized as follows. Section 2 describes the problem formulation of the dereverberation task. Section 3 considers the TasNet architecture for dereverberation. Section 4 provides the details about the experiments. Section 5 concludes the paper.

2. Problem Description

A reverberant speech signal is composed of the direct signal $x^{(d)}(t)$ and the remaining reverberant noise $x^{(e)}(t)$

$$x(t) = x^{(d)}(t) + x^{(e)}(t) \tag{1}$$

In real-time applications, audio signals typically come in streams or segments. At each time step, we assume that an au-

10.21437/Interspeech.2018-2290

dio stream with length of L samples is received

$$\begin{cases} \mathbf{x}_k = x(t) \\ \mathbf{x}_k^{(d)} = x^{(d)}(t) \\ \mathbf{x}_k^{(e)} = x^{(e)}(t) \end{cases} \quad t \in [kH, kH + L),\ k = 1, \ldots, K \quad (2)$$

where $\mathbf{x}_k, \mathbf{x}_k^{(d)}, \mathbf{x}_k^{(e)} \in \mathbb{R}^{1\times L}$ and H denotes the hop size between streams. K stands for the total number of audio streams and varies from utterance to utterance. We drop the notation k where there is no ambiguity.

The aim of dereverberation is to estimate the direct signal $\mathbf{x}^{(d)}$ from $\mathbf{x}$. Following the idea from the original TasNet, a set of trainable basis signals $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_N] \in \mathbb{R}^{N\times L}$ is used to represent each of the segments with a set of non-negative weights through a deconvolutional operation

$$\begin{cases} \mathbf{x} = Deconv(\mathbf{w}, \mathbf{B}) \\ \mathbf{x}^{(d)} = Deconv(\mathbf{w}^{(d)}, \mathbf{B}) \end{cases} \quad (3)$$

where the weight vectors $\mathbf{w}, \mathbf{w}^{(d)} \in \mathbb{R}^{1\times N}$. With the non-negativity constraint, a mask-like vector $\mathbf{m}^{(d)} \in \mathbb{R}^{1\times N}$ can be estimated for $\mathbf{w}^{(d)}$

$$\mathbf{w}^{(d)} = \mathbf{w} \odot (\mathbf{w}^{(d)} \oslash \mathbf{w}) \quad (4)$$

$$:= \mathbf{w} \odot \mathbf{m}^{(d)} \quad (5)$$

where $\odot$ and $\oslash$ denotes element-wise multiplication and division. Therefore, the problem of estimating the direct path is equivalent to estimating a mask-like vector which is applied to a representation of the reverberant speech.

3. TasNet for Dereverberation

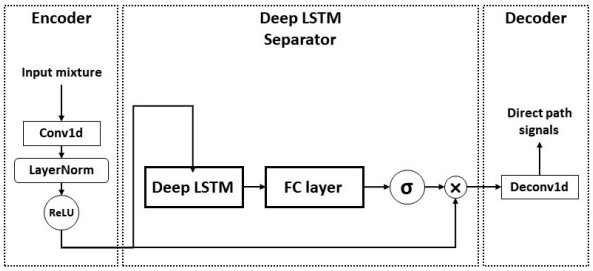

Figure 1: *Time-domain Audio Separation Network (TasNet) models the input signal in the time-domain using a convolutional encoder-decoder framework. The output of the encoder forms the non-negative representation for the input, and a mask for the target direct path is learned from the separator and applied to the encoder output. The decoder then reconstructs the waveform through a deconvolutional operation.*

The TasNet architecture contains one 1-D convolutional layer as the non-negative encoder and several recurrent layers for mask estimation, and one linear 1-D deconvolutional layer as the decoder. The encoder output serves as an adaptive front-end representation for the time-domain signal to replace the STFT feature. The decoder inverts the convolutional operation in the encoder by performing deconvolution with a set of trainable basis signals (filters) and reconstructs the waveforms. The recurrent layers estimate the masks using the representation generated by the encoder. Figure 1 shows the flowchart of the system.

3.1. 1-D convolutional encoder

The encoder consists of a 1-D convolutional layer with ReLU activation for the non-negativity constraint

$$\mathbf{w} = ReLU(LN(\mathbf{x} \circledast \mathbf{U})) \quad (6)$$

where $\mathbf{U} \in \mathbb{R}^{N\times L}$ is the trainable parameter, $\mathbf{x} \in \mathbb{R}^{1\times L}$ is a segment of the input mixture, and N is the number of channels (i.e. the number of the filters). $\circledast$ denotes the convolution operator. LN corresponds to the layer normalization operation [13]. The layer normalization operation is applied here to ensure that the encoder is invariant to input rescaling, meaning that changing the energy of the signal will not affect the separation performance.

In [4], it was mentioned that a gated CNN architecture [14] is helpful for the convergence speed and final performance. However, we find empirically that with proper training, using ReLU as the only activation function does not harm the performance of the network.

3.2. Deep LSTM separation module

The separation module contains several stacked LSTM layers followed by a fully-connected layer for mask estimation. The input to the separation module is the sequence of K input weight vectors $\mathbf{w}_1, \ldots \mathbf{w}_K \in \mathbb{R}^{1\times N}$, and the output is the mask-like vectors for the target sources. For dereverberation, the output is only one mask $\mathbf{m}^{(d)}$ corresponds to the direct path. Sigmoid activation function is used in the fully-connected layer.

A layer normalization style operation is applied to the input of the separation module in order to speed up and stabilize the training process

$$\bar{\mathbf{w}} = \frac{\mathbf{g}}{\sigma} \odot (\mathbf{w} - \mu) + \mathbf{b} \quad (7)$$

$$\mu = \frac{1}{N}\sum_{j=1}^{N} w_j \quad \sigma = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(w_j - \mu)^2} \quad (8)$$

where parameters $\mathbf{g} \in \mathbb{R}^{1\times N}$ and $\mathbf{b} \in \mathbb{R}^{1\times N}$ are gain and bias vectors that are jointly optimized with the network. We find this important for the network to converge reliably.

In order to accelerate the training process and enhance the gradient flow, an identity skip connection [15] is added between every two LSTM layers. A linear fully-connected layer is applied to the input to the separation module for reshaping it to the same size as the output of the second LSTM layer.

After the mask vector $\mathbf{m}^{(d)} \in \mathbb{R}^{1\times N}$ for the direct path of each source is generated, the weight vector $\hat{\mathbf{w}}_i^{(d)} \in \mathbb{R}^{1\times N}$ for each segment is calculated by multiplying $\mathbf{m}_i^{(d)}$ with the input weight vector $\mathbf{w}$, as in equation 5.

3.3. 1-D deconvolutional decoder

The decoder is a 1-D deconvolutional layer to invert the convolution operation in the encoder for time-domain signal reconstruction. The waveform for each signal in each segment is calculated by the 1-D deconvolution between the weight vector and a set of trainable 1-D filters $\mathbf{B} \in \mathbb{R}^{N\times L}$

$$\hat{\mathbf{x}}_i^{(d)} = Deconv(\hat{\mathbf{w}}_i^{(d)}, \mathbf{B}) \quad (9)$$

$$\hat{\mathbf{x}}_i = Deconv(\hat{\mathbf{w}}_i, \mathbf{B}) \quad (10)$$

The 1-D filters $\mathbf{B}$ are parameters in the deconvolutional layer and are jointly optimized with all the other parts of the

network. The entire waveforms are then obtained by concatenating all the segments. The reconstructions in the overlapped parts in consecutive segments are summed up to form the final output.

3.4. Training objective

In [4], it is reported that using scale-invariant source-to-noise ratio (SI-SNR) as the objective led to better performance in the separation task. However, SI-SNR leads to much slower convergence and worse performance on the dereverberation task, possibly due to the auto-correlated structure between the direct path and the reverberant noise. Here, we use the mean-square error (MSE) between the estimated direct path and the real direct path, as well as the estimated recovered input and the noisy input as the objective

$$\mathcal{L} = MSE(\hat{\mathbf{x}}^{(d)}, \mathbf{x}^{(d)}) + MSE(\hat{\mathbf{x}}, \mathbf{x}) \tag{11}$$

Note that the second term in equation 11 is to ensure that $\mathbf{w}$ correctly represent the input signal, which is necessary because the direct path in dereverberation problem is part of the input.

4. Experiments

4.1. Dataset

4.1.1. Dereverberation

A simulated reverberant speech dataset is generated from the Wall Street Journal (WSJ0) dataset with three different room reverb characteristics. Table 1 shows the characteristics of the rooms. One microphone is located at the center of the room. The room impulse responses (RIRs) are generated with the image method [16]. A training set of 20000 samples (30 hours in total) and a validation set of 6000 samples (10 hours in total) are generated from randomly selected utterances from the WSJ0 training set si_tr_s. A test set of 5000 samples (8 hours in total) is generated from randomly selected speakers in WSJ0 si_dt_05 and si_et_05 datasets. The sample rate for all utterances is set to 8kHz.

During the generation of the reverberant speech, a random utterance is first drawn from the clean speech dataset. A room and its corresponding T_{60} is also randomly selected. The speaker is then randomly placed in the room with at least 0.5m from the borders. The height of the speaker is restricted between 1m to 2m.

Table 1: *Characteristics of different rooms for dereverberation simulation.*

	Size (m)	T_{60} (s)
Small	$3 \times 5 \times 3$	0.3
Medium	$5 \times 8 \times 3$	0.6
Large	$8 \times 11 \times 3$	0.9

4.1.2. Separation

For the separation task, we use the WSJ0-2mix dataset [1, 2, 10], which contains 30 hours of training and 10 hours of validation data. The mixtures are generated by randomly selecting utterances from different speakers in WSJ0 training set si_tr_s, and mixing them at random signal-to-noise ratios (SNR) between 0 dB and 5 dB. Five hours of evaluation set are generated in the same way using utterances from 16 unseen speakers from si_dt_05 and si_et_05 in the WSJ0 dataset. The sample rate is also set to 8kHz.

4.2. Network configuration

Table 2: *Network configurations for different tasks.*

Task	Causal	(N, H, L)	Separator
Dereverb	✓	$(250, 40, 40)$	$4 \times 500 + 250$
	×		$4 \times 250 + 250$
Separate	✓	$(500, 40, 40)$	$4 \times 1000 + 1000$
	×		$4 \times 500 + 1000$

The parameters of the system include the segment length L, the hop size H, the number of basis signals N, and the configuration of the separator module. The parameters of the separator include the number of (Bi-)LSTM layers, the number of hidden units in each (Bi-)LSTM layer, and the number of hidden units in the fully-connected layer. Table 2 shows the configuration of networks for different tasks. Note that setting the hop size H to be equal to the window size L means that there is no overlap between two consecutive segments. In Section 4.4 we will discuss the effect of overlap in both tasks.

For the dereverberation task, we design another baseline deep LSTM (DLSTM) DAE model with log-power magnitude spectrogram input. The window size and hop size of STFT are 256 samples (32ms) and 64 samples (8ms), respectively. This results in a 129-dimensional input feature. The DLSTM DAE contains 4 (Bi-)LSTM layers with the same size as the separator in TasNet, with a fully-connected layer of 129 hidden units for estimating the log-power magnitude spectrogram of the direct path. No activation function is applied in the fully-connected layer. Identity skip connections are added between every two (Bi-)LSTM layers the same way as in Section 3.2.

We also apply the curriculum training strategy [17] in a similar fashion to [4]. We start training the network on 1 second-long utterances for dereverberation and 0.5 second-long utterances for separation, and continue training on 4 second-long utterances afterward. For the DLSTM DAE model, we first train on 100 frame-long utterances (0.8s) and continue on 400 frame-long utterances (3.2s).

4.3. Evaluation metrics

For the dereverberation task, we evaluate the systems using the perceptual evaluation of speech quality (PESQ) [18] and the scale-invariant signal-to-noise ratio (SI-SNR) [1, 4]. For the separation problem, we evaluated the systems with both SI-SNR improvement (SI-SNRi) and SDR improvement (SDRi) [19] metrics used in [1, 2, 10].

4.4. Experiment results

We first investigate the effect of stride size in the convolutional autoencoder of TasNet on the performance. A hop size of $H \leq L$ corresponds to a stride size of $L - H$ in the 1-D convolutional and deconvolutional layers. Table 3 provide the effect of stride on the performance of dereverberation task after having 50% hop size (i.e. adding 50% overlap between segments). We find that adding overlap between segments significantly helps the performance.

We then compare TasNet DAE with DLSTM DAE baseline on the dereverberation task. Table 4 presents the results

Table 3: *PESQ and SI-SNR (dB) for different hop size in TasNet DAE.*

Overlap	Causal	PESQ	SI-SNR
0%	✓	2.13	1.52
50%	✓	**2.24**	**2.20**
0%	×	2.43	2.61
50%	×	**2.55**	**3.38**

of PESQ and SI-SNR of the two systems. We find that TasNet DAE performs significantly better on SI-SNR but worse on PESQ. This can be explained by the usage of the MSE-based objective function which favors SNR more. Nevertheless, we can see that a causal TasNet DAE can still outperform a non-causal DLSTM DAE in terms of SI-SNR. This means that TasNet DAE is able to learn a better mapping between the waveforms of the anechoic signals.

Table 5 compares the system latency in causal TasNet and DLSTM DAE. Similar to [4], the system latency T_{tot} is expressed as the sum of the initial delay of the system T_i and the processing time for a segment T_p. T_i is the length of the segment required to produce the first output, and T_p is estimated as the average per-segment processing time across the entire test set. Both models are loaded on a Titan X Pascal GPU before the processing starts. We observe that the overall latency for TasNet is significantly smaller than the DLSTM DAE, due to the fact that TasNet decouples the window size and the frequency resolution in STFT. This enables the TasNet model to be deployed to real-time and low-latency applications.

Finally, we examine the effect of stride (H) on speech separation task and compared with the other state-of-the-art systems. During the training for TasNet with 50% overlap (TasNet-50%), gradient clipping with maximum norm of 3 was applied to alleviate the gradient explosion problem. We find that this significantly improves the performance. As shown in table 6, although TasNet without overlap (TasNet-0%) already has comparable performance with other systems, TasNet with 50% overlap significantly outperforms all the other systems in both causal and non-causal configurations. This performance boost further proves the efficacy of TasNet in both online and offline settings in comparison with STFT-based systems.

Table 4: *PESQ and SI-SNR (dB) for TasNet DAE and DLSTM DAE baseline.*

	Causal	PESQ	SI-SNR
Mixture	–	2.23	-0.07
TasNet DAE	✓	2.24	**2.20**
DLSTM DAE	✓	**2.42**	0.79
TasNet DAE	×	2.55	**3.38**
DLSTM DAE	×	**2.60**	0.93

Table 5: *Minimum latency (ms) of TasNet and DLSTM DAE in dereverberation task.*

Method	T_i	T_p	T_{tot}
TasNet	5	0.11	**5.11**
DLSTM DAE	32	0.09	32.09

Table 6: *SI-SNR (dB) and SDR (dB) improvements comparison for different hop size in TasNet for separation.*

Method	Causal	SI-SNRi	SDRi
uPIT-LSTM [2]	✓	–	7.0
TasNet-0% [4]	✓	7.9	8.2
TasNet-50%	✓	**10.8**	**11.2**
DPCL++ [1]	×	10.8	–
DANet [10]	×	10.5	–
ADANet [3]	×	10.5	–
uPIT-BLSTM-ST [2]	×	–	10.0
cuPIT-Grid-RD [20]	×	–	10.2
CBLDNN-GAT[21]	×	–	11.0
Chimera++ [22]	×	11.5	12.0
WA-MISI-5 [23]	×	12.6	13.1
TasNet-0% [4]	×	10.9	11.2
TasNet-50%	×	**13.2**	**13.6**

5. Conclusion

In this paper, we investigated the performance of a recently proposed neural network for speech separation, the time-domain audio separation network (TasNet), on the task of speech dereverberation. We formulated the dereverberation problem as a denoising problem where the direct path was separated from the echoic noise. Experiments showed that TasNet outperformed a deep LSTM baseline with spectrogram input, and adjusting the stride size in the convolutional autoencoder further improved the performance in both separation and dereverberation tasks.

6. Acknowledgement

This work was funded by a grant from National Institute of Health, NIDCD, DC014279, National Science Foundation CAREER Award, and the Pew Charitable Trusts.

7. References

[1] Y. Isik, J. Le Roux, Z. Chen, S. Watanabe, and J. R. Hershey, "Single-channel multi-speaker separation using deep clustering," *Interspeech 2016*, pp. 545–549, 2016.

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

[3] Y. Luo, Z. Chen, and N. Mesgarani, "Speaker-independent speech separation with deep attractor network," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 4, pp. 787–796, 2018.

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

[5] X. Feng, Y. Zhang, and J. Glass, "Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1759–1763.

[6] M. Mimura, S. Sakai, and T. Kawahara, "Speech dereverberation using long short-term memory," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[7] K. Han, Y. Wang, D. Wang, W. S. Woods, I. Merks, and T. Zhang, "Learning spectral mapping for speech dereverberation and denoising," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 6, pp. 982–992, 2015.

[8] Y. Ueda, L. Wang, A. Kai, X. Xiao, E. S. Chng, and H. Li, "Single-channel dereverberation for distant-talking speech recognition by combining denoising autoencoder and temporal structure normalization," *Journal of Signal Processing Systems*, vol. 82, no. 2, pp. 151–161, 2016.

[9] Y. Zhao, Z.-Q. Wang, and D. Wang, "A two-stage algorithm for noisy and reverberant speech enhancement," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5580–5584.

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

[11] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 708–712.

[12] D. S. Williamson and D. Wang, "Speech dereverberation and denoising using complex ratio masks," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5590–5594.

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

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

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 630–645.

[16] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[17] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. ICML*, 2009, pp. 41–48.

[18] I.-T. Recommendation, "Perceptual evaluation of speech quality (pesq): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," *Rec. ITU-T P. 862*, 2001.

[19] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE transactions on audio, speech, and language processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[20] C. Xu, X. Xiao, and H. Li, "Single channel speech separation with constrained utterance level permutation invariant training using grid lstm," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*. IEEE, 2018.

[21] C. Li, L. Zhu, S. Xu, P. Gao, and B. Xu, "CBLDNN-based speaker-independent speech separation via generative adversarial training," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*. IEEE, 2018.

[22] Z.-Q. Wang, J. Le Roux, and J. R. Hershey, "Alternative objective functions for deep clustering," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[23] Z.-Q. Wang, J. L. Roux, D. Wang, and J. R. Hershey, "End-to-end speech separation with unfolded iterative phase reconstruction," *arXiv preprint arXiv:1804.10204*, 2018.