INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

A Unified Framework for Low-Latency Speaker Extraction in Cocktail Party Environments

Yunzhe Hao[1,2], *Jiaming Xu*[1,2†], *Jing Shi*[1,2], *Peng Zhang*[1,2], *Lei Qin*[4], *Bo Xu*[1,2,3†]

[1]Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing
[2]University of Chinese Academy of Sciences, Beijing
[3]Center for Excellence in Brain Science and Intelligence Technology, CAS, Beijing
[4]Huawei Consumer Business Group

{haoyunzhe2017,jiaming.xu,shijing2014,zhangpeng2018,xubo}@ia.ac.cn, qinlei9@huawei.com

Abstract

Speech recognition technology in single-talker scenes has matured in recent years. However, in noisy environments, especially in multi-talker scenes, speech recognition performance is significantly reduced. Towards cocktail party problem, we propose a unified time-domain target speaker extraction framework. In this framework, we obtain a voiceprint from a clean speech of the target speaker and then extract the speech of the same speaker in a mixed speech based on the previously obtained voiceprint. This framework uses voiceprint information to avoid permutation problems. In addition, a time-domain model can avoid the phase reconstruction problem of traditional time-frequency domain models. Our framework is suitable for scenes where people are relatively fixed and their voiceprints are easily registered, such as in a car, home, meeting room, or other such scenes. The proposed global model based on the dual-path recurrent neural network (DPRNN) block achieved state-of-the-art under speaker extraction tasks on the WSJ0-2mix dataset. We also built corresponding low-latency models. Results showed comparable model performance and a much shorter upper limit latency than time-frequency domain models. We found that performance of the low-latency model gradually decreased as latency decreased, which is important when deploying models in actual application scenarios.

Index Terms: cocktail party problem, speaker extraction, low-latency, time domain

1. Introduction

Automatic speech recognition (ASR) systems have achieved impressive results for single-talker speech recognition tasks. However, such systems remain unsatisfactory when under complex auditory scenes, especially in noisy and multi-talker environments, i.e., the so-called cocktail party problem [1, 2].

Many researchers have attempted to solve this issue using traditional methods, such as computational auditory scene analysis (CASA) [3] and non-negative matrix factorization (NMF) [4, 5], or deep learning technology, such as deep clustering (DPCL) [6], deep attractor network (DANet) [7], permutation invariant training (PIT) [8], time-domain audio separation network (TasNet) [9, 10], Wavesplit [11] and VoiceFilter [12]. DPCL maps time-frequency features into high-dimensional space and then uses the clustering algorithm to cluster the mixed speech features into several speakers. DANet obtains each speaker's voiceprint feature vector from mixed speech, which is then used as the center point for clustering. Excluding DPCL and DANet, more models use a spectral mask-based method, i. e., they first calculate the short-time Fourier transform (STFT) features of the mixed speech [13], then generate a spectral mask from the amplitude spectrum features, and use the phase spectrum of mixed speech to reconstruct a clean signal.

Researchers hope that the order of model output does not affect performance. Yu et al. [8] first proposed PIT technology to minimize reconstruction errors under minimum energy ranking. Various models used PIT afterwards [10, 14, 15, 16]. However, the specific number of speakers in mixed speech must be precise and known in advance, and the speaker label of the output channel cannot be obtained directly [7], i. e., permutation problem [8]. Under actual application scenes, we usually do not know the specific number of speakers and are not necessarily interested in everyone, rather a small number of people in the crowd. Therefore, the speaker extraction model may be more suitable for scenes at a cocktail party. Xu et al. [17] utilized a reference speech to extract voiceprints to help track a target speaker, and then VoiceFilter [12] trained voiceprint network with a large speech dataset. However, VoiceFilter did not optimize the voiceprint network jointly with the overall model, which may degrade performance. Xu et al. [18] optimized the voiceprint network and speaker extraction network jointly, but performance was not better than the speech separation model [10].

On the other hand, actual application scenarios have higher requirements for the separation model's real-time processing capability, e. g., front-end of ASR and hearing aids. However, the upper limit of real-time processing of time-frequency domain models, e. g., SBF-MTSAL-Concat [18] and VoiceFilter [12], are limited by the STFT frame length. That is, the time-frequency domain model's ideal latency is equal to the STFT frame length, 32 ms in general. Although Wang et al. [19] reduced the model delay by shortening the STFT frame length from 32 ms to 8 ms, performance degradation was severe. Luo et al. [10] proposed a time-domain model (TasNet) that replaced STFT and inverse STFT with a trained encoder and decoder pair. Instead of mapping waveforms into the time-frequency domain, the encoder maps them into a new temporal resolution space. This coding method improves modeling granularity, removes the constraints between time precision and feature dimension size in STFT, and avoids phase reconstruction problems of time-frequency domain models [20, 21]. Recently, time-domain speech separation models, such as the dual-path recurrent neural network (DPRNN) [22] and Wavsplit [11], have obtained the new state-of-the-art on the benchmark dataset for speech separation task. Xu et al. [23] also proposed a time-

†Corresponding author

http://dx.doi.org/10.21437/Interspeech.2020-2085

domain model, SpEx, for speaker extraction task. However, SpEx uses future information to predict the current time step mask, i. e., not real-time, and its performance is worse than ours.

In this paper, we propose a unified framework for speaker extraction. The framework aims to use a target speaker's reference speech to obtain a voiceprint clue, which is then used to extract the target speaker's speech from noisy speech. The novelty of this paper includes the following:

1. We proposed a unified framework based on voiceprint for speaker extraction tasks, which can be easily upgraded by updating submodules. We built the global models and low-latency models under the framework. The framework worked well under the different settings and exhibited robustness;
2. Our model achieved state-of-the-art results on benchmark datasets for speaker extraction tasks with fewer parameters;
3. We used interfering speech as a supervision signal in the speaker extraction paradigm, which helped separate the target speaker's speech and improved model performance.

This paper is organized as follows. The proposed speaker extraction framework and specific models are introduced in detail in Sec. 2. Experimental evaluation is presented in Sec. 3. Finally, Sec. 4 summarizes this paper.

2. Proposed framework

Our proposed framework is composed of four main modules: i. e., voiceprint encoder, speech encoder, speech decoder, and target speaker extraction module, as shown in Fig. 1. We first input mixed speech composed of the target and interfering speakers' speech, and another reference speech of the target speaker into the framework. The voiceprint encoder processes the reference speech to generate the target speaker's voiceprint. The speech encoder processes the mixed speech to obtain mixed speech features. The voiceprint and mixed speech features are taken as inputs for the speaker extraction module. With the target speaker's voiceprint as a clue, the module tracks and extracts the target speaker's feature mask in the mixed speech features. After that, the mask multiplies the mixed speech features to obtain the features of the target speaker. Finally, the speech decoder decodes the target speaker's features to generate the target speaker's pure voice.

2.1. Voiceprint encoder

Firstly, STFT is performed on the reference speech to obtain time-frequency features. Two layers of bidirectional long short-term memory (BiLSTM) are then adopted to process the magnitude of time-frequency features and perform mean-pooling to compress the time dimension and obtain a vector. The linear layer performs a dimensional transformation on the vector to match the feature size of the mixed speech. This vector is then used as the voiceprint of the target speaker.

2.2. Speech encoder and decoder

Time-frequency domain coding, e. g., STFT, and time-domain coding are commonly used methods in speech separation and speaker extraction models. Time-domain coding has many advantages over time-frequency domain coding, such as trainable parameters, shorter frame length, and no phase reconstruction problem [20]. Therefore, we adopted time-domain coding here, as seen in Fig. 1. We used convolutional layers and transposed convolutional layers as the encoder and decoder, respectively. The encoder encodes waveforms into speech features, and the decoder reconstructs waveforms from speech features.

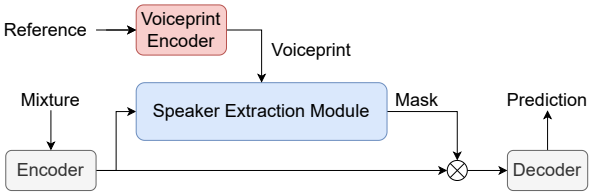

Figure 1: *Unified speaker extraction framework.*

2.3. Speaker extraction module

A speaker extraction module should possess strong time-series data processing capabilities. We adopted a deep dilated temporal convolutional network (TCN) [24, 25] and DPRNN as the main structure of the module, which have been proven effective in speech separation tasks [10, 22].

Similar to the separation module in Conv-TasNet [10], our TCN speaker extraction module consists of stacked one-dimensional (1-D) dilated convolutional blocks, i. e., TCN blocks, as shown in Fig. 2. We first normalize speech features and transform their dimensions. After it, there are several layers of TCN block. We multiply voiceprint and speech features for input into each TCN block. The output of each TCN block is then summarized and multiplied by voiceprint and processed by the parametric rectified linear unit (PReLU), pointwise convolution (1×1 Conv), and sigmoid activation function to obtain the target speaker's mask $mask_{target}$. The mask multiplies mixed speech features to obtain features of the target speaker. Finally, the speech decoder decodes clean target speech. The complement of $mask_{target}$ is then used as the interfering speaker's mask only during the training phase to improve performance. The interfering speaker's mask can be defined as:

$$mask_{interfering} = M - mask_{target}, \tag{1}$$

where M is a matrix with the same shape as $mask_{target}$ and all elements are 1.

Specifically, the TCN block consists of several components connected in a series, i. e., 1×1 Conv, PReLU, normalization, depthwise convolution (D-Conv), PReLU, normalization, and 1×1 Conv, as shown in Fig. 4(A). The TCN block has two output paths. One is used as the input of the next TCN block, and the other is summarized with the other TCN block outputs as the mask. There are residual connections within the TCN block. The two 1×1 Conv components integrate information at the channel dimension. The D-Conv performs convolution on the time dimension of each channel. There are three TCN block groups in this module and eight TCN blocks in each group. The dilation factor of each group gradually increases with an exponent of 2 as the network deepens, i. e., 1, 2, 4, ..., 128, allowing the convolution receptive field to expand gradually for more information. At the beginning of each group, we reset the dilation factors to 1. The speaker extraction module forms a fine-rough-fine-rough style.

Inspired by [22], we utilized DPRNN as the speaker extraction module, as seen in Fig. 3. After normalization and

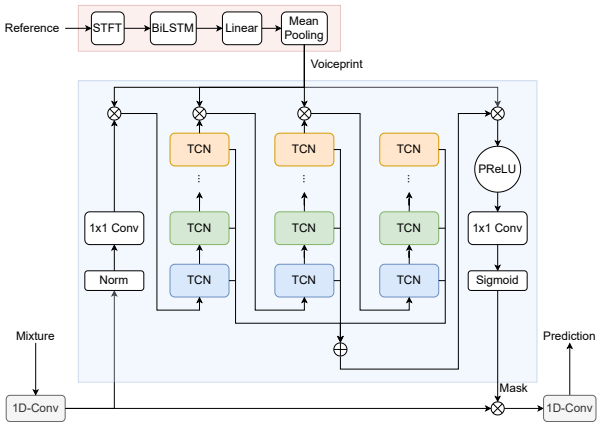

Figure 2: *Speaker extraction model based on TCN blocks. Different colors in TCN blocks denote different dilation factors.*

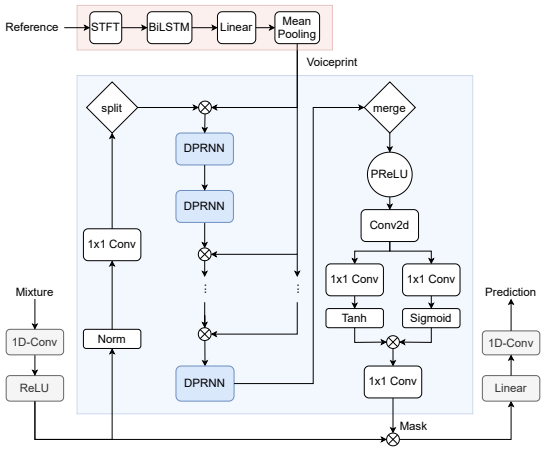

Figure 3: *Speaker extraction model based on DPRNN block.*

1×1 Conv, the speech features are split into fixed length chunks. The obtained chunks are spliced in parallel to form a two-dimensional (2-D) structure. Similar to TCN, our DPRNN separation module is composed of stacked DPRNN blocks. The voiceprint and speech features are multiplied as inputs into the DPRNN blocks at intervals of 1. Each DPRNN block consists of several components connected in a series, i. e., intra-chunk LSTM, linear, normalization, inter-chunk LSTM, linear and normalization, as displayed in Fig. 4(B). There are also residual connections in the DPRNN block. Specifically, the intra-chunk LSTM extracts features within chunks on a short time scale, and inter-chunk LSTM extracts features between chunks on a long time scale. Reasonable setting of chunk length and speech length can ensure that each LSTM processes a sequence that is not too long, which enhances the model's ability to process long sequences and increases the upper limit of modeling precision. We used nine DPRNN block groups, with each group consisting of one intra-chunk LSTM and inter-chunk LSTM to form a fine-rough-fine-rough style.

2.4. Loss function

The traditional speech separation models usually use the mean square error (MSE) between the predicted time-frequency mask and actual mask as a training goal. Recently, Luo et al. [10, 22] utilized the scale-invariant source-to-noise ratio (SI-SNR) between predicted waveforms and the target to optimize the SI-SNR value. Here, we used SI-SNR loss for training. We also reconstructed the interfering speech to improve performance. The SI-SNR can be defined as:

$$\begin{cases} s_{target} = \frac{\langle \hat{s}, s \rangle s}{\|s\|^2}; \\ e_{noise} = \hat{s} - s_{target}; \\ SI-SNR = 10\log_{10} \frac{\|s_{target}\|^2}{\|e_{noise}\|^2}, \end{cases} \tag{2}$$

where $\hat{s}$ and s are the predicted speech and target speech, respectively, and $\langle s, s \rangle$ and $\|s\|^2$ are the signal power.

2.5. Low-latency capability

Our framework can meet real-time requirements by following simple modifications. Certain occasions are easy to register and restore voiceprints, such as in a car, at home, or in a meeting room. We can directly use the retained voiceprints for registered speakers and skip the voiceprint encoder. The speech encoder and decoder generally adopt a window length of 2 ms or even less, which is significantly shorter than the time-frequency domain model, 32 ms in general. Therefore, our low latency upper limit is much shorter than the time-frequency domain model. The real-time capability of the speaker extraction module is critical. We adopt causal D-Conv in each TCN block, whose convolution receptive field only covers historical information, and replace all global normalization functions with a causal normalization function, i. e., cumulative layer normalization (cLN):

$$\begin{cases} cLN(f_k) = \frac{f_k - E[f_{t\leq k}]}{\sqrt{Var[f_{t\leq k}]+\epsilon}} \odot \gamma + \beta; \\ E[f_{t\leq k}] = \frac{1}{Nk} \sum_{Nk} f_{t\leq k}; \\ Var[f_{t\leq k}] = \frac{1}{Nk} \sum_{Nk} (f_{t\leq k} - E[f_{t\leq k}])^2, \end{cases} \tag{3}$$

where f_k is the k-th frame of the entire feature F, $f_{t\leq k}$ corresponds to the feature of k frames $[f_1, \ldots, f_k]$, and γ, β are trainable parameters applied to all frames. Based on the above modifications, our model is entirely causal. The theoretical delay only depends on the delay of the speech encoder and decoder module. If we reserve non-causal D-Conv in the underlying TCN blocks and use causal D-Conv in the upper TCN blocks, our model can utilize certain future information, resulting in a slight increase in model performance. As a cost, the theoretical delay will increase. Furthermore, by setting inter-chunk LSTM as unidirectional, using causal normalization, we can also achieve a low-latency model based on DPRNN. We can modify the size of the chunk to adjust the length of the delay. We can achieve different low-latency models to balance the trade-off between performance and real-time according to the real-time requirements of specific application scenarios.

3. Experiments and results

3.1. Dataset and experiment settings

We evaluated model performance on the speech separation benchmark dataset WSJ0-2mix [6]. Our training data preparation method was similar to that of VoiceFilter [12]. We randomly selected two speakers and set the first as the target speaker and the other as the interfering speaker by default. We then randomly selected two speeches from the target speaker and one speech from the interfering speaker, and prepared them

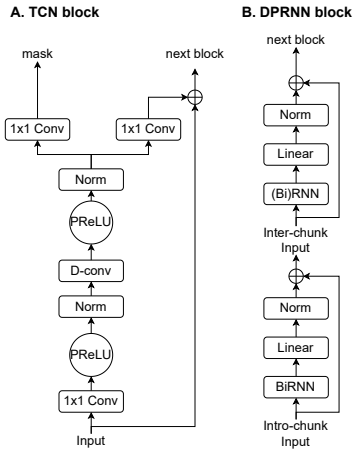

Figure 4: *A: Details of TCN block. 'Norm' is global layer normalization in the global model and cLN in low-latency model. 'D-Conv' indicates a depthwise convolution. B: Details of DPRNN block. '(Bi)RNN' is a BiLSTM in global model and a unidirectional LSTM in low-latency model.*

as training triples, namely target speech, reference speech, and interfering speech. Speeches were cuted at random location and divided into two segments, which were then reversed back and forth and combined again to augment training data. The length of target and interfering speech was limited to 6 seconds. If it exceeds 6 seconds, we used the first 6 seconds; if this was not enough, we extended it with zeros. The speech mixtures were generated by mixing them at random signal-to-noise ratios (SNR) between -2.5 dB and 2.5 dB. We performed STFT on the reference speech, with a hanning window length of 32 ms and frameshift of 8 ms. In the test phase, we made corresponding modifications to the WSJ0-2mix test set for speaker extraction task. Specifically, we set one speaker as the target the other as the interfer. We then mixed the target speech and interfering speech according to the given SNR without additional processing. We used another speech of the target speaker in test set as a reference speech. In this way, one original test speech will generate two test speeches with different target speakers. We downsampled all speeches to 8 kHz.

We trained the networks until the performance of the evaluation set did not improved in 5 consecutive epochs. Adam [26] was used as the optimizer. Gradient clipping with a maximum L2-norm of 5 is applied during training. The learning rate was set to $1e^{-3}$. We built our model in Pytorch.

3.2. Global model performance

Our global model performed significantly better than other models in the speaker extraction task on the WSJ0-2mix dataset, as seen in Tab. 1. It should be noted that although the best signal-to-distortion ratio (SDR) reported by SpEx is 15.9 dB, this is the result of a 60 seconds reference speech spliced using multiple reference speeches. When using only one reference speech, like ours, its performance is 15.1 dB. Besides, SpEx uses a multi-scale encoding method with a maximum window length of 20 ms, which severely reduces the low latency potential of the model. Although VoiceFilter reported a very competitive performance, i. e., 17.9 dB SDR in their test set (i. e., LibriSpeech) [12], its SDR improvement (SDRi) is 7.8 dB, which is caused by clean signals mixed with silent parts of interference signals.

Table 1: *SDR with different speaker extraction methods based on the WSJ0-2mix dataset. '*' indicates that a 60 seconds reference speech was used.*

Methods	#Params	SDR(dB)
SBF [27]	19.3M	6.48
SBF-MTSAL [18]	19.3M	10.36
SBF-MTSAL-Concat [18]	8.9M	11.39
SpEx [23]	10.8M	15.1
SpEx* [23]	10.8M	15.6
ours using TCN	7.4M	16.19
ours using DPRNN	6.3M	**17.44**

3.3. Low-latency model performance

Our low-latency models showed good performance, as shown in Tab. 2. We compared the performance of TCN under different latencies. Unsurprisingly, the performance of the model gradually decreased with decreasing latency. Lower latency means less future information is used. We also compared the performance of DPRNN with 100 ms latency and found that the performance degradation was not severe. In application scenarios with different latency requirements, our framework could easily balance latency and performance.

Table 2: *SDR improvements (dB) with different low latencies on the WSJ0-2mix dataset. Frame length of following models is 2 ms.*

Block used	Low-latency	SDR(dB)	SDRi(dB)
TCN	global	16.19	16.06
TCN	0 ms	12.19	12.07
TCN	1 ms	12.21	12.08
TCN	7 ms	12.35	12.19
TCN	15 ms	12.51	12.40
DPRNN	global	17.44	17.30
DPRNN	100 ms	16.77	16.62

4. Conclusions

We proposed a unified time-domain speaker extraction framework and built a variety of time-domain models under this framework. The global models based on TCN and DPRNN blocks all surpassed current state-of-the-art results in the speaker extraction tasks using WSJ0-2mix dataset. We then built and evaluate corresponding low-latency models. Results showed comparable performance with much shorter upper limit latency than the time-frequency domain models. We also found that low-latency model's performance gradually decreases as latency decreases, which is important for deploying models in actual application scenarios.

5. Acknowledgments

This work was supported by the Major Project for New Generation of AI (Grant No. 2018AAA0100400), and the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDB32070000).

6. References

[1] E. C. Cherry, "Some experiments on the recognition of speech, with one and with two ears," *Journal of the Acoustical Society of America*, vol. 25, no. 5, pp. 975–979, 1953.

[2] M. Elhilali, *Modeling the Cocktail Party Problem*. Cham: Springer International Publishing, 2017, pp. 111–135.

[3] G. J. Brown and M. Cooke, "Computational auditory scene analysis," *Computer Speech & Language*, vol. 8, no. 4, pp. 297–336, 1994.

[4] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *International Conference on Neural Information Processing Systems*, 2000.

[5] M. N. Schmidt and R. K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," in *ISCA International Conference on Spoken Language Proceesing (INTERSPEECH)*, 2006.

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

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

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

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

[10] L. Yi and M. Nima, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

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

[12] Q. Wang, H. Muckenhirn, K. Wilson, P. Sridhar, Z. Wu, J. R. Hershey, R. A. Saurous, R. J. Weiss, Y. Jia, and I. L. Moreno, "VoiceFilter: Targeted Voice Separation by Speaker-Conditioned Spectrogram Masking," in *ISCA International Conference on Spoken Language Proceesing (INTERSPEECH)*, 2019, pp. 2728–2732.

[13] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.

[14] Z. Shi, H. Lin, L. Liu, R. Liu, J. Han, and A. Shi, "Deep attention gated dilated temporal convolutional networks with intra-parallel convolutional modules for end-to-end monaural speech separation," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2019-Septe, pp. 3183–3187, 2019.

[15] M. Kolbaek, D. Yu, Z. H. Tan, J. Jensen, M. Kolbaek, 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.

[16] Y. Luo and N. Mesgarani, "Real-time single-channel dereverberation and separation with time-domain audio separation network." in *ISCA International Conference on Spoken Language Proceesing (INTERSPEECH)*, 2018, pp. 342–346.

[17] J. Xu, J. Shi, G. Liu, X. Chen, and B. Xu, "Modeling attention and memory for auditory selection in a cocktail party environment," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, 2018, pp. 2564–2571.

[18] C. Xu, W. Rao, E. S. Chng, and H. Li, "Optimization of speaker extraction neural network with magnitude and temporal spectrum approximation loss," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[19] S. Wang, G. Naithani, and T. Virtanen, "Low-latency deep clustering for speech separation," in *Icassp IEEE International Conference on Acoustics*, 2019.

[20] F. Bahmaninezhad, J. Wu, R. Gu, S. X. Zhang, Y. Xu, M. Yu, and D. Yu, "A comprehensive study of speech separation: Spectrogram vs waveform separation," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2019-Septe, pp. 4574–4578, 2019.

[21] Z.-Q. Wang, K. Tan, and D. Wang, "Deep learning based phase reconstruction for speaker separation: A trigonometric perspective," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 71–75.

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

[23] C. Xu, W. Rao, E. S. Chng, and H. Li, "Spex: Multi-scale time domain speaker extraction network," *arXiv preprint arXiv:2004.08326*, 2020.

[24] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks: A unified approach to action segmentation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9915 LNCS, pp. 47–54, 2016.

[25] C. Lea, M. D. F. Ren, A. Reiter, and G. D. Hager, "Temporal Convolutional Networks for Action Segmentation and Detection," pp. 156–165, 2016.

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

[27] M. Delcroix, K. Zmolikova, K. Kinoshita, A. Ogawa, and T. Nakatani, "Single channel target speaker extraction and recognition with speaker beam," in *ICASSP 2018 - 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.