A two-stage speaker extraction algorithm under adverse acoustic conditions using a single-microphone

Aviad Eisenberg
Bar-Ilan University, OriginAI

Sharon Gannot
Bar-Ilan University

Shlomo E. Chazan
OriginAI, Bar-Ilan University

***Abstract*—In this work, we present a two-stage method for speaker extraction under reverberant and noisy conditions. Given a reference signal of the desired speaker, the clean, but the still reverberant, desired speaker is first extracted from the noisy-mixed signal. In the second stage, the extracted signal is further enhanced by joint dereverberation and residual noise and interference reduction. The proposed architecture comprises two sub-networks, one for the extraction task and the second for the dereverberation task. We present a training strategy for this architecture and show that the performance of the proposed method is on par with other state-of-the-art (SOTA) methods when applied to the WHAMR! dataset. Furthermore, we present a new dataset with more realistic adverse acoustic conditions and show that our method outperforms the competing methods when applied to this dataset as well.**

***Index Terms*—Speaker extraction, Dereverberation**

I. Introduction

Extracting a desired speaker from a mixture of overlapping speakers using only a single microphone is a cumbersome task, particularly in noisy and reverberant environments. In this paper, we address this challenge by focusing on the extraction of a single participant from a mixture of two speakers acquired by a single microphone, given a prerecorded utterance of the speaker to be extracted.

There has been significant progress in the single-microphone blind source separation (BSS) domain in the past years. The Conv-Tasnet [1] and the dual-path recurrent neural network (DPRNN) [2], are both applied in the time domain with similar encoder-masking-decoder architecture. Other works that followed this approach were presented [3]–[10], demonstrating a considerable improvement in the separation results. The SepFormer was introduced in [3] leveraging the benefits of the attention layers, which led to a significant improvement in performance and to SOTA results. An efficient convolutional neural network (CNN)-based model, denoted Sudo rm-rf, was presented in [9] and demonstrated high separation capabilities. Most of the above-mentioned BSS models were trained and tested on clean and anechoic mixtures. Such acoustic conditions can hardly be met in reality. Several algorithms [3], [6], [9], [10] were also trained on reverberant data without any changes in their architecture. Cord-Landwehr et al. showed in [11] that despite the significant improvement achieved in clean conditions, only marginal improvements can be obtained in realistic reverberant and noisy conditions.

Given a reference signal of the desired speaker turns the BSS problem into an extraction problem, in which the permutation problem is alleviated. The SpeakerBeam algorithm, introduced in [12], estimates a mask for the desired speaker in the spectral domain using the spectrum of the reference signal. While magnitude-domain processing might be sufficient in clean and anechoic conditions, it might be insufficient in noisy and reverberant conditions. In [13], this model was improved by using the time-domain signal, as it allows the exploitation of the entire signal information. A similar approach was presented in [14], where the i-vector [15] of the reference signal was used as the embedding of the desired speaker. In [16], a multi-task training procedure was proposed in which a speaker classification task is carried out in parallel for improving the embedding of the desired speaker.

Time domain processing, despite the above advantages, ignores the time-frequency patterns typical to speech signals. In our prior work, [17], a fully convolutional Siamse-Unet architecture was proposed. The algorithm is applied in the short-time Fourier transform (STFT) domain to the Real-Imaginary (RI) representation of the signals while the loss is applied in the time-domain, exploiting the entire signal, on the one hand, and leveraging its spectral patterns, on the other hand. Yet, the performance of this approach is insufficient in adverse acoustic conditions.

In the current contribution, we present a two-stage algorithm to extract a desired speaker from a mixture of two signals under reverberant and noisy conditions. We split the extraction task into two stages. In the first stage, given the noisy and reverberant mixture and the reference signals, a Siamse-Unet architecture is applied to extract the *reverberant* desired speaker. The encoders used for both the mixture and the reference signals are identical, thus the resulting outputs have matching dimensions. While the mixture encoder preserves the frame dimensions, which is essential for the mixture processing, the reference encoder aims to exclusively represent the desired speaker's identity while ignoring the content of the utterance. To achieve this outcome, we average the reference embedding over the frame dimension. The reference embedding vector is finally multiplied with each of the frames in the mixture embedding. The outcome of this multiplication is used as an

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

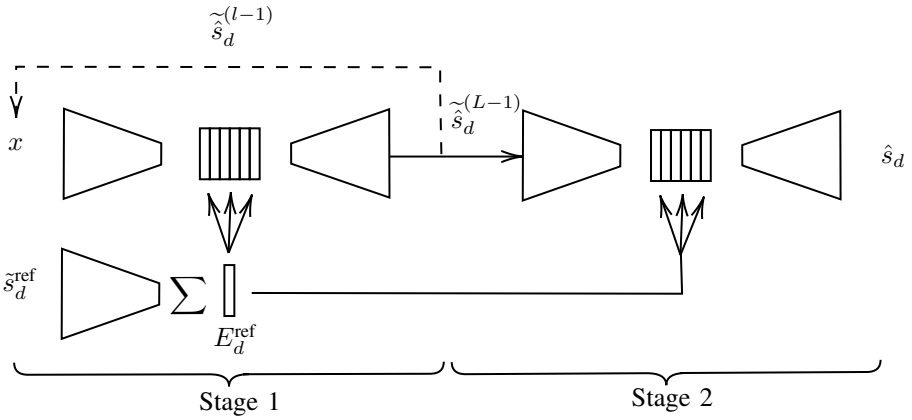

Fig. 1: Block diagram of the proposed two-stage architecture. In the first iteration, the model takes the given observations as input. For subsequent iterations, the output of the previous iteration is used as input instead of the mixture. The network is using skip connections between the mixture encoder and decoder (not shown explicitly in the diagram). No skip connections from the reference encoder are implemented. The two encoders share the same weights. The arrows denote the element-wise multiplication of the reference embedding with the embedding of each frame. Only the output of the final iteration is used as input to the second stage.

input to the decoder, which in turn extracts the reverberant desired speaker. We show that training this stage in an iterative manner is beneficial.

In the second stage, an additional Unet model is applied to dereverberate and enhance the output of the first stage. Similarly, the encoder output preserves the frame size of its input signal. The resulting embedding is multiplied by the embedding of the reference from the first stage. The second decoder is finally applied to extract the desired *clean and dereverberated* signal.

Furthermore, in this paper, we introduce a new simulated dataset with more realistic conditions than the WHAMR! dataset, and show that our model outperforms other SOTA models on both the WHAMR! dataset and the new, more challenging, dataset.

II. Problem Formulation

The signal $x(t)$, captured by a single microphone, is a combination of Q concurrent speakers, represented by:

$$x(t) = \sum_{q=1}^{Q} \{s_q * h_q\}(t) + v(t) \quad t = 0, 1, \ldots, T-1 \tag{1}$$

where $s_q(t)$ is the signal of the qth speaker, $h_q(t)$ is the room impulse response (RIR) between the qth speaker position and the microphone position, and $v(t)$ is an additive noise. In a noise-free, non-reverberant environment, $h_q(t)$ is dominated by the first arrival, and $v(t) = 0$ for all q.

In the STFT domain, the microphone signal can be approximately expressed as:

$$x(n,k) = \sum_{q=1}^{Q} s_q(n,k) h_q(n,k) + v(n,k) \tag{2}$$

where $n = 0, 1, \ldots, N-1$ and $k = 0, 1, \ldots, K-1$ represent the time-frame and frequency-bin indexes, respectively, and N and K are the total number of time-frames and frequency bands, respectively.

This paper focuses on the case where there are only two concurrent speakers, namely $Q = 2$, referred to as the desired speaker $s_d(n,k)$ and the interference speaker $s_i(n,k)$. The reverberant desired signal is defined as $\tilde{s}_d(n,k) = s_d(n,k)h_d(n,k)$. The reference signal is denoted $s_d^{\text{ref}}(n,k)$. We aim at the extraction of the desired speaker signal, $\hat{s}_d(n,k)$, using the mixed signal $x(n,k)$, and a reverberant reference signal, $\tilde{s}_d^{\text{ref}}(n,k) = s_d^{\text{ref}}(n,k)h_d^{\text{ref}}(n,k)$. In this paper we assume the reference and the reverberant desired signals have the same RIR.

III. Proposed Model

A. Architecture and Training Procedure

Our model is composed of two sub-stages. The first is a Simase-Unet, which consists of three parts: two encoders and a decoder. We share weights between the encoders to encourage joint embeddings of both the mixture and the reference signals in the same latent space. The encoder architecture consists of several convolution layers followed by two-dimensional batch normalization and a 'Relu' function (similar to the one introduced in [17]). Next, we combine the dimensions of the channels and frequencies and employ a fully-connected layer to reduce the dimensions. After this step, we apply a single transformer-encoder layer. The decoder architecture consists of six transformer-encoder layers, followed by fully connected (FC) layer to restore the original dimension. Then transpose-convolution layers are employed to adapt to the convolution layers in the encoder, enabling the application of skip connections as required. A transformer-encoder layer is subsequently applied after all the steps mentioned above. We repeat the first stage several times to further enhance the extraction process. In the first iteration, the mixture signal is

processed, while in the subsequent iterations, the separated (but still reverberant) signals from the previous iteration are processed. Formally, the process can be expressed as:

$$\text{Input}^{(\ell)} = \begin{cases} x(n,k) & \ell = 0 \\ \hat{\tilde{s}}_d^{(\ell-1)}(n,k) & \ell > 0 \end{cases}$$

where $\ell = 0, ..., L-1$ is the iteration index. By repeating this process for L iterations, we obtain L estimates of $\tilde{s}_d(n,k)$, which are all used to train the entire model.

The second stage of the model uses the same architecture as the first stage. Our empirical results showed that using the reveberant reference signal in the second phase can improve the results. Rather than passing the reference signal again through an encoder, we can simply use the learned embedding vector from the first stage.

Alternative ways for integrating the information from the reference signal are described in [12], including concatenation, addition, and multiplication, the latter achieving the best results. To obtain a single vector that represents the speaker's identity, we average across the frame dimensions of the reference embedding, thus ignoring the temporal information and emphasizing the speaker's identity. The final embedding vector is denoted E_d^{ref}. Unlike [17], in the Unet architecture, skip connections are only implemented from the mixture encoder and not from the reference encoder. Instead, we only use the output of the last layer of the reference encoder in the bottleneck stage. While most single microphone DNN-based algorithms apply a masking operation to the mixture signal, the proposed scheme is trained to directly estimate the time-frequency (TF) representation of the target source.

The two sub-stages are trained together in an end-to-end manner, while the first stage feeds the second phase with an estimate of the last iteration of the first stage and the reference embedding. A block diagram of the entire model is shown in Fig. 1.

B. Features

In this work, we adopted the Real-Imaginary (RI) components of the STFT as both the input features of the model and its output. The real and imaginary parts were concatenated in the channel dimension. The model is trained with the scale-invariant signal-to-distortion ratio (SI-SDR) loss function, which is sensitive to phase distortion. Using the RI features may alleviate such problems (see discussion in [17]).

C. Objectives

As mentioned above, we use the SI-SDR loss function to train our model. The loss is formulated as

$$\text{SI-SDR}\left(s, \hat{s}\right) = 10 \log_{10} \left(\frac{\left\| \frac{\langle \hat{s}, s \rangle}{\langle s, s \rangle} s \right\|^2}{\left\| \frac{\langle \hat{s}, s \rangle}{\langle s, s \rangle} s - \hat{s} \right\|^2} \right). \quad (3)$$

The model is trained using all output signals, namely, $\hat{s}_d$ and $\hat{\tilde{s}}_d^{(\ell)}, \ell = 0, \ldots, L-1$:

$$\mathcal{L}_{\text{SISDR}_d} = \sum_{\ell=0}^{L-1} \text{SI-SDR}\left(\tilde{s}_d, \hat{\tilde{s}}_d^{(\ell)}\right) + \text{SI-SDR}\left(s_d, \hat{s}_d\right). \quad (4)$$

For the extraction task to be successful, the network must be able to learn a unique embedding for each speaker to prevent errors in identifying the correct speaker. To achieve this goal, an additional, triplet loss function, was implemented:

$$\text{TRIPLET}(a, p, n) = \max(\text{cd}(a, p) - \text{cd}(a, n) + m, 0) \quad (5)$$

where a is the anchor input, p is the positive input and n is the negative input, with p closer to a than n. The function $\text{cd}(\cdot)$ is the cosine distance and m is a margin hyperparameter. The triplet loss function encourages the distance between the anchor and the positive inputs to be smaller than the distance between the anchor and negative inputs, by a margin of at least m. In our case, we would like the embedding of the reverberant reference to be as close as possible to the embedding of the output of the first phase (namely, the estimated desired and reverberant speaker), and as far as possible from the embedding of the reference of the second speaker. In explicit terms:

$$\mathcal{L}_{\text{TRIPLET}_d} = \text{TRIPLET}(E_{\hat{\tilde{s}}_d}, E_d^{\text{ref}}, \overline{E_d^{\text{ref}}}) \quad (6)$$

where $E_{\hat{\tilde{s}}_d}$ is obtained by passing $\hat{\tilde{s}}_{d_{L-1}}$ through the encoder of stage 1 and $\overline{E_d^{\text{ref}}}$ is the embedding of the reference of the interference signal.

During training, we encountered a convergence problem when using both loss functions simultaneously. To address this issue, we implemented a warm-up training procedure in which the network is initially trained using only the SI-SDR loss, and the triplet loss is added at a later stage in the training process. This approach successfully resolved the convergence issues.

In an effort to improve the training process, we alternated the desired and interference signals within each training batch, while maintaining consistency in the mixture employed. That is, inserting the mixture signal with the reference signal of one of the speakers and then repeating the process with the reference of the other speaker in the same batch, and summing the losses for both speakers. In short, the overall loss function takes the following form:

$$\mathcal{L} = (\mathcal{L}_{\text{SISDR}_d} + \mathcal{L}_{\text{SISDR}_i})/2 + \alpha \cdot \mathbb{1}_{\text{warm-up}} \cdot (\mathcal{L}_{\text{TRIPLET}_d} + \mathcal{L}_{\text{TRIPLET}_i})/2 \quad (7)$$

where α represents a hyperparameter, and the indicator function $\mathbb{1}_{\text{warm-up}}$ determines the point at which the triplet objective function should be taken into consideration in the training process.

IV. Experimental Study

A. Datasets

We used the WHAMR! dataset to train our model. This dataset is created by taking the WSJ0-2Mix dataset [18] and modifying it by incorporating environmental noise from the WHAM dataset [19] and reverberation. To adapt the dataset to the extraction task, we modified it in the following manner. For each speaker included in the mixture, we selected a different utterance and convolve it with the same RIR used to generate the mixed signal, namely $h_d^{\text{ref}} = h_d$. This procedure

TABLE I: Noisy reverberant data specification.

Room dim. [m]	H_x	$U[4,8]$
	H_y	$U[4,8]$
	H_z	$U[2.5,3]$
Reverb. time [sec]	T_{60}	$U[0.2,0.6]$
Mic. Pos. [m]	x	$\frac{H_x}{2}+U[-0.5,0.5]$
	y	$\frac{H_y}{2}+U[-0.5,0.5]$
	z	1.5
Sources Pos. [°]	θ	U[0,180]
Sources Distance [m]		$1+U[-0.5,0.5]$

reflects the fact that in a typical conversation, segments in which only a single speaker is active can always be found. However, it is implicitly assumed that the scenario is static, hence that the RIR does not significantly change during the entire conversation.

We note that, according to our tests, the reverberation level in the WHAMR! dataset does not exceed 600 milliseconds, in contradiction to the reported reverberation level, which is in the range of $[0.2,1]$.[1]

The dataset includes 20,000 signals for training, 5,000 for validation, and 300 for the test phase, and it uses the 'min' and '8k' sampling rate configuration. (With 'min' setting the longer target is truncated to match the length of the shorter target.)

In addition to WHAMR!, we generated a new dataset for the purpose of enriching the data. This is equivalent to *dynamic mixing* training, which randomly generates the mixture from the existing speakers during training. We also took speakers from the WSJ0 corpus, along with noise from the WHAM and the reverberation generated from a RIR generator [20] with parameters listed in Table I.

During training, each signal is truncated to a variable length between 2 to 5 seconds. Since we are using a Siamese architecture, the mixture and the reference signal must have the same length. If the reference signal is longer, it will be truncated, and if it is shorter, it will be duplicated until it is the same length as the mixture.

B. Algorithm Settings

The frame-size of the STFT is 256 samples with 50% overlap. Due to the symmetry of the discrete Fourier transform (DFT) only the first half of the frequency bins are used. The value of α was empirically set to 2, emphasizing the triplet loss due to the significant difference in scales between the two objective functions. The triplet loss margin was set to $m=0.5$.

The number of iterations for the first phase was chosen as $L=2$, because there was minimal improvement when increasing the number from 2 to 3 iterations, while a noticeable improvement was observed between 2 iterations to no iterations, $L=1$.

[1]Due to space constraints, we will not give a detailed analysis of the dataset in the current contribution.

In the training procedure, we used the Adam optimizer [21]. The learning rate was set to 0.001 and the training batch size to 6. The weights are randomly initialized, and the lengths of the signals were randomly changed at each batch.

C. Evaluation Measures

To evaluate the proposed algorithm we use five evaluation measures: SI-SDR, signal-to-interference ratio (SIR), signal to distortion ration (SDR), short-time objective intelligibility (STOI), and perceptual evaluation of speech quality (PESQ). While the first three are used as a measurement of the quality of the speaker separation, the last two give an indication of the audio intelligibility and quality.

The proposed algorithm is compared to the current SOTA separation methods, i.e., the Sepformer [3] and the Sudo rm-rf [9]. These are time-domain source separation masking-based methods. We decided to compare our method with separation methods rather than extraction methods since these are the most effective methods in the field.

D. Results

The results for the WHAMR! dataset are depicted in Table II. Our model achieves an SI-SDR of 9.67 dB, SDR of 10.88 dB, and SIR of 24.2 dB. It is evident that our proposed method outperforms the SOTA methods in almost all measures. In addition, the method also achieves the best scores for the intelligibility measure (STOI) and the quality measure (PESQ), with scores 92% and 2.72, respectively.

The new dataset imposes a greater challenge on the extraction algorithm, as evidenced by the lower scores in Table III for all measures, compared to the scores obtained on the WHAMR! dataset, as reported in Table II. While the absolute separation results obtained for the new dataset are lower, the improvement in terms of SI-SDR is 14.2 dB, which is very high and significantly outperforms the competing methods. The intelligibility results (90.2%) are on par with the results obtained for the WHAMR! dataset.

TABLE II: Results for WHAMR! dataset

Model	SI-SDR	SDR	SIR	STOI	PESQ
Unprocessed	-3.84	-0.59	0.19	65.3	1.51
Sudo rm-rf [9]	8.13	10.7	23.7	90.2	2.5
Sepformer [3]	8.86	10	**25**	91.3	2.57
Proposed	**9.67**	**10.88**	24.2	**92**	**2.72**

TABLE III: Results for the new dataset

Model	SI-SDR	SDR	SIR	STOI	PESQ
Unprocessed	-7.99	-0.79	0.12	52.5	1.54
Sudo rm-rf [9]	1.7	3.46	15.8	69.9	2.1
Sepformer [3]	1.89	4.82	18.48	68.8	2.05
Proposed	**6.21**	**7.98**	**22.14**	**90.2**	**2.62**

E. Ablation Study

We present an ablation study for our model. We examined four different configurations:

1) One iteration in the first stage. The loss function for the desired source is given by:

$$\mathcal{L}_{\text{SISDR}_d} = \text{SI-SDR}\left(\tilde{s}_d, \hat{\tilde{s}}_d^{(L-1)}\right) + \text{SI-SDR}\left(s_d, \hat{s}_d\right) \quad (8)$$

with $L = 1$, and the overall loss is given by $\mathcal{L} = (\mathcal{L}_{\text{SISDR}_d} + \mathcal{L}_{\text{SISDR}_i})/2$. Triplet loss is not applied.
2) Two iterations in the first stage. The SI-SDR loss is only applied to the final output $\hat{\tilde{s}}_d^{(L-1)}$, i.e. $L = 2$ in (8). Triplet loss is not applied.
3) The SI-SDR loss is applied to all intermediate results $\ell = 0, \ldots, L-1$, as in (4), with $L = 2$. Triplet loss is not applied.
4) The full implementation of the proposed model with all its components active.

Table IV depicts the breakdown of the results for the WHAMR! and the new datasets. It is evident that each additional component enhances the quality of the network output for both datasets. In total, the SI-SDR measure improved from 8.62 dB to 9.67 dB for the WHAMR! dataset and from 5.45 dB to 6.21 dB for the new dataset. Respectively, STOI improved from 90.4% to 92% for WHAMR!, and from 88% to 90.2% for the new dataset

Training the model to accurately identify the intended speaker from a mixture is challenging in speaker extraction, particularly in reverberant conditions and when the speakers have similar voices. This may result in the extraction of the incorrect speaker or a permutation between the output signals. To address this issue, the triplet loss was added. Our experiments showed that the addition of the triplet loss alleviated such permutation problems.

TABLE IV: Ablation Study for all 4 configurations.

	WHAMR!		New Dataset	
Config.	SI-SDR	STOI	SI-SDR	STOI
1)	8.62	90.4	5.45	88
2)	9.13	91	5.71	88.9
3)	9.26	91.8	6.02	**90.2**
4)	**9.67**	**92**	**6.21**	**90.2**

V. Conclusions

We have proposed a two-stage approach for speaker extraction under reverberant conditions. The first stage separates the desired and yet reverberated speaker, while the second stage reduces reverberation and further enhances separation quality. Our results indicate that our model performs comparably or better than current state-of-the-art separation methods, with the added benefits of faster and more consistent training. Furthermore, an ablation study identifies the role of the various components in improving performance.

References

[1] 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*, vol. 27, no. 8, pp. 1256–1266, 2019.

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

[3] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 21–25.

[4] K. Wang, H. Huang, Y. Hu, Z. Huang, and S. Li, "End-to-end speech separation using orthogonal representation in complex and real time-frequency domain." in *Interspeech*, 2021, pp. 3046–3050.

[5] S. Lutati, E. Nachmani, and L. Wolf, "SepIt approaching a single channel speech separation bound," *arXiv preprint arXiv:2205.11801*, 2022.

[6] S. E. Chazan, L. Wolf, E. Nachmani, and Y. Adi, "Single channel voice separation for unknown number of speakers under reverberant and noisy settings," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 3730–3734.

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

[8] J. Chen, Q. Mao, and D. Liu, "Dual-Path Transformer Network: Direct Context-Aware Modeling for End-to-End Monaural Speech Separation," in *Proc. Interspeech 2020*, 2020, pp. 2642–2646.

[9] E. Tzinis, Z. Wang, X. Jiang, and P. Smaragdis, "Compute and memory efficient universal sound source separation," *Journal of Signal Processing Systems*, vol. 94, no. 2, pp. 245–259, 2022.

[10] N. Zeghidour and D. Grangier, "Wavesplit: End-to-end speech separation by speaker clustering," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2840–2849, 2021.

[11] T. Cord-Landwehr, C. Boeddeker, T. Von Neumann, C. Zorilă, R. Doddipatla, and R. Haeb-Umbach, "Monaural source separation: From anechoic to reverberant environments," in *2022 International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2022.

[12] K. Žmolíková, M. Delcroix, K. Kinoshita, T. Ochiai, T. Nakatani, L. Burget, and J. Černockỳ, "Speakerbeam: Speaker aware neural network for target speaker extraction in speech mixtures," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 800–814, 2019.

[13] M. Delcroix, T. Ochiai, K. Žmolíková, K. Kinoshita, N. Tawara, T. Nakatani, and S. Araki, "Improving speaker discrimination of target speech extraction with time-domain speakerbeam," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 691–695.

[14] C. Xu, W. Rao, E. S. Chng, and H. Li, "Time-domain speaker extraction network," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 327–334.

[15] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[16] C. Xu, W. Rao, E. S. Chng, and H. Li, "Spex: Multi-scale time domain speaker extraction network," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 28, pp. 1370–1384, 2020.

[17] A. Eisenberg, S. Gannot, and S. E. Chazan, "Single microphone speaker extraction using unified time-frequency siamese-unet," in *30th European Signal Processing Conference (EUSIPCO)*, 2022, pp. 762–766.

[18] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2016, pp. 31–35.

[19] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. L. Roux, "WHAM!: Extending Speech Separation to Noisy Environments," in *Proc. Interspeech 2019*, 2019, pp. 1368–1372.

[20] E. A. Habets, "Room impulse response generator," *Technische Universiteit Eindhoven, Tech. Rep*, vol. 2, no. 2.4, p. 1, 2006.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.