ALL-NEURAL ONLINE SOURCE SEPARATION, COUNTING, AND DIARIZATION FOR MEETING ANALYSIS

Thilo von Neumann[1,2]*, Keisuke Kinoshita*[1]*, Marc Delcroix*[1]*, Shoko Araki*[1]*,*
Tomohiro Nakatani[1]*, Reinhold Haeb-Umbach*[2]

[1] NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan
[2] Paderborn University, Department of Communications Engineering, Paderborn, Germany

ABSTRACT

Automatic meeting analysis comprises the tasks of speaker counting, speaker diarization, and the separation of overlapped speech, followed by automatic speech recognition. This all has to be carried out on arbitrarily long sessions and, ideally, in an online or block-online manner. While significant progress has been made on individual tasks, this paper presents for the first time an all-neural approach to simultaneous speaker counting, diarization and source separation. The NN-based estimator operates in a block-online fashion and tracks speakers even if they remain silent for a number of time blocks, thus learning a stable output order for the separated sources. The neural network is recurrent over time as well as over the number of sources. The simulation experiments show that state of the art separation performance is achieved, while at the same time delivering good diarization and source counting results. It even generalizes well to an unseen large number of blocks.

***Index Terms*—** Blind source separation, neural network, meeting diarization, online processing, source counting.

1. INTRODUCTION

The automatic analysis of meetings promises to relieve humans from tedious transcription and annotation work. It comprises the tasks: (a) diarization, i.e., determining who is speaking when, (b) source counting, i.e., estimating the number of speakers in a meeting, (c) separating overlapped speech, i.e., carrying out (blind) source separation, and (d) recognizing the separated streams. All of these are challenging tasks by themselves, which become even more demanding considering the fact that meetings can be arbitrarily long, making batch processing practically unfeasible and asking for block-online processing instead.

In recent years, a substantial amount of research has been devoted to the meeting scenario [1–3]. One of the key challenges is the separation and recognition of overlapped speech. Perhaps surprisingly, even in professional meetings, the percentage of overlapped speech, i.e., time segments where more than one person is speaking, is in the order of 5% - 10%[1], while in informal get-togethers it can easily exceed 20%[2]. Recently, many promising neural network (NN)-based single-channel approaches have been proposed to solve the problem of source separation, such as Deep Clustering (DC) [4], Deep Attractor Network (DAN) [5] and Permutation Invariant Training (PIT) [6, 7]. DC and DAN can be viewed as two-stage algorithms, where in the first stage embedding vectors are estimated for each time-frequency (T-F) bin. In the second stage, these embedding vectors are clustered to obtain masks, from which the sources can be recovered by applying the masks to the speech mixture. Note that the number of sources has to be known to determine the correct number of clusters. PIT, on the contrary, is a single-stage algorithm, because it lets NNs directly estimate source separation masks without an explicit clustering step. In PIT, however, the network architecture depends on the maximum number of sources to be extracted.

Considering this dependency on the number of sources, we proposed the Recurrent Selective Attention Network (RSAN), which is a purely NN-based mask estimator capable of, in theory, handling an arbitrary number of speakers [8]. Specifically, RSAN is predicated on a recurrent neural network (RNN) which can learn and determine how many iterations, i.e., source extraction processes, have to be performed to extract all sources [9]. It extracts one source at a time from the mixture and repeats this process until all sources are extracted. In experiments it achieved source separation performance comparable with PIT, and excellent source number counting accuracy.

None of these NN-based source separation algorithms [4–8] has been extended to block or block-online processing in realistic situations, which consist of long recordings of an arbitrary number of intermittent speakers. Furthermore, a diarization component should be included, which ensures that the same speaker appears always at the same output node, even if he/she remains silent for some time.

Most conventional meeting diarization approaches perform block-offline or block-online processing by carrying out the following two steps sequentially [1, 10–13]. First, at each block, they perform separation (if necessary) and obtain speaker identity information about each speaker in the block in the form of, e.g., i-vectors [14], x-vectors [15], or spatial signatures [10, 11, 16]. Then, the correct association of speaker identity information across block boundaries, i.e., the eventual diarization result, is established by clustering this information in offline [12, 13] or online manners [11]. Here, block-offline processing is allowed to utilize future data, while the block-online processing is not. In [17], joint separation and diarization is attempted using spatial mixture models. This, however, requires multichannel input and does not exploit spectral information for speaker re-identification.

Here, we also consider separation and diarization jointly, however proposing a novel all-neural block-online approach that performs source separation, source number counting and diarization all together. The fact that the model is all-neural makes it possible to optimize the entire block-online process through error backpropagation during NN training. Importantly, in theory, the proposed method can handle any meeting situation where, for example, a new speaker starts speaking in the middle of the meeting, or one or more of the meeting attendees remain silent for a significant amount of time after his/her first utterances. The method is an extension of [8].

[1] measured on the AMI meeting corpus [3].
[2] measured on the *Computational Hearning in Multisource Environments* (CHiME-5) database.

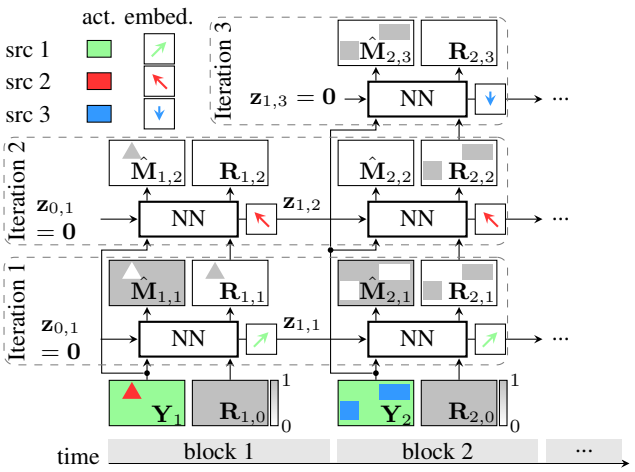

Fig. 1. Proposed method unrolled over two time blocks and a maximum of three iterations. In block 1, src1 ■, corresponding to backgound noise, and src2 ■ are separated. Then, in block 2, the NN receives embedding vectors for src1 and src2, extracts src1, estimates an empty mask for silent src2, and extracts the new src3 ■.

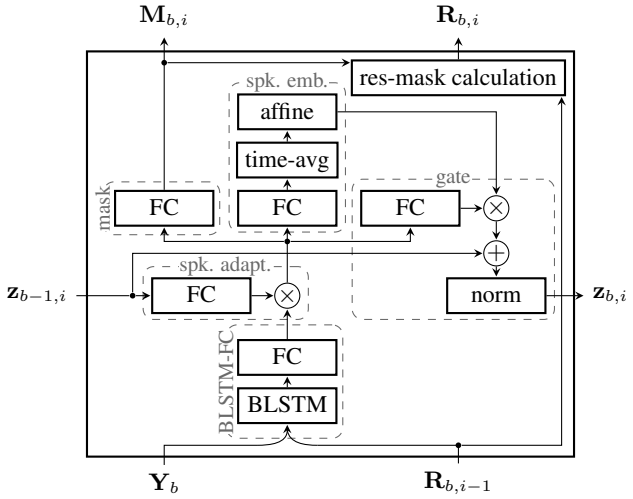

Fig. 2. Detailed structure of the neural network.

2. PROPOSED METHOD

2.1. Overall Structure

The algorithm works in a block-online manner and in each time block it successively extracts the sources until no sources are found anymore. Fig. 1 depicts the processing steps for the first two time blocks and for up to three source extraction iterations per block.

Let b denote the time block index and i the iteration index within a block. At every iteration in a block, the network receives three inputs: the input spectrogram $\mathbf{Y}_b$, a residual mask $\mathbf{R}_{b,i-1}$ which is the output of the previous iteration on the same block, and a speaker adaptation input $\mathbf{z}_{b-1,i}$ from the previous block. These inputs are processed in a neural network ("NN"), which outputs a source separation mask $\hat{\mathbf{M}}_{b,i}$, an updated residual mask $\mathbf{R}_{b,i}$, and a speaker embedding $\mathbf{z}_{b,i}$, which represents the identity of the extracted speaker.

The residual mask can be seen as an attention map that, once initialized with $\mathbf{R}_{b,0} = \mathbf{1}$ in every first iteration and updated in every following iteration, guides the network where to attend in order to extract a speaker that was not extracted in a previous iteration. During test, the model decides when to stop the iterations based on a thresholding operation applied to the mean of the residual mask; it stops processing after iteration i, if the residual mask is virtually empty, i.e., $\frac{1}{TF}\sum_{tf}[\mathbf{R}_{b,i}]_{tf} < t_{\text{res-mask}}$.

In the first processing block, $b = 1$, no speaker information is available from the previous block. Therefore, the input speaker information is set to zero: $\mathbf{z}_{0,i} = \mathbf{0}$. Without guidance, the network decides on its own in which order to extract the source signals. The embedding vector $\mathbf{z}_{b,i}$ is passed as an adaptation input to the next time block, $b+1$, and guides the i-th iteration on that block to extract the same speaker as in (b, i). This is related to the 'SpeakerBeam' concept to adapt a mask estimation network to a particular speaker [18]. Thus, it is ensured that all blocks extract the speakers in the same order. In Fig. 1, the different sources are indicated by their color, and it can be seen that the green source (src 1) is always extracted in the first, the red in the second and the blue in the third iteration. If a source happens to be silent in a particular block (see the red source in block 2), then the mask is filled with zeros ($\hat{\mathbf{M}}_{b,i} = \mathbf{0}$), and the residual mask stays unmodified ($\mathbf{R}_{b,i} = \mathbf{R}_{b,i-1}$) in the iteration i that is in charge of that source.

If the criterion to stop the speaker extraction iterations is not met after extracting all speakers found in previous blocks, the model increases the number of iterations to extract any new speaker (see iteration 3 of block 2 in Fig. 1) until the stopping criterion is finally fulfilled. To summarize, the network essentially attempts a guided source extraction for each source found in earlier blocks, and performs blind source separation on the remaining signal.

Note that the original RSAN [8] was formulated for processing only one block and does not receive and output speaker embeddings, whereas the proposed method, an extension of [8], has enhanced capability of tracking speakers from block to block by doing so.

2.2. Details of used Neural Network

Fig. 2 depicts the detailed structure of the neural network "NN" in Fig. 1. In the figure, "FC" corresponds to a fully connected layer with a sigmoid activation, "affine" to affine transformation, "time-avg" to time averaging, and "norm" to length normalization. The network consists of a common stack of bidirectional long short term memory (BLSTM) RNNs and a fully connected layer, hereafter denoted by BLSTM-FC, followed by multiple specialized parts (gray dashed boxes): A speaker adaptation network, a mask estimation network, a speaker embedding estimation network, and a gate to control the update of speaker embedding vectors.

Two inputs to the NN, the spectrogram $\mathbf{Y}_b$ and the residual mask $\mathbf{R}_{b,i-1}$, are concatenated before being passed to the BLSTM-FC, and its output is passed through the speaker adaptation network, whose output is fed into the three remaining specialized networks.

Speaker adaptation is achieved by multiplying the transformed speaker adaptation input $\mathbf{z}_{b-1,i}$ with the activations from BLSTM-FC. By weighting the neurons based on the speaker embedding, the network behavior is modified to extract a specific speaker [19].

The speaker embedding estimation is inspired by 'Deep Speaker' [20]. Here, the output of a FC layer is averaged over time to condense the speaker information of the whole block b into one embedding vector $\mathbf{z}_{b,i}$. If a cosine distance-based loss function (described later) is used, this vector is further transformed and normalized.

An optional gate is used to be able to pass the speaker embedding vector unmodified to the next time block, if the speaker is silent in the current block. The gating mechanism ensures that the speaker

information is unmodified if that speaker is absent in the current block, and updated if it is present.

Finally, a source separation mask $\hat{\mathbf{M}}_{b,i}$ is estimated by the mask estimation network. This mask is used to update the residual mask by subtracting it from the residual mask obtained from the previous iteration and clipping to a range of $[0, 1]$:

$$\mathbf{R}_{b,i} = \max(\mathbf{R}_{b,i-1} - \hat{\mathbf{M}}_{b,i}, 0). \quad (1)$$

2.3. Training Objectives

During training, the network is unrolled over multiple blocks and iterations and can be trained with back-propagation using the following multi-task cost function:

$$\mathcal{L} = Ł^{(\text{MMSE})} + \alpha Ł^{(\text{res-mask})} + \beta Ł^{(\text{CE})} + \gamma Ł^{(\text{TRIPLET})}, \quad (2)$$

which is a weighted sum of the reconstruction loss $Ł^{(\text{MMSE})}$, the source counting loss $Ł^{(\text{res-mask})}$, and the speaker embedding losses $Ł^{(\text{CE})}$ and $Ł^{(\text{TRIPLET})}$.

The network is required to output a mask for a certain source at each iteration, but the order in which they will be extracted when they first appear, is not predictable. Thus, a permutation-invariant loss function is required. Once a source was extracted and the permutation was chosen to minimize the error on its first occurrence, its position is fixed for all following blocks. This is achieved, as explained earlier, by passing the embedding vectors from block to block. Silent target masks $\mathbf{A}_{b,i} = \mathbf{0}$ are inserted when a source was active before, but is silent in the current block.

A permutation-invariant utterance-level mean square error (MSE) loss can be used as in RSAN [8]:

$$Ł^{(\text{MMSE})} = \frac{1}{IB} \sum_{i,b} |\hat{\mathbf{M}}_{i,b} \odot \mathbf{Y}_b - \mathbf{A}_{\phi_b}|^2, \quad (3)$$

where I and B are the total number of iterations and blocks, respectively. $\mathbf{A}$ is the target magnitude spectrogram. The permutation ϕ_b for the b-th block is formed by concatenating the permutation used for the previous block ϕ_{b-1} with the permutation ϕ_b^* that minimizes the separation error for the newly discovered sources in block b:

$$\phi_b = [\phi_{b-1}, \phi_b^*]. \quad (4)$$

To meet the iteration stopping criterion, the following loss function is employed, that pushes the values of the residual mask to 0 if no speaker is remaining [8]:

$$Ł^{(\text{res-mask})} = \sum_{b,tf} \left[\max\left(1 - \sum_i \hat{\mathbf{M}}_{b,i}, 0\right) \right]_{tf} \quad (5)$$

The speaker embedding vectors can be trained with a variety of loss functions. Two possibilities are using an embedding layer followed by a softmax cross-entropy (CE) loss, hereafter called $Ł^{(\text{CE})}$, and a triplet loss $Ł^{(\text{TRIPLET})}$ [20].

The triplet loss ensures the cosine similarity between each pair of embedding vectors for the same speaker is greater than for any pair of vectors of differing speakers. Triplets are formed by first choosing an anchor $\mathbf{a}$, and then for that anchor a positive $\mathbf{p}$ and a negative vector $\mathbf{n}$, which belong to the same and a different speaker than the anchor, respectively, from all embedding vectors of a mini-batch. Based on the cosine similarity $s_i^{\mathbf{an}}$ between the anchor and the negative, and the cosine similarity $s_i^{\mathbf{ap}}$ between the anchor and the positive, the triplet loss for N triplets can be formulated as

$$Ł^{(\text{TRIPLET})} = \sum_{n=1}^{N} \max(s_n^{\mathbf{an}} - s_n^{\mathbf{ap}} + \delta, 0). \quad (6)$$

where δ is a small positive constant.

3. EXPERIMENTS

We evaluate the proposed method in terms of source separation and speaker diarization performance. It is compared with two conventional methods and two simple extension of the conventional method for block processing: (i) PIT and (ii) RSAN applied to the whole mixture (called PIT batch and RSAN batch, hereafter), extensions of RSAN to perform diarization in (iii) block-online and (iv) block-offline manners. These simple extensions are 2-stage methods similar to the conventional methods in [1, 10–13], which, based on NN, first separate the speakers, estimate associated speaker embedding vectors, and then cluster the vectors to estimate the correct association of speaker identity information among blocks. The methods (iii) and (iv) are referred to as online and offline clustering, hereafter. As the clustering method, we use a leader-follower and a hierarchical clustering algorithm for online and offline clustering, respectively. While the offline clustering is performed with the correct number of speakers being given, the other methods estimate it. For reference purpose, we also show the performance of an oracle experiment using the ideal ratio mask (IRM), and a guess-level performance which assumes that exactly one speaker speaks all the time. Throughout the experiments, the block size for all block processing schemes is set to 2.5 seconds, which amounts to about 150 time frames.

3.1. Data

We generated meeting-like training and test data based on utterances taken from the single-channel WSJ0 corpus [21]. To generate a speech mixture, one or two speech signals are mixed at a power ratio uniformly chosen between 0 dB and 5 dB relative to each other. Note, however, that the signals are not reverberant.

We created 55 hours of training data, which were organized as a collection of 10-second (4-block) mixtures. Each mixture was generated such that the first 5 s contain a single or two speakers with a probability of 50 % each, while the second half contains silence/zero speakers, a single speaker or two speakers with a probability of 15 %, 55 % and 30 %, respectively.

For evaluation, we generated 16 hours of testing data in total, which comprises (a) 10 s (4-block) mixtures whose utterance length and mixture characteristics match the training data, (b) 30 s (12-block) long homogeneous mixtures and (c) 30 s (12-block) long conversation-like mixtures. The sets of speakers used for training and test are not overlapping. The homogeneous and conversation-like mixtures are considerably longer than the training data, and thus can be used to test generalization capability of the proposed model. In the homogeneous mixtures, one speaker talks throughout the whole mixture while another one starts speaking randomly in the first half of the mixture and continues speaking till the end. Since there are no cases where a speaker stops speaking in the middle of the test data, the proposed model does not have to remember speakers over silent blocks. The conversation-like mixtures are generated such that the first 5 s of the test utterance contain a single or two speakers (50 % each), while the mixture in the remaining time is generated such that it contains silence/zero, a single or two speakers with a probability of 15 %, 55 % and 30 %, respectively.

3.2. Network Configurations

Each neural network, including PIT and RSAN, had one fully connected layer on top of two BLSTM layers. This is the BLSTM-FC configuration referred to earlier. The speaker embedding dimensionality was set to 128. The speaker embedding estimation network consisted of 3 fully connected layers with 50, 50 and 128 neurons, respectively. The weight for $Ł^{(\text{res-mask})}$ was set to $\alpha = 0.1$.

Table 1. SDR improvement, speaker diarization and speaker confusion error rates

		Model		(a) 4-block			(b) 12-block homogeneous			(c) 12-block conv.-like		
		spk loss	gate	SDR [dB]	DER [%]	SCER [%]	SDR [dB]	DER [%]	SCER [%]	SDR [dB]	DER [%]	SCER [%]
Proposed	1	—	—	**19.4**	4.2	3.1	7.5	5.5	5.3	11.5	7.8	6.5
	2	—	✓	18.5	**4.0**	**2.6**	**7.6**	**5.2**	**4.0**	11.7	**6.6**	4.9
	3	CE	✓	15.8	5.6	3.4	7.3	5.4	6.0	11.6	7.1	6.1
	4	triplet	✓	17.9	4.2	2.9	7.2	5.6	4.8	**11.9**	7.4	5.5
	guess level			—	47.1	25.0	—	45.4	27.4	—	38.8	27.4
	ideal ratio mask (IRM)			28.9	0.8	0.0	14.2	1.0	0.0	24.0	0.8	0.1
Baseline	i	PIT batch		13.3	14.5	4.3	6.8	6.7	5.1	10.9	9.8	**4.4**
	ii	RSAN batch		13.5	7.3	3.7	5.5	9.2	8.3	10.5	10.0	7.4
	iii	online clustering		8.2	15.8	9.8	—	—	—	−10.0	52.1	37.8
	iv	offline clustering		10.9	9.7	4.5	3.2	17.5	7.9	5.3	15.8	6.2

We employed 4 different architectures for the proposed method as in Table 1: Model (1) does not use the gating mechanism while all other models (2), (3) and (4) do. Models (1) and (2) are trained only using the reconstruction and residual mask losses, without a speaker loss ($\beta = \gamma = 0$). Models (3) and (4) use the cross-entropy ($\beta = 0.01, \gamma = 0$) and triplet ($\gamma = 0.1, \beta = 0$) losses, respectively.

3.3. Evaluation Metrics

We evaluate the performance in terms of signal-to-distortion ratio (SDR), diarization error rate (DER) [22], and speaker confusion error rate (SCER). DER indicates the percentage of time that the system outputs speech activity which is wrongly labeled:

$$DER = \frac{\text{\#frames with wrongly estimated speaker}}{\text{total \#frames}} \times 100\% \quad (7)$$

The error consists of missed speaker time (MST), false active time (FAT) and speaker error time (SET). Note that if the system confuses speakers, i.e., it correctly labels speakers as active, but confuses its output order, then this is not considered as an error in DER. Therefore, SCER is additionally introduced as the percentage of time that the system confuses the output order of the speakers:

$$SCER = \frac{\text{\#frames with confused speaker labels}}{\text{total \#frames}} \times 100\% \quad (8)$$

The number of frames with confused speaker labels is determined by comparing the optimal speaker assignment calculated for the whole mixture with the speaker assignment calculated for each frame.

3.4. Results

As in Table 1, the proposed methods outperform all four baselines i) - iv) in most cases in all three tested conditions in terms of SDR, DER and SCER. The conventional two-stage methods, online clustering and offline clustering, failed to find correct association of speaker identity information among blocks, and thus tend to work more poorly as the number of processed blocks increases. As expected, PIT batch and RSAN batch worked significantly better than the two-stage methods. However, interestingly, the proposed method generally worked better than these batch methods even though it performs block-online processing.

Model (2) of the proposed method outperforms model (1) in most scenarios, which shows the effectiveness of the gating function in Fig. 2. Again, interestingly, model (2) also outperforms model (3) and model (4) in almost all scenarios. This suggests that the speaker embedding loss, be it $Ł^{(\text{CE})}$ or $Ł^{(\text{TRIPLET})}$, actually disturbs the embedding process and the optimal adaptation vectors may contain additional information other than the speaker identity, e.g., about interfering signals. Looking at a 2-dim. projection of the embedding space, we noticed that the embedding vectors form fairly condensed clusters for each speaker if an embedding loss is used, while it was not the case otherwise. Last but not least, the models performed very well in source number counting (over 98 % acc. in conv.-like data and over 99 % acc. in all other cases), which is reflected in low DER. Some demos of the proposed method are available at [23].

4. RELATION TO PRIOR WORKS

Some researchers tried block-processing and online-processing based on NN-based source separation. However, none of them has the capability of performing diarization in realistic situations where speakers can stop talking in the middle of the meeting and remain silent for some time before they start talking again. In [24], PIT is applied to each block, and then associations of estimated masks between adjacent blocks are estimated by a simple cross-correlation scheme. Clearly it cannot track speaker characteristics over silent blocks. A method proposed in [25] performs source separation in a frame-by-frame manner, by exploiting temporal dependencies and continuity of the speech signal. Specifically, a certain number of past frames of separated signals is used as additional input to a NN which outputs separated signals for the current frame such that they can smoothly continue from past context data. While it is similar to the proposed method in a sense that it utilizes speaker information appearing in the past, it cannot deal with long silent regions since it can see only a limited past context of fixed length, e.g., 600 ms in [25]. On the other hand, the proposed method can naturally handle arbitrarily long silent regions, which we believe is a very important property when dealing with real meeting scenarios.

5. CONCLUSIONS

In this paper, we proposed an all-neural mask estimator which is capable of block-online processing and which can adaptively change the number of output separation masks in each block. It can track speakers even through silent blocks and detect new speakers in every block. The experiments confirmed that the proposed method shows promising performance, both in terms of separation performance and in terms of diarization and speaker confusion error performance.

6. REFERENCES

[1] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 356–370, Feb 2012.

[2] N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman, *First DIHARD Challenge Evaluation Plan*, 2018, https://zenodo.org/record/1199638.

[3] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, , and P. Wellner, "The AMI meeting corpus: A pre-announcement," in *The Second International Conference on Machine Learning for Multimodal Interaction, ser. MLMI'05*, 2006, pp. 28–39.

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

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

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

[7] M. Kolbæk, D. Yu, Z. 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, Oct 2017.

[8] K. Kinoshita, L. Drude, M. Delcroix, and T. Nakatani, "Listening to each speaker one by one with recurrent selective hearing networks," in *Proc. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 5064–5068.

[9] A. Graves, "Adaptive computation time for recurrent neural networks," 2016, arXiv:1603.08983.

[10] S. Araki, H. Sawada, and S. Makino, "Blind speech separation in a meeting situation with maximum SNR beamformers," in *Proc. 2007 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2007, vol. 1, pp. I–41–I–44.

[11] S. Araki, M. Fujimoto, K. Ishizuka, H. Sawada, and S. Makino, "Speaker indexing and speech enhancement in real meetings / conversations," in *Proc. 2007 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2008, vol. 1, pp. 93–96.

[12] M. Diez, F. Landini, L. Burget, J. Rohdin, A. Silnova, K. Zmolikova, O. Novotný, K. Veselý, O. Glembek, O. Plchot, L. Mošner, and P. Matějka, "BUT system for DIHARD speech diarization challenge 2018," in *Proc. Interspeech 2018*, 2018, pp. 2798–2802.

[13] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe, and S. Khudanpur, "Diarization is hard: Some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge," in *Proc. Interspeech 2018*, 2018, pp. 2808–2812.

[14] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, , and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 19(4), pp. 788–798, 2011.

[15] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, , and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Proc. IEEE Spoken Language Technology Workshop*, 2016.

[16] L. Drude, T. Higuchi, K. Kinoshita, T. Nakatani, and R. Haeb-Umbach, "Dual frequency- and block-permutation alignment for deep learning based block-online blind source separation," in *Proc. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 691–695.

[17] D. Kounades-Bastian, L. Girin, X. Alameda-Pineda, R. Horaud, and S. Gannot, "Exploiting the intermittency of speech for joint separation and diarization," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Oct 2017, pp. 41–45.

[18] M. Delcroix, K. Zmolikova, K. Kinoshita, A. Ogawa, and T. Nakatani, "Single channel target speaker extraction and recognition with speaker beam," in *Proc. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5554–5558.

[19] M. Delcroix, K. Zmolikova, T. Ochiai, K. Kinoshita, S. Araki, and T. Nakatani, "Compact network for SpeakerBeam target speaker extraction," in *Proc. 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, (submitting).

[20] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep Speaker: an end-to-end neural speaker embedding system," 2017, arXiv:1705.02304v1.

[21] J. Garofolo, D. Graff, P. Doug, and D. Pallett, *CSR-I (WSJ0) Complete LDC93s6a*, Linguistic Data Consortium, Philadelphia, New Jersey, 1993.

[22] NIST Speech Group, "Spring 2007 (rt-07) rich transcription meeting recognition evaluation plan," 2007.

[23] http://www.kecl.ntt.co.jp/icl/signal/kinoshita/publications/ICASSP19/online_RSAN_demo/index.html.

[24] T. Yoshioka, H. Erdogan, Z. Chen, X. Xiao, and F. Alleva, "Recognizing overlapped speech in meetings: A multichannel separation approach using neural networks," in *Proc. Interspeech 2018*, 2018, pp. 3038–3042.

[25] Z.-X. Li, Y. Song, L.-R. Dai, and I. McLoughlin, "Source-aware context network for single-channel multi-speaker speech separation," in *Proc. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2018, pp. 681–685.