EFFICIENT MONAURAL SPEECH SEPARATION WITH MULTISCALE TIME-DELAY SAMPLING

Shuangqing Qian[1], *Lijian Gao*[1], *Hongjie Jia*[1], *Qirong Mao*[1,2,*]

[1] School of Computer Science and Communication Engineering, Jiangsu University, China
[2] Jiangsu Engineering Research Center of Big Data
Ubiquitous Perception and Intelligent Agriculture Applications, China
[*] Corresponding Author: mao_qr@ujs.edu.cn

ABSTRACT

Recently, the segmented sample-level modeling approach based on Dual-Path Recurrent Neural Network (DPRNN) has been proved to be effective in Monaural Speech Separation (MSS). Many dual-path networks such as Dual-Path Transformer Network (DPTNet), with a series of improvements to DPRNN, have also improved the separation performance since these methods are effective to process long sequences. However, the receptive fields of these methods are fixed during local and global features learning, which makes it difficult to capture different scale local and global information in long sequences. In this paper, we propose a novel Multiscale Time-Delay Sampling method (MTDS) for the dual-path networks in MSS to learn sequence features from fine to coarse by multiscale time-delay sampling, which effectively integrates different scale local and global information for long sequences. Our experiments on the notable benchmark WSJ0-2mix data corpus result in 21.7dB SDRi and 21.5dB SI-SNRi, which obviously outperforms the state-of-the-arts without data augmentation.

Index Terms— Monaural speech separation, Multiscale time-delay sampling, Dual-path networks

1. INTRODUCTION

Monaural Speech Separation (MSS) is a fundamental task in signal processing with a wide range of real-world applications, such as voice assistants, hearing aids and so on. In recent years, MSS methods based on deep learning have developed rapidly [1, 2] with the popularity of deep learning. And these methods can be broadly divided into two categories: time-frequency domain separation and time domain separation. The first category methods use Short-Time Fourier Transform (STFT) to extract time-frequency features for separation, and then reconstruct the source waveforms by inverse STFT [3, 4, 5, 6]. The second category methods directly model the mixture waveform using an encode-decoder framework, which overcomes the upper bound on the accuracy of the reconstructed waveforms in time-frequency domain separation [7], and achieve better performance [8, 9, 10, 11, 12, 13]. Thus, in this paper, we focus on the dominate time domain separation.

Recently, for the category of time domain separation, many dual-path networks have been proposed to improve the separation performance, such as Dual-Path Recurrent Neural Network (DPRNN), Dual-Path Transformer Network (DPTNet) and so on [14, 15, 16, 17]. The main idea of dual-path network is first dividing the long sequence into segments, and then iteratively learning local features within segments and global features between segments. Although these approaches are good at processing long sequences, the segment length of division is fixed, which makes the receptive fields also fixed during local and global features learning. To solve this problem, we propose Multiscale Time-Delay Sampling (MTDS) for sequence features learning from fine to coarse. Specifically, MTDS gradually expands the receptive fields with the same time step through time-delay sampling, and effectively integrates the sequence features at different scales from fine to coarse during sequence features learning.

The contributions of this paper are summarized as follows: (I) We propose the MTDS method for dual-path networks in MSS to effectively learn sequence features at different scales from fine to coarse. (II) The proposed MTDS can be introduced to any dual-path networks in MSS to learn more sufficient local and global features from long sequences. (III) The experiment results show that our method outperforms the state-of-the-arts without data augmentation (21.7 dB SDRi and 21.5dB SI-SNRi on the public WSJ0-2mix data corpus).

2. MULTISCALE TIME-DELAY SAMPLING FOR DUAL-PATH NETWORK IN MSS

2.1. The Proposed Overall Structure

In this paper, we propose a novel multiscale time-delay sampling method for the dual-path network in MSS. The architecture of our system is called MTDS, as shown in Figure 1. It mainly consists of Encoder, Dual-Path Network, Time-Delay

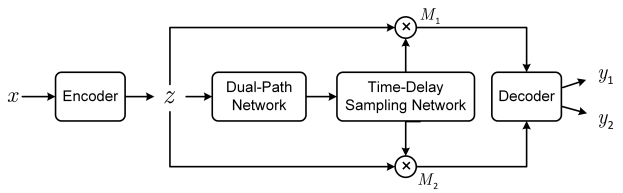

Fig. 1. The overall architecture of our system, which mainly consists of Encoder, Dual-Path Network, Time-Delay Sampling Network (TDSN) and Decoder.

Sampling Network (TDSN) and Decoder. And TDSN is the key component following the Dual-Path Network in MTDS.

Encoder: In order to extract speech features, the encoder takes the given waveform mixture $x \in \mathbb{R}^{1\times L}$ as input, and outputs a latent representation $z \in \mathbb{R}^{T\times N}$, where L denotes the length of speech, N is the feature dimension and T is the number of time steps. Specifically, encoder is a 1-D convolutional layer with a kernel size of W and a stride of $W/2$.

Dual-Path Network: In order to learn features from long sequences, the output of encoder z, followed as [14, 17], is divided into many overlapping chunks with length K, the overlapping ratio of which is 50%. And the first and last chunks are zero-padded to generate R chunks with the same size. Then all the chunks are concatenated to be a 3-D tensor $v \in \mathbb{R}^{N\times K\times R}$.

The 3-D tensor v is input into the dual-path network to learn intra-chunk features and inter-chunk features. More specifically, intra-chunk features learning is conducted on the chunk length dimension K of v, and inter-chunk features learning is conducted on the dimension of the number of chunks R to summarize the feature information from all chunks. And the dual-path network will be stacked B times.

Time-Delay Sampling Network: To learn the sequence features from fine to coarse, the output of dual-path network $\tilde{v} \in \mathbb{R}^{N\times K\times R}$ is fed into Time-Delay Sampling Network (TDSN). We stack Q TDSN, and each TDSN consists of four main operations as shown in Figure 2: sequence recombination, time-delay sampling, sequence features learning, and sequence restoration. We will give the details of each component in the following subsection 2.2. The main idea of TDSN is to expand the receptive fields with the same time step during features learning. And the time-delay sampling rate in the stacked Q TDSN increases exponentially, i.e., the q-th TDSN has a time-delay sampling rate of 2^{q-1}. Note that the receptive fields of TDSN are positively correlated with the time-delay sampling rate, and when we increase the rate exponentially, the receptive fields also increase exponentially. This design allows the model to obtain larger receptive fields in deeper layers, and integrates sequence features at different scales for fine to coarse sequence features learning.

The output of the last TDSN $D_Q \in \mathbb{R}^{N\times K\times R}$ is input into a 2-D convolution layer to learn a mask for each source. The

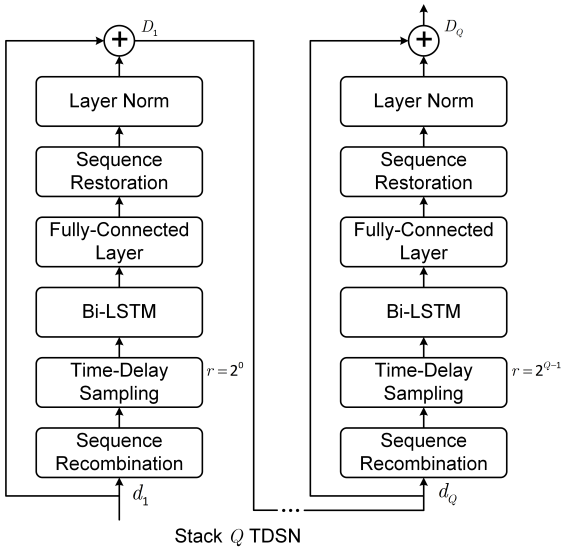

Fig. 2. The structure of stacked TDSN, and r denotes the time-delay sampling rate.

masks are then transformed back into sequence by overlap-add, and masked encoder features for each source are obtained by the element-wise multiplication between the mask and the encoder output.

Decoder: In the decoder, a transposed convolution layer is used to reconstruct separated speech signal for s-th source:

$$y_s = M_s * V \tag{1}$$

where $M_s \in \mathbb{R}^{T\times N}$ is the masked encoder features, $V \in \mathbb{R}^{N\times 1}$ is the parameter of the transposed convolution layer, and $y_s \in \mathbb{R}^{1\times L}$ denotes the separated s-th source.

2.2. Time-Delay Sampling Network

The TDSN contains four main operations: sequence recombination, time-delay sampling, sequence features learning, and sequence restoration. The specific flowchart of these operations is shown in Figure 3. Let us first denote the input of the q-th TDSN as $d_q \in \mathbb{R}^{N\times K\times R}$. The sequence recombination operation is based on the current time-delay sampling rate 2^{q-1} to reshape d_q into $\tilde{d}_q \in \mathbb{R}^{N\times 2^{q-1}K\times R'}$. In fact, this operation takes the R chunks of length K in d_q and splices them once every 2^{q-1} chunks, in order to obtain R' chunks of length $2^{q-1}K$. Note that R may not be divisible by 2^{q-1}. Thus, we directly take the last 2^{q-1} chunks for splicing as the remainder part, and R' equals $\lceil R/2^{q-1} \rceil$.

The time-delay sampling operation is then applied along the chunk length dimension $2^{q-1}K$ for $\tilde{d}_q$. Specifically, time-delay sampling is to extract frames in a sequence at equal intervals, that is, to extract one frame every 2^{q-1} frames, and then the extracted frames are composed into a new sequence. For example, given a sequence of $[1, 2, 3, 4, ..., 500]$,

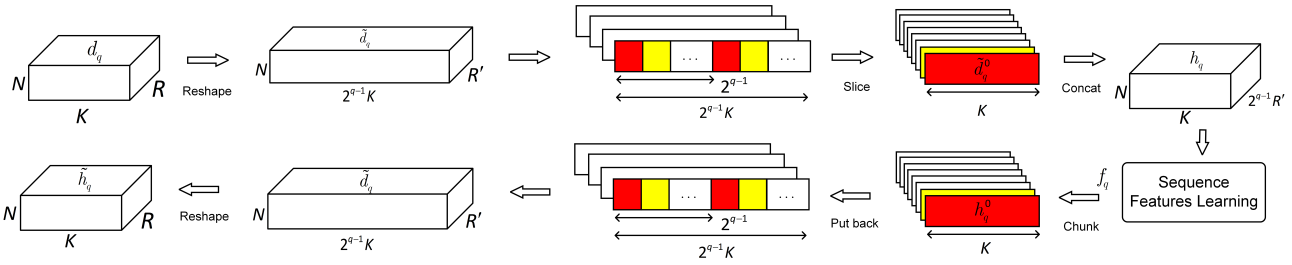

Fig. 3. The flowchart of the four main operations in TDSN, which implements the expansion of the receptive fields from K to $2^{q-1}K$ with the same time step K during features learning.

assuming that the sequence is sampled with a time-delay rate of 2, we can obtain $[1, 3, 5, 7, ..., 499]$ and $[2, 4, 6, 8, ..., 500]$. This operation can be implemented with sequence slicing in Python, and it is formulated as follows:

$$\tilde{d}_q^m = [\tilde{d}_q[:, m :: 2^{q-1}, :], m = 0, ..., 2^{q-1} - 1] \quad (2)$$

where $\tilde{d}_q[:, m :: 2^{q-1}, :]$ means to use the sequence slicing operation for $\tilde{d}_q$. We slice along the chunk length dimension $2^{q-1}K$ of $\tilde{d}_q$, and the length of each slice is 2^{q-1}. Then the frames with index m are taken out of each slice and spliced to get $\tilde{d}_q^m \in \mathbb{R}^{N \times K \times R'}$. And then we concatenate each $\tilde{d}_q^m$ along the dimension of R':

$$h_q = Concat(\tilde{d}_q^0, \tilde{d}_q^1, ..., \tilde{d}_q^{2^{q-1}-1}) \quad (3)$$

where $h_q \in \mathbb{R}^{N \times K \times 2^{q-1}R'}$ is the 3-D tensor reconstructed after time-delay sampling.

Subsequently, sequence features learning applies Bi-directional Long Short-Term Memory (Bi-LSTM) on the chunk length dimension K of h_q to capture the relationships between the sampled frames. A fully-connected layer is then used to transform the feature dimension:

$$f_q = [W_q g_q(h_q[:, :, j]) + b_q, j = 0, ..., 2^{q-1}R' - 1] \quad (4)$$

where $h_q[:, :, j] \in \mathbb{R}^{N \times K}$ is the sequence defined by the index j, $g_q(\cdot)$ refers to the Bi-LSTM network, $W_q \in \mathbb{R}^{N \times H}$ and $b_q \in \mathbb{R}^{N \times 1}$ are the weight and bias of the fully-connected layer, H is the dimension of the hidden layer in the Bi-LSTM, and $f_q \in \mathbb{R}^{N \times K \times 2^{q-1}R'}$ is the output.

Finally, sequence restoration puts the transformed sample frames in f_q back into $\tilde{d}_q$. To achieve this, we need to chunk f_q in the order in which h_q was concatenated:

$$[h_q^0, h_q^1, ..., h_q^{2^{q-1}-1}] = Chunk(f_q) \quad (5)$$

where $h_q^0, h_q^1, ..., h_q^{2^{q-1}-1} \in \mathbb{R}^{N \times K \times R'}$ and these sequences will be assigned to $\tilde{d}_q$:

$$\begin{cases} \tilde{d}_q[:, 0 :: 2^{q-1}, :] = h_q^0 \\ ..... \\ \tilde{d}_q[:, 2^{q-1} - 1 :: 2^{q-1}, :] = h_q^{2^{q-1}-1} \end{cases} \quad (6)$$

And then we reshape $\tilde{d}_q \in \mathbb{R}^{N \times 2^{q-1}K \times R'}$ into $\tilde{h}_q \in \mathbb{R}^{N \times K \times R}$ to realize the sequence restoration. Note that we directly take the last 2^{q-1} chunks for splicing as the remainder part in sequence recombination, so the duplicates in this 2^{q-1} chunks need to be removed during reshaping. In addition, layer normalization and residual structure are added for gradient propagation. And the final output of the q-th TDSN is $D_q \in \mathbb{R}^{N \times K \times R}$. With this TDSN, we can expand the receptive fields from K to $2^{q-1}K$ with the same time step K for features learning.

3. EXPERIMENT

3.1. Dataset

Our experiments are performed on the two-speaker speech WSJ0-2mix dataset [4], which is a benchmark dataset for two speaker monaural speech separation in recent years. The WSJ0-2mix dataset is derived from the WSJ0 data corpus [18], containing 30 hours of training and 10 hours of validation data. The input mixtures are generated by randomly selecting utterances of different speakers from WSJ0 training set, and mixing them at random signal-to-noise ratios between -5 dB and 5 dB. 5 hours of evaluation set is generated in the same way, using utterances from 16 unseen speakers in WSJ0 validation set and evaluation set. And the sampling frequency of waveform is 8 kHz.

3.2. Model Configurations and Training Details

In encoder and decoder, the kernel size W, the feature dimension N and H, and the chunk length K are set as 2, 64, 128, 250, respectively. The number of stacked dual-path networks B is set according to which dual-path network is used. In our experiments, when using the DPRNN, B is set as 5 which is different compared to the original 6 in [14]. And when using the DPTNet, B is set as 6 followed in [17]. What's more, the strategy of model training is also determined by the dual-path network used. Specifically, we train the whole model over 120 epochs with the initial learning rate $1e-3$ and decaying 0.98 every two epochs followed in [14], when using the DPRNN. While using the DPTNet, the warm up strategy

is employed followed in [17], and we train the whole model over 180 epochs. The reason for this difference in training strategies is mainly brought about by the different structures of the dual-path networks used. And we set the number of stacked TDSN Q as 6.

The speech durations in the dataset are all variable. To facilitate the model training, the input speech is cut to a fixed length of 4 seconds. Three cutting strategies are used: 1) Cutting the first 4 seconds of the speech; 2) Cutting the last 4 seconds of the speech; 3) Randomly selecting start point to cut for 4 seconds. All the models are trained with utterance-level permutation invariant training [19] to maximize Scale-Invariant Source-to-Noise Ratio (SI-SNR) [20]. And Adam [21] is used as the optimizer.

4. RESULTS AND DISCUSSIONS

In all experiments, we use two dominant evaluation criteria for speech separation, SI-SNR improvement (SI-SNRi) and Signal-to-Distortion Ratio improvement (SDRi) [22], to measure the performance of our proposed method.

Effectiveness Of TDSN: In order to evaluate the effectiveness of TDSN, we compare the performance of stacking different numbers of TDSN without dual-path networks, and the results are listed in Table 1. The n of TDSN-n in Table 1 indicates the number of stacked TDSN. As shown in Table 1, while the number of stacked TDSN increases, the performance becomes better. Moreover, the maximum number of TDSN in our experiment can be stacked to 6, and under this condition, we achieve the best result of 16.5 dB SDRi and 16.2 dB SI-SNRi with 1.3M parameters, which demonstrates the effectiveness of TDSN. It is because that the stacked TDSN learns sequence features with different scales from fine to coarse, and results in high performance in MSS.

Table 1. Comparison of stacking different numbers of TDSN without dual-path network.

Method	Model size	SI-SNRi(dB)	SDRi(dB)
TDSN-1	0.3M	9.8	10.1
TDSN-2	0.5M	12.6	12.9
TDSN-3	0.7M	14.1	14.4
TDSN-4	0.9M	14.8	15.1
TDSN-5	1.1M	15.6	15.9
TDSN-6	1.3M	16.2	16.5

Comparison With Different Separation Methods: Table 2 compares the performance of our proposed MTDS with the best results reported in the literatures on the WSJ0-2mix dataset. The MTDS(DPRNN) and MTDS(DPTNet) in Table 2 indicate the results of using DPRNN and DPTNet as the dual-path network, respectively. These results show that the MTDS attains an absolute improvement of 1.3 dB SI-SNRi compared with either the original DPRNN or DPTNet. According to this, we infer that MTDS can be added to any dual-path networks for fine to coarse sequence features learning. And the best MTDS results reach the SI-SNRi of 21.5 dB and the SDRi of 21.7dB, which outperform the state-of-the-arts without data augmentation. In addition, to explore whether this performance improvement is brought about by the increasing of parameters, we also conduct the experiments by adding 6 layers of Bi-LSTM without MTDS after DPRNN with the same setting . The results (DPRNN + 6 Bi-LSTM) in Table 2 show that, performance of DPRNN becomes worse instead without MTDS. This proves that the performance improvement indeed relies on our proposed MTDS.

Table 2. Comparison with different methods on WSJ0-2mix.

Method	Model size	SI-SNRi(dB)	SDRi(dB)
DPCL++ [23]	13.6M	10.8	-
uPIT-BLSTM-ST [19]	92.7M	-	10.0
BLSTM-TasNet [11]	23.6M	13.2	13.6
Conv-TasNet [7]	8.8M	15.3	15.6
FurcaNeXt [24]	51.4M	18.4	-
DPRNN [14]	2.6M	18.8	19.0
DPTNet [17]	2.6M	20.2	20.6
SepFormer [25]	26M	20.4	20.5
Wavesplit [26]	29M	21.0	21.2
DPRNN + 6 Bi-LSTM	3.5M	17.8	18.0
MTDS(DPRNN)	3.5M	20.1	20.3
MTDS(DPTNet)	4.0M	21.5	21.7

5. CONCLUSIONS

In this paper, we propose a novel neural network for monaural speech separation called Multiscale Time-Delay Sampling (MTDS). MTDS expands the receptive fields when learning the features at a fixed time step. And MTDS aggregates the sequence features at different scales, which effectively learns the sequence features from fine to coarse. Our results, reported on the WSJ0-2mix dataset, highlight that MTDS outperforms state-of-the-arts results without data augmentation. And the experiment results also prove that MTDS can be introduced to any dual-path networks, which makes our study more meaningful. For further work, we would like to verify the effectiveness of MTDS in noisy environments.

6. ACKNOWLEDGEMENTS

This work is supported in part by the Key Projects of the National Natural Science Foundation of China under Grant U1836220, the National Nature Science Foundation of China under Grant 62176106, 61906077, and Jiangsu Province key research and development plan under Grant BE2020036.

7. REFERENCES

[1] DeLiang Wang and Jitong Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM TASLP*, vol. 26, no. 10, pp. 1702–1726, 2018.

[2] Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis, "Deep learning for monaural speech separation," in *Proc. ICASSP*. IEEE, 2014, pp. 1562–1566.

[3] Hui Wang, Yan Song, Zeng-Xi Li, Ian McLoughlin, and Li-Rong Dai, "An online speaker-aware speech separation approach based on time-domain representation," in *Proc. ICASSP*. IEEE, 2020, pp. 6379–6383.

[4] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. ICASSP*. IEEE, 2016, pp. 31–35.

[5] Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. ICASSP*. IEEE, 2017, pp. 241–245.

[6] Yi Luo, Zhuo Chen, John R Hershey, Jonathan Le Roux, and Nima Mesgarani, "Deep clustering and conventional networks for music separation: Stronger together," in *Proc. ICASSP*. IEEE, 2017, pp. 61–65.

[7] Yi Luo and Nima Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM TASLP*, vol. 27, no. 8, pp. 1256–1266, 2019.

[8] Ziqiang Shi, Huibin Lin, Liu Liu, Rujie Liu, Jiqing Han, and Anyan Shi, "Deep attention gated dilated temporal convolutional networks with intra-parallel convolutional modules for end-to-end monaural speech separation," in *Interspeech*, 2019, pp. 3183–3187.

[9] Ziqiang Shi, Huibin Lin, Liu Liu, Rujie Liu, Shoji Hayakawa, Shouji Harada, and Jiqing Han, "End-to-end monaural speech separation with multi-scale dynamic weighted gated dilated convolutional pyramid network," in *Interspeech*, 2019, pp. 4614–4618.

[10] David Ditter and Timo Gerkmann, "A multi-phase gammatone filterbank for speech separation via tasnet," in *Proc. ICASSP*. IEEE, 2020, pp. 36–40.

[11] Yi Luo and Nima Mesgarani, "Tasnet: time-domain audio separation network for real-time, single-channel speech separation," in *Proc. ICASSP*. IEEE, 2018, pp. 696–700.

[12] Naoya Takahashi, Sudarsanam Parthasaarathy, Nabarun Goswami, and Yuki Mitsufuji, "Recursive speech separation for unknown number of speakers," *arXiv preprint arXiv:1904.03065*, 2019.

[13] Ziqiang Shi, Rujie Liu, and Jiqing Han, "Lafurca: Iterative refined speech separation based on context-aware dual-path parallel bi-lstm," *arXiv preprint arXiv:2001.08998*, 2020.

[14] Yi Luo, Zhuo Chen, and Takuya Yoshioka, "Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation," in *Proc. ICASSP*. IEEE, 2020, pp. 46–50.

[15] Ziqiang Shi, Rujie Liu, and Jiqing Han, "Speech separation based on multi-stage elaborated dual-path deep bilstm with auxiliary identity loss," *arXiv preprint arXiv:2008.03149*, 2020.

[16] Keisuke Kinoshita, Thilo von Neumann, Marc Delcroix, Tomohiro Nakatani, and Reinhold Haeb-Umbach, "Multi-path rnn for hierarchical modeling of long sequential data and its application to speaker stream separation," *arXiv preprint arXiv:2006.13579*, 2020.

[17] Jingjing Chen, Qirong Mao, and Dong Liu, "Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation," *arXiv preprint arXiv:2007.13975*, 2020.

[18] John Garofolo, David Graff, Doug Paul, and David Pallett, "Csr-i (wsj0) complete ldc93s6a," *Web Download. Philadelphia: Linguistic Data Consortium*, vol. 83, 1993.

[19] Morten Kolbæk, Dong Yu, Zheng-Hua Tan, and Jesper Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM TASLP*, vol. 25, no. 10, pp. 1901–1913, 2017.

[20] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R Hershey, "Sdr–half-baked or well done?," in *Proc. ICASSP*. IEEE, 2019, pp. 626–630.

[21] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte, "Performance measurement in blind audio source separation," *IEEE/ACM TASLP*, vol. 14, no. 4, pp. 1462–1469, 2006.

[23] Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R Hershey, "Single-channel multi-speaker separation using deep clustering," *arXiv preprint arXiv:1607.02173*, 2016.

[24] Liwen Zhang, Ziqiang Shi, Jiqing Han, Anyan Shi, and Ding Ma, "Furcanext: End-to-end monaural speech separation with dynamic gated dilated temporal convolutional networks," in *International conference on multimedia modeling*. Springer, 2020, pp. 653–665.

[25] Cem Subakan, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong, "Attention is all you need in speech separation," in *Proc. ICASSP*. IEEE, 2021, pp. 21–25.

[26] Neil Zeghidour and David Grangier, "Wavesplit: End-to-end speech separation by speaker clustering," *IEEE/ACM TASLP*, vol. 29, pp. 2840–2849, 2021.