INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Empirical Analysis of Generalized Iterative Speech Separation Networks

Yi Luo, Cong Han, Nima Mesgarani

Department of Electrical Engineering, Columbia University, USA
yl3364@columbia.edu, ch3212@columbia.edu, nima@ee.columbia.edu

Abstract

Although most existing speech separation networks are designed as a one-pass pipeline where the sources are directly estimated from the mixture, multi-pass or iterative pipelines have been shown to be effective by designing multiple rounds of separation and utilizing separation outputs from a previous iteration as additional inputs for the next iteration. Moreover, such iterative separation pipeline can also be extended to a more general framework where a training objective designed to minimize the discrepancy between the estimated and target sources is applied to different parts of the network. In this paper, we empirically investigate the effect of such generalized iterative separation pipeline by adjusting its configuration in multiple aspects in both training and inference phases. For the training phase, we compare the separation performance of both time-domain and frequency-domain networks with different numbers of iterations following the recent discussions on the model architecture organizations. We also evaluate the effect of parameter sharing across iterations and the necessity of additional training objectives. For the inference phase, we measure the separation performance of various numbers of iterations. Our results show that iterative speech separation is a promising direction and deserves more in-depth analysis and exploration.

Index Terms: speech separation, speech enhancement, iterative separation

1. Introduction

The design of a wide range of speech separation networks follows a general *one-pass* pipeline, where the input mixture waveform is passed to a neural network to directly estimate the target sources [1–6]. On the other hand, recent developments on the *multi-pass* or *iterative* pipeline have shown improved separation performance [7–11]. Conventional iterative separation pipelines were typically designed by certain iterative algorithms, such as Expectation-Maximization (EM) and nonnegative matrix factorization (NMF) [12–14], where multiple iterations are required for the algorithms to converge and achieve a satisfying performance. In neural network-based systems, an iterative speech separation pipeline can be defined as a system that contains multiple rounds of the separation process, where (1) each iteration performs a full separation pipeline, and (2) the separation outputs from a previous iteration can be used as additional information in an upcoming iteration. In single-channel applications, the iterative pipeline has proven better than one-pass pipelines with comparable model complexity [10, 11]. In multi-channel applications, the iterative pipeline can improve the performance of either a vocal activity detector (VAD) or a beamformer [9, 15, 16]. Different features such as speaker-specific embeddings can also be extracted from the previous separation outputs and serve as the additional feature for following iterations [17–20].

A common configuration for such iterative separation pipeline is that the training objective, typically the discrepancy between the estimated and target sources, is applied to all the iterations. By unfolding the iterations into additional layers or modules in a deeper network, such training objective corresponds to a *layer-wise* objective in one-pass pipelines [21, 22]. Moreover, the objective can be further extended to models where a *post-enhancement* stage is applied to each of the separation outputs [8, 23–26]. This gives us a *generalized* iterative separation pipeline where the same training objective is applied to different parts of the network. Different model design and architecture configurations in such generalized iterative separation pipelines may result in different effects on the separation performance. However, a better understanding on the components as well as their combinations is still beneficial for the investigation of the reason behind their effectiveness and the design of improved pipelines.

In this paper, we empirically investigate the effect of multiple components in the generalized iterative separation pipelines. From the model architecture perspective, we adopt networks in both time-domain and frequency-domain similar to the configuration in [10], while we further compare different model architecture organizations discussed in recent studies [27]. Moreover, we compare the training configurations of whether gradients are accumulated for all the iterations or truncated within each iteration. The latter one corresponds to the setting of treating each iteration as a full pipeline with certain *additional bias information* defined by the separation outputs from a previous iteration. Such additional bias information does not involve in the backpropagation process of the current iteration and can be viewed as a data augmentation method for a one-pass system. We also look into the effect of parameter sharing across iterations, the number of iterations, and the use of oracle source as bias information on the separation performance. For the generalized pipeline, we explore whether adding *layer-wise* training objectives as in [21] helps the overall separation performance. For the inference phase, we evaluate the models with up to 3 training iterations on up to 4 inference iterations to test their performance on matched and mismatched training-inference conditions. Our results show that the iterative separation pipeline is a promising framework to reduce the storage requirements of the model parameters without sacrificing the separation performance, and more in-depth analysis and exploration is still needed to further exploit its potentials.

The rest of the paper is organized as follows. Section 2 makes an overview on the standard pipeline of a generalized iterative speech separation system. Section 3 introduces the experiment configurations. Section 4 provides and analyzes the experiment results. Section 5 concludes the paper.

2. Generalized Iterative Speech Separation Pipelines

In this section we briefly overview three types of generalized iterative speech separation pipelines shown in Figure 1. We focus

http://dx.doi.org/10.21437/Interspeech.2021-1161

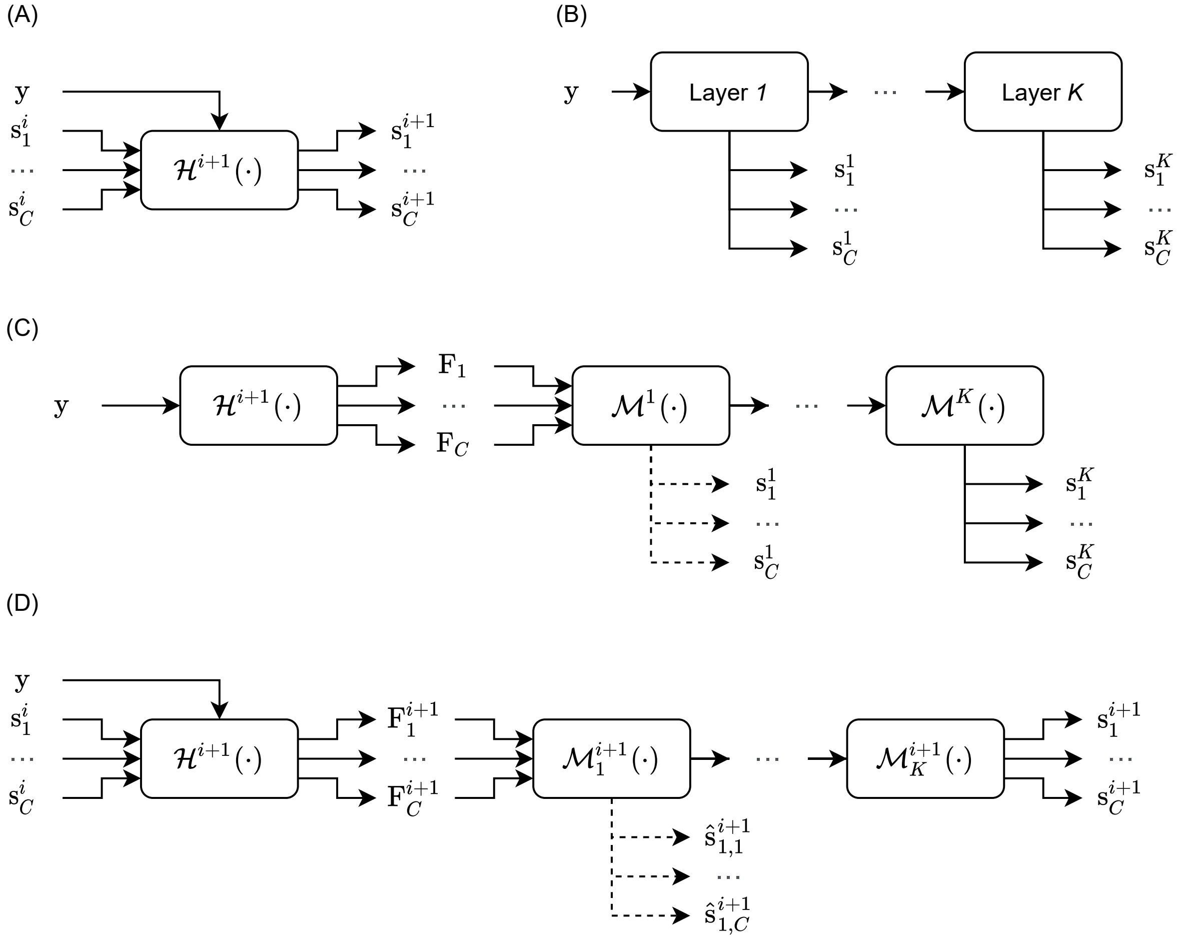

Figure 1: *Standard pipelines for iterative speech separation networks. (A) The separation outputs at iteration i is used as auxiliary inputs at iteration $i+1$. (B) An output layer is added to each layer in the network to generate intermediate separation outputs, and the training objective is applied to each of them. (C) A single-input-multi-output (SIMO) separation module first separates the mixture into either outputs or intermediate features, and K single-input-single-output (SISO) post-enhancement layers is further applied to each of the outputs to generate the targets. (D) The combination of the three aforementioned pipelines.*

our discussion on single-channel speech separation, although multi-channel systems can also be represented in a similar way.

2.1. Standard iterative separation

Figure 1 (A) shows the most common pipeline for iterative separation. The mixture signal $\mathbf{y}$, together with the separation output from i-th iteration $\{\mathbf{s}_c^i\}_{c=1}^C$, is passed to the $(i+1)$-th single-input-multi-output (SIMO) mapping $\mathcal{H}^{(i+1)}(\cdot)$ defined by a neural network, to generate the separation outputs at the $(i+1)$-th iteration $\{\mathbf{s}_c^{i+1}\}_{c=1}^C$. For the first iteration, $\{\mathbf{s}_c^0\}_{c=1}^C$ are initialized as zero signals. The output permutation at the i-th iteration is used as the input permutation at the $(i+1)-th$ iteration, and empirically we find that this can maintain the output permutation across all iterations without affecting the separation performance.

There are two optional designs in this pipeline. First, the model parameters across different iterations can be either shared or different. This can be related to the general definition of time-invariant and time-variant systems. Second, when performing backpropagation, the gradient of the input to the $(i+1)$-th iteration can either pass to the i-th iteration or be discarded. In the latter case, each iteration can be viewed as an independent process, and the output from the previous iteration can be treated as additional bias information for data augmentation.

2.2. Layer-wise training objective as iterative separation

Figure 1 (B) shows a typical generalized iterative separation pipeline where the training objective is applied to all the layers in the separator. For the k-th layer where $K = 1, \ldots, K$, a shared output layer is applied to its output to generate intermediate separation outputs $\{\mathbf{s}_c^k\}_{c=1}^C$. The pipeline is defined as a generalized iterative separation network because all layers in the separator are directly optimized with the same training objective to minimize the discrepancy between the (intermediate) separation outputs and the target sources, hence layer k with $k \geq 2$ can be treated as iterative separation layers receiving separation outputs from layer $k-1$. This is unlike the standard pipeline where the training objective is only applied to the output at layer K and the outputs at other layers do not have a clear and explicit meaning.

Table 1: *Experiment results for different configurations in the iterative separation pipeline.*

SIMO layers	SISO layers	Iteration	Effective network depth	Output detach	SI-SDR (dB)	
					Time domain	Freq domain
3	0	1	3	–	8.7	8.4
2	1				9.0	8.7
1	2				9.2	9.1
1	2	2	6	✓	9.8	9.6
				✗	9.7	9.5
2	4	1		–	10.0	9.5
1	2	3	9	✓	10.1	**9.8**
				✗	9.6	9.5
3	6	1		–	**10.2**	9.4

2.3. Pre-separation and post-enhancement as iterative separation

Figure 1 (C) shows the pipeline with a pre-separation module and a post-enhancement module. The pre-separation module receives the mixture as input and generates C outputs $\{\mathbf{F}_c\}_{c=1}^{C}$ for the C target sources. Note that $\{\mathbf{F}_c\}_{c=1}^{C}$ can either be intermediate features or the estimated targets themselves. Each feature $\mathbf{F}_c$ is then passed to a post-enhancement module with K SISO layers for output refinement, and the separation outputs can be generated from either each SISO layer (i.e., layer-wise objective) or the last SISO layer only. It is defined as a generalized iterative separation pipeline when the training objective is applied to each of the post-enhancement layers. Moreover, the training objective can also be applied to $\{\mathbf{F}_c\}_{c=1}^{C}$ when they correspond to the separation outputs.

2.4. Combined pipeline for generalized iterative separation

Figure 1 (D) shows a combined pipeline for the three pipelines above. The separation-enhancement pipeline is inserted into the standard iterative pipeline, where the SISO enhancement layers together with optional layer-wise objectives are jointly applied together with the SIMO separation module. The outputs at the k-th SISO enhancement layer at the $(i+1)$-th iteration become $\{\hat{\mathbf{s}}_{k,c}^{i+1}\}_{c=1}^{C}$.

3. Experiment Configurations

3.1. Dataset

We use the same dataset proposed in [28] where 20000, 5000, and 3000 two-speaker mixtures were simulated in a noisy reverberant environment for training, validation, and evaluation, respectively. All utterances were 4 second long and sampled at 16k Hz sample rate. The speech signals were randomly sampled from the 100-hour Librispeech subset [29], and the noise signals were sampled from the 100 Nonspeech Corpus [30]. The length and width of all the rooms were randomly sampled between 3 and 10 meters, and the height was randomly sampled between 2.5 and 4 meters. The reverberation time (T60) was randomly sampled between 0.1 and 0.5 seconds, and the corresponding room impulse responses were simulated by the image method [31] using the gpuRIR toolbox [32]. For other details of the dataset, we encourage the readers to refer to [28]. We only use the first channel in our experiments for single-channel separation.

3.2. Model architectures

We use the combined pipeline in Figure 1 and follow the model architecture in [27], where an encoder-separator-decoder design is applied for the separation framework and the dual-path RNN (DPRNN) [33] is selected as the basic building block. The separator estimates a set of multiplicative masks applied to the mixture's encoder output, and we follow the typical configuration where the masks are constrained to be nonnegative by a ReLU function. We evaluate both time-domain and frequency-domain models by selecting either learnable encoder and decoder or short-time Fourier transform (STFT) and its inverse (ISTFT). For STFT/ISTFT, we only use the magnitude spectrogram as the input and use the mixture's phase spectrogram for the ISTFT of the separation outputs. The window size for the learnable encoder/decoder and STFT/ISTFT is set to 2 ms (32 points) and 32 ms (512 points), respectively. The number of filters in the encoder and decoder for the learnable encoder/decoder and STFT/ISTFT is set to 128 and 257, respectively. A linear bottleneck layer with 64 hidden units is always applied to the encoder output for dimension reduction. The number of hidden units in each of the LSTM layers in the DPRNN modules is set to 128. The segment size for DPRNN is set to 100 frames and 24 frames for the learnable encoder/decoder and STFT/ISTFT, respectively.

3.3. Training and evaluation

All models are trained for 100 epochs with the Adam optimizer [34]. The learning rate is initialized to 0.001 and decayed by 0.98 for every two epochs. Gradient clipping by a maximum gradient norm of 5 is always applied. Early stopping is applied when no best validation model is found for 10 consecutive epochs. No other training tricks or regularization techniques are applied. Negative signal-to-noise ratio (SNR) together with permutation invariant training (PIT) [8] is used as the training objective for all models, and the training target is set to the reverberant clean signals. SI-SDR [35] is reported as the evaluation of the signal quality.

4. Results and Discussions

Table 1 shows the experiment results of the models with different configurations. We assume that the model parameters are shared across all iterations in the models. Each SIMO and SISO layer corresponds to a DPRNN block, and 0 SISO layers means that the output of the SIMO layers are the separation outputs. We first notice that for both time-domain and frequency-domain models, a deeper SISO module leads to better performance. Since [27] already showed that a deep SISO module improves the performance of time-domain networks, here we further confirm that frequency-domain networks can also benefit from this configuration. For iterative networks with the number of iterations larger than 1, we also compare them with one-pass models with a same effective network depth. The "output detach" column corresponds to the configuration where the gradient is constrained within each iteration (which can be implemented by *detach* function in Pytorch or *stop_gradient* function in Tensorflow). We observe that for the 2-iteration configuration, both time-domain and frequency-domain models have comparable performance with their corresponding one-pass models. Moreover, detaching the gradient from the previous output leads to a minor improvement. For the 3-iteration configuration, the improvement introduced by gradient detachment becomes more salient, and the frequency-domain model even outperform the one-pass counterpart. The results here show that the iterative separation pipeline can serve as an effective way to reduce the storage requirement of the separation networks.

Table 2: *Effect of layer-wise training objective.*

SIMO layers	SISO layers	Iteration	SI-SDR (dB)	
			Time domain	Freq domain
1	2	3	9.9	9.4
3	6	1	10.1	9.2

Table 2 provides the separation performance of the iterative systems with layer-wise objective in Section 2.4. The output at each SISO layer in each iteration is passed to the shared mask estimation layer to generate the separated waveforms, and the negative SNR objective is applied to all outputs in the entire pipeline. We use the configuration where output detachment is applied and parameters are shared across iterations. Comparing the results with the ones in Table 1, layer-wise objective does not further improve the performance in either one-pass or iterative configurations. [21] and [22] reported that layer-wise objective leads to a performance improvement in both monaural and binaural separation tasks, however here we observe that the objective may not be a universal option and its effect can vary in different problem settings and architectures. Since layer-wise objective belongs to the definition of a generalized iterative separation pipeline in our discussion, the results also show that the way the iterative separation is performed also needs to be carefully designed.

Table 3: *Effect of iteration-specific SIMO modules with STFT/ISTFT.*

SIMO layers	SISO layers	Iteration	SI-SDR (dB)
1	2	2	9.6
		3	9.7

Table 3 presents the separation performance on the models with different parameters in different iterations with STFT/ISTFT. We only evaluate the frequency-domain configuration. Here we use iteration-specific SIMO layers and still share the SISO layers across iterations. The rationale behind this configuration is that the additional bias information, i.e., the separation outputs from the previous iteration, is directly used by the SIMO separator, and the bias information differs from iteration to iteration. Iteration-specific separator may then have the potential to perform better separation based on the characteristics of the separation outputs from each iteration. However, we find that the performance obtained by iteration-specific SIMO layers is on par with that of shared SIMO layers. The results indicate that the use of iterative-specific model parameters, or more general, time-variant model parameters when we treat each iteration as a discrete time step, may require further investigation in the iterative separation pipelines.

Table 4 measures the effect of different inference iterations. For the models trained with 1, 2, and 3 iterations with 1 SIMO layer and 2 SISO layers, we evaluate their performance on the inference phase with 1 to 4 iterations. This experiment is conducted to look into the effect of a mismatched number of iterations in the training and inference phases. We observe that the model trained with 1 iteration completely fails when more than 1 iteration is applied in the inference phase, which is expected since the bias information starting from the second iteration is completely unseen for the SIMO separation. When the model is trained for no fewer than 2 iterations, the inference phase performance becomes stable even if the inference phase iteration is larger than the training phase iteration. Moreover, we also evaluate the performance of the frequency-domain model when the oracle bias information, i.e., the clean target sources, is used for the SIMO module, and an auxiliary training objective is added to the overall training objectives. We can see that adding this oracle bias information allows the model trained with 1 iteration to perform much better in inference phase and does not harm the performance of the models trained with 2 and 3 iterations. However, no obvious performance improvement is achieved by the auxiliary loss. How to further improve the separation performance of those iterative models remains an important topic to explore.

Table 4: *Effect of different number of inference iterations and oracle bias information.*

Training iterations	Inference iterations	SI-SDR (dB)		
		Time	Freq	Freq + oracle bias
1	1	9.2	9.1	9.1
	2	5.4	3.4	8.9
	3	6.2	4.5	8.6
	4	5.5	3.7	8.3
2	1	9.4	9.2	9.1
	2	9.8	9.6	9.7
	3	9.7	9.6	9.7
	4	9.7	9.6	9.6
3	1	9.2	9.2	9.2
	2	9.4	9.7	9.8
	3	10.1	9.8	9.8
	4	10.1	9.8	9.8

5. Conclusion

In this paper, we empirically evaluated the performance of various configurations in the generalized iterative speech separation pipeline. A generalized iterative speech separation pipeline was defined as a model that performed *multi-pass* separation where the separation outputs from a previous iteration can be used as additional bias information for the next iteration. We designed experiments on both time-domain and frequency-domain networks with different hyperparameter configurations, and observed that the iterative separation pipeline can be adopted as a storage-efficient pipeline with fewer network parameters without sacrificing the separation performance. We also found that it required a better understanding and analysis on the intrinsic mechanisms in the pipeline in order to further improve the separation performance and design better architectures.

6. Acknowledgments

This work was funded by a grant from the National Institute of Health, NIDCD, DC014279; and a grant from Marie-Josée and Henry R. Kravis.

7. References

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

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

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

[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] S.-W. Fu, T.-W. Wang, Y. Tsao, X. Lu, and H. Kawai, "End-to-end waveform utterance enhancement for direct evaluation metrics optimization by fully convolutional neural networks," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 9, pp. 1570–1584, 2018.

[6] 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 (TASLP)*, vol. 27, no. 8, pp. 1256–1266, 2019.

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

[8] 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 (TASLP)*, vol. 25, no. 10, pp. 1901–1913, 2017.

[9] Y.-H. Tu, J. Du, L. Sun, F. Ma, H.-K. Wang, J.-D. Chen, and C.-H. Lee, "An iterative mask estimation approach to deep learning based multi-channel speech recognition," *Speech Communication*, vol. 106, pp. 31–43, 2019.

[10] I. Kavalerov, S. Wisdom, H. Erdogan, B. Patton, K. Wilson, J. L. Roux, and J. R. Hershey, "Universal sound separation," *arXiv preprint arXiv:1905.03330*, 2019.

[11] E. Tzinis, S. Wisdom, J. R. Hershey, A. Jansen, and D. P. Ellis, "Improving universal sound separation using sound classification," in *Acoustics, Speech and Signal Processing (ICASSP), 2020 IEEE International Conference on*. IEEE, 2020, pp. 96–100.

[12] M. I. Mandel, R. J. Weiss, and D. P. Ellis, "Model-based expectation-maximization source separation and localization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 18, no. 2, pp. 382–394, 2009.

[13] J. Le Roux, J. R. Hershey, and F. Weninger, "Deep NMF for speech separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 66–70.

[14] S. Wisdom, T. Powers, J. Pitton, and L. Atlas, "Deep recurrent nmf for speech separation by unfolding iterative thresholding," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2017, pp. 254–258.

[15] Z. Chen, X. Xiao, T. Yoshioka, H. Erdogan, J. Li, and Y. Gong, "Multi-channel overlapped speech recognition with location guided speech extraction network," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 558–565.

[16] Z.-Q. Wang, H. Erdogan, S. Wisdom, K. Wilson, D. Raj, S. Watanabe, Z. Chen, and J. R. Hershey, "Sequential multi-frame neural beamforming for speech separation and enhancement," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 905–911.

[17] J. Wang, J. Chen, D. Su, L. Chen, M. Yu, Y. Qian, and D. Yu, "Deep extractor network for target speaker recovery from single channel speech mixtures," *Proc. Interspeech*, pp. 307–311, 2018.

[18] P. Wang, Z. Chen, X. Xiao, Z. Meng, T. Yoshioka, T. Zhou, L. Lu, and J. Li, "Speech separation using speaker inventory," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 230–236.

[19] N. Zeghidour and D. Grangier, "Wavesplit: End-to-end speech separation by speaker clustering," *arXiv preprint arXiv:2002.08933*, 2020.

[20] P. Wang, Z. Chen, D. Wang, J. Li, and Y. Gong, "Speaker separation using speaker inventories and estimated speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, 2020.

[21] E. Nachmani, Y. Adi, and L. Wolf, "Voice separation with an unknown number of multiple speakers," in *International Conference on Machine Learning*. PMLR, 2020, pp. 7164–7175.

[22] K. Tan, B. Xu, A. Kumar, E. Nachmani, and Y. Adi, "SAGRNN: Self-attentive gated RNN for binaural speaker separation with interaural cue preservation," *IEEE Signal Processing Letters*, 2020.

[23] H. Erdogan, J. R. Hershey, S. Watanabe, M. I. Mandel, and J. Le Roux, "Improved MVDR beamforming using single-channel mask prediction networks." in *Proc. Interspeech*, 2016, pp. 1981–1985.

[24] Y. Xu, J. Du, Z. Huang, L.-R. Dai, and C.-H. Lee, "Multi-objective learning and mask-based post-processing for deep neural network based speech enhancement," *arXiv preprint arXiv:1703.07172*, 2017.

[25] Y. Luo, E. Ceolini, C. Han, S.-C. Liu, and N. Mesgarani, "FaSNet: Low-latency adaptive beamforming for multi-microphone audio processing," in *Automatic Speech Recognition and Understanding (ASRU), 2019 IEEE Workshop on*. IEEE, 2019.

[26] C. Fan, J. Tao, B. Liu, J. Yi, Z. Wen, and X. Liu, "End-to-end post-filter for speech separation with deep attention fusion features," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 28, pp. 1303–1314, 2020.

[27] Y. Luo, Z. Chen, C. Han, C. Li, T. Zhou, and N. Mesgarani, "Rethinking the separation layers in speech separation networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2021 IEEE International Conference on*. IEEE, 2021.

[28] Y. Luo, Z. Chen, N. Mesgarani, and T. Yoshioka, "End-to-end microphone permutation and number invariant multi-channel speech separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2020 IEEE International Conference on*. IEEE, 2020, pp. 6394–6398.

[29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5206–5210.

[30] G. Hu, "100 Nonspeech Sounds," http://web.cse.ohio-state.edu/pnl/corpus/HuNonspeech/HuCorpus.html.

[31] 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.

[32] D. Diaz-Guerra, A. Miguel, and J. R. Beltran, "gpuRIR: A python library for room impulse response simulation with gpu acceleration," *Multimedia Tools and Applications*, pp. 1–19, 2020.

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

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

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