SINGLE CHANNEL TARGET SPEAKER EXTRACTION AND RECOGNITION WITH SPEAKER BEAM

Marc Delcroix[1], *Katerina Zmolikova*[2], *Keisuke Kinoshita*[1], *Atsunori Ogawa*[1], *Tomohiro Nakatani*[1]

[1]NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan
[2]Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Czechia
marc.delcroix@lab.ntt.co.jp

ABSTRACT

This paper addresses the problem of single channel speech recognition of a target speaker in a mixture of speech signals. We propose to exploit auxiliary speaker information provided by an adaptation utterance from the target speaker to extract and recognize only that speaker. Using such auxiliary information, we can build a speaker extraction neural network (NN) that is independent of the number of sources in the mixture, and that can track speakers across different utterances, which are two challenging issues occurring with conventional approaches for speech recognition of mixtures. We call such an informed speaker extraction scheme "SpeakerBeam". SpeakerBeam exploits a recently developed context adaptive deep NN (CADNN) that allows tracking speech from a target speaker using a speaker adaptation layer, whose parameters are adjusted depending on auxiliary features representing the target speaker characteristics. SpeakerBeam was previously investigated for speaker extraction using a microphone array. In this paper, we demonstrate that it is also efficient for single channel speaker extraction. The speaker adaptation layer can be employed either to build a speaker adaptive acoustic model that recognizes only the target speaker or a mask-based speaker extraction network that extracts the target speech from the speech mixture signal prior to recognition. We also show that the latter speaker extraction network can be optimized jointly with an acoustic model to further improve ASR performance.

Index Terms— Speech Recognition, Speech mixtures, Speaker extraction, Adaptation, Robust ASR

1. INTRODUCTION

With the deployment of speech driven home devices, there has been an increased interest for noise robust automatic speech recognition (ASR) [1, 2]. Recently, significant progress has been made exploiting microphone arrays by combining traditional signal processing approaches with deep learning [3–6]. In contrast, single channel robust speech recognition remains a challenging task [7], especially in presence of interfering speakers.

There has been much research aiming at separating speech signals observed in a mixture using deep learning [8–10]. Initial attempts proposed to train a DNN to output as many signals as there is in the mixture. However, such approaches present several limitations. First they are limited to mixture composed of signals with distinct characteristics such as different genders [8]. Indeed, without such constraints it is not possible to control which output corresponds to which speaker and therefore the models are difficult to train. We call this problem the *frame level permutation problem*. In addition, these approaches impose a hard constraint on the number of speakers it can handle as it is fixed by the architecture of the network and thus these approaches can be hard to generalize to unknown number of speakers.

Recently, deep clustering [9] and deep attractor networks [11] have been proposed to release these limitations. They solve the *frame level permutation problem* by learning a DNN that outputs embeddings for time-frequency bins, such that time-frequency corresponding to a same speaker are close to each other in the embedding space. Speech separation masks for each source can then be computed by clustering the embedding vectors. Although the DNN does not have a hard constraint on the number of speakers in the mixtures, the clustering step requires knowing or estimating the number of speakers. Moreover, there remains a *permutation problem across utterances*, as there is no guarantee that embedding vectors for a given speaker will take similar values across different processing segments.

Permutation invariant training [12] is another approach, which mitigates the *frame level permutation problem* at the training stage, by modifying the training objective function such that labels are permuted to find the closest match with the output of the DNN. The learned model can separate and track speakers within an utterance [13], and generalize to unknown number of speakers [14]. Moreover, permutation invariant training can be easily used to jointly optimize a speech separation and acoustic model [13, 15]. However, the *permutation problem across utterances* remains unaddressed.

We have recently proposed an alternative approach for recognizing speech in mixtures using a microphone array. Instead of aiming at separating all signals and recognizing them, we focus on building a speaker extraction DNN that extracts only a target signal. We employ an adaptation utterance consisting of recordings of the target speaker voice only, to inform the speaker extraction DNN about which speaker to extract. We call such a scheme, SpeakerBeam. In [16], we showed that a key to achieve high target speaker extraction performance was to employ a context adaptive DNN (CADNN) architecture proposed for speaker adaptation of the acoustic model [17], which can adjust its parameters depending on auxiliary features representing the target speaker characteristics. Since SpeakerBeam only outputs a target speaker, it does not assume any knowledge of the number of sources present in the mixture. Moreover, as it can track the target speaker across utterances, it can solve the *permutation problem globally*. Note that being able to track a target speaker across utterances has very practical implications. For example, it opens the possibility of building personalized home devices that can focus on recognizing commands or speech from a target speaker, e.g. the owner of the device.

In this paper, we explore the extraction capabilities of SpeakerBeam in single channel case. We investigate three different configurations, i.e. an adaptive acoustic model performing recognition of the target speaker (SpeakerBeam-AM), a mask-based speaker extraction front-end (SpeakerBeam-FE), and a joint system combining

mask-based speaker extraction and recognition (SpeakerBeam-JT).

The remainder of the paper is organized as follows. In Section 2 we introduce the problem and present the principles of single channel SpeakerBeam and its different implementations. Section 3 discusses relation with prior works. We then report experimental results in Section 4 and conclude the paper in Section 5

2. SINGLE CHANNEL SPEAKERBEAM

Let us first introduce the problem and describe the different SpeakerBeam configurations that we investigate in this paper.

2.1. Problem formulation

We model the observed mixture signal in the short-term Fourier transform (STFT) domain, $Y(t,f)$, as,

$$Y(t,f) = X^{(s)}(t,f) + N(t,f), \tag{1}$$

where $X^{(s)}(t,f)$ is the speech signal corresponding to the target speaker s, $N(t,f)$ is the interference signal consisting of interference speakers and background noise, (in the experiments we only considered interfering speakers), and t and f are time and frequency indexes, respectively. We denote by $\mathbf{y}_t$ the feature vector containing the log mel filterbank coefficients of signal $Y(t,f)$. We aim at recognizing only the target speech $X^{(s)}(t,f)$ out of the mixture signal.

2.2. SpeakerBeam front-end (SpeakerBeam-FE)

We treat the target speaker extraction task as a speaker adaptation of a speech extraction DNN, which inputs speech features of the observed mixture signal and outputs a time-frequency mask that extracts the target speaker out of the observed mixture. In [18], such masks were used in a microphone array configuration to compute beamformer coefficients. Here, we focus on the single microphone configuration, where the time-frequency masks obtained from the DNN are simply applied to the mixture to estimate the target speech as,

$$\hat{X}^{(s)}(t,f) = M^{(s)}(t,f)Y(t,f), \tag{2}$$

where $\hat{X}^{(s)}(t,f)$ is an estimated target speech and $M^{(s)}(t,f)$ is a time-frequency mask computed with the speaker extraction DNN.

Without any guidance, the speaker extraction neural network has no way of knowing which signal in the mixture is the target. Therefore, we use an adaption utterance to extract speaker characteristics and guide the system. The adaptation utterance $A^{(s)}(t,f)$ consists of a speech signal containing only the target speaker and differs from the target speech in the mixture.

We use a speaker adaptive layer described below as one of the layers of a DNN to adapt the speech extraction DNN to the target speaker.

2.2.1. Speaker adaptation layer

We have recently proposed a CADNN as an alternative approach for auxiliary feature-based DNN adaptation [17, 19]. A CADNN is a DNN, which has a speaker adaptation layer as shown in Figure 1, which consists of a weighted sum of the contribution of sub-layers,

$$\mathbf{h}^{(o)} = \sigma\left(\sum_{m=1}^{M} \alpha_m^{(s)} F_m\left(\mathbf{h}^{(i)}\right)\right), \tag{3}$$

where $\mathbf{h}^{(i)}$ and $\mathbf{h}^{(o)}$ are the input and output of the adaptation layer, respectively, $F_m^{(l)}$ is a transformation of the layer input, $\alpha_m^{(s)}$ is an adaptation weight associated with the target speaker s, m is the index of the sub-layer, M is the number of sub-layers, and $\sigma(\cdot)$ is an activation function such a sigmoid or ReLU. Here we use affine transformations and $F_m\left(\mathbf{h}^{(i)}\right) = \mathbf{W}_m\mathbf{h}^{(i)} + \mathbf{b}_m$, where $\mathbf{W}_m$ and $\mathbf{b}_m$ are weight matrices and bias vectors, respectively. The speaker adaption layer can greatly modify the network behavior because it performs adaptation of both the bias and the weight matrix, which is needed to extract the target speaker.

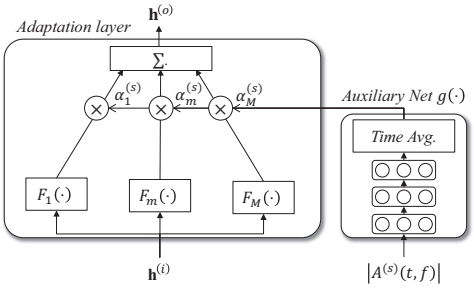


Fig. 1. Schematic diagram of the speaker adaptation layer and the sequence summary auxiliary network.

2.2.2. Sequence summarization for adaption weight computation

The behavior of the adaptive layer is governed by the adaptation weights $\alpha_m^{(s)}$, which allow the network to adapt itself for extracting specifically the target speaker. We derive these adaptation weights directly from the adaptation utterance, $A^{(s)}(t,f)$, using the sequence summary scheme proposed in [18, 20] as,

$$\boldsymbol{\alpha}^{(s)} = \frac{1}{T_A}\sum_{t=1}^{T_A} g(|A^{(s)}(t,f)|), \tag{4}$$

where $\boldsymbol{\alpha}^{(s)} = [\alpha_1^{(s)}, \ldots, \alpha_M^{(s)}]$ is a vector containing the adaptation weights for speaker s, T_A is the length of the adaptation utterance and $g(\cdot)$ is an auxiliary neural network that inputs the amplitude spectrum of the adaptation utterance, $|A^{(s)}(t,f)|$.

Note that $g(\cdot)$ is trained jointly with the main network. Directly computing the target speaker dependent weights $\boldsymbol{\alpha}^{(s)}$ from the adaptation utterance avoids using intermediate feature representations of the speakers such as i-vectors, and thus provides a speaker representation that is optimal for the speaker extraction task [18].

2.3. SpeakerBeam with joint training (SpeakerBeam-JT)

SpeakerBeam-FE learns the speaker extraction DNN by minimizing the cross entropy w.r.t ideal binary masks (IBMs) [18]. However, this may not be optimal for recognition. For example, the obtained masks may suppress important information for the recognizer, or excessively leak the interfering speakers. Solving such mismatch between a speech enhancement front-end and an ASR back-end has been addressed by jointly training both modules [15, 21, 22].

Here, we apply a similar strategy as [22], i.e. we connect the speech extraction DNN and an acoustic model with a deterministic feature extraction module that converts the extracted speech spectrum to log-mel filterbank coefficients with context.

2.4. SpeakerBeam acoustic model (SpeakerBeam-AM)

In [15], it was shown that permutation invariant training could be used to directly train an acoustic model to perform separation and

recognition. These results suggest that an acoustic model well adapted to a target speaker may focus on recognizing only the speech signal from the target speaker and ignore the interferences. An alternative to SpeakerBeam-JT is to simply adapt an acoustic model using the target speaker characteristics.

We propose using a speaker adaptation layer as one of the layers of an acoustic model to make it speaker adaptive. We use the speaker adaptation layer and sequence summary scheme described in Sections 2.2.1 and 2.2.2. SpeakerBeam-AM is similar to our previous work on acoustic model adaptation with CADNN [19], with the difference that the adaptation weights are derived using the sequence summary scheme instead of i-vectors.

3. RELATION TO PRIOR WORK

There have been many studies on adaptation of DNN-based acoustic models exploiting auxiliary features [19, 20, 23, 24]. Conventional approaches simply concatenate the auxiliary feature to the input of a DNN (auxiliary input DNN) [20, 23, 24]. However, simply inputting the speaker representation to the input of the network realizes only bias adaptation of the input layer, which may be insufficient to guide the network to extract the target speaker [16].

In [24], a related scheme was proposed to extract a speaker representation from a wake-up keyword for home assistant. The speaker representation was used for end-point detection and acoustic model adaptation. However, they employed the last output of an LSTM as speaker representation and use it as an auxiliary input feature to a DNN-based acoustic model. In our preliminary experiments, we observed superior performance using the simple averaging operation of Eq. (4), which may in our case better capture the overall speaker characteristics since the adaptation utterances are relatively long.

Joint training of a speech enhancement DNN and an acoustic model have been investigated for single and multi-channel cases [15, 22, 25]. With SpeakerBeam, we also jointly train the auxiliary network that computes speaker characteristics, aiming at obtaining speaker representation optimal for the target speech recognition. Note that, in parallel to this work, we have been investigating joint training of SpeakerBeam for microphone arrays [26].

4. EXPERIMENTS

We tested the different SpeakerBeam configurations using mixtures of two speakers. Figure 2 illustrates the SpeakerBeam configurations we investigated, and details their network architectures.

4.1. Data

To evaluate the proposed method, we created single channel speech mixtures using recordings from the Wall Street Journal (WSJ) corpus [27]. We used 7138 utterances from 83 speakers for the training set, 410 utterances from 10 speakers in the development set and 330 utterances from 10 speakers in the evaluation set. For each utterance, we mixed an interference utterance from a different speaker within the same set. The training set was mixed with signal-to-interference ratio (SIR) of 0 dB on average. It is the same as that used in [26]. To evaluate performance for various input SIR conditions, we created 5 development and evaluation sets varying only the SIR between 0 and 20 dBs. In this preliminary experiments, all recordings included moderate reverberation (about 0.2 sec), but no background noise.

For each mixture, we randomly chose an adaptation utterance from the target speaker (different than the utterance in the mixture). The length of the adaptation utterance was about 10 sec on average.

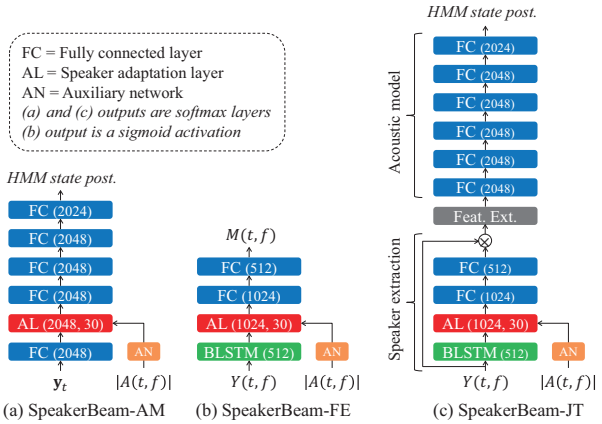


Fig. 2. Schematic diagram of the three different SpeakerBeam configurations. The numbers in the parentheses indicate the number of nodes and the number of sub-layers for the adaptation layer. The auxiliary networks consisted of two FC layers with 50 nodes and ReLU activations and an output FC layer with a linear activation followed by the averaging operation. We used ReLU for all hidden layer activation functions.

4.2. Settings

4.2.1. *Baseline acoustic model*

The baseline acoustic model consisted of 5 fully connected hidden layers with 2048 nodes and ReLU activations functions. The output layer had 2024 nodes corresponding to the HMM states. This model was trained on single speaker recordings with alignments obtained from a GMM-HMM system. The input of the acoustic model consists of 40 log mel filterbank coefficients with a context extension window of 11 frames. The features were mean normalized per utterance. The AM and all other models were trained using the ADAM optimizer [28]. As a comparison, we also tested the auxiliary input feature based adaptation (auxiliary input AM) using speaker characteristics obtained by processing the adaptation utterance with the sequence summary scheme of Section 2.2.2.

4.2.2. *SpeakerBeam-AM*

SpeakerBeam-AM used a network architecture similar to the baseline acoustic model, but with its second layer replaced with an adaptation layer as shown in Fig. 2-(a). The input of the network consisted of the speech features of the mixture signals. The input of the auxiliary network consisted of the 401 dimension amplitude spectrum coefficients of the adaption utterance. The weights of the SpeakerBeam-AM were initialized with those of the baseline AM as it improved performances slightly compared to starting from a randomly initialized model.

4.2.3. *SpeakerBeam-FE*

The configuration of SpeakerBeam-FE is shown in Fig. 2-(b). The input of SpeakerBeam-FE consisted of 401 dimension amplitude spectrum computed using a STFT with a window size of 25 msec and a 10 msec shift. The speaker extraction DNN was trained to minimize the cross entropy w.r.t IBMs. In the SpeakerBeam-FE recognition experiments, we retrained the baseline acoustic model on the training mixture signals processed with SpeakerBeam-FE.

Table 1. WER as a function of the input SIRs for the eval set. WER a single speaker recognized with the baseline AM was 4.1 %.

	0dB	5dB	10dB	15dB	20dB
Mixture w/ baseline AM	95.7	70.4	40.3	14.0	5.9
Auxiliary input AM	85.2	72.6	66.5	70.5	76.8
SpeakerBeam-AM	45.8	28.3	20.3	18.1	17.3
SpeakerBeam-FE	54.5	39.7	32.8	30.0	29.2
SpeakerBeam-JT	34.0	17.5	9.8	7.5	6.5

4.2.4. SpeakerBeam-JT

The configuration of SpeakerBeam-JT is shown in Fig. 2-(c). Except otherwise mentioned, the parameters of SpeakerBeam-JT were initialized with pre-trained modules, i.e. the mask estimation network of SpeakerBeam-FE and the baseline acoustic model trained on single speaker speech.

4.3. Results

Table 1 shows the word error rate (WER) for the eval sets as a function of the input SIRs. We used the development set to choose the best decoding configuration (language model weight). We omitted the results on the dev set because they exhibited similar tendency.

The baseline results were obtained by recognizing the single speaker speech and the mixture with the baseline AM trained on single speaker speech. Not surprisingly, recognizing the mixture signal is very challenging especially for low input SIRs. In addition, using the target speaker representation at the input of the acoustic model (Auxiliary input AM) fails to improve performance on this task. This indicates that the simple bias adaptation is insufficient to track the target speaker.

The following three rows of Table 1 show the WERs of the different SpeakerBeam configurations. Both SpeakerBeam-AM and SpeakerBeam-FE greatly reduce WER compared to the mixture results. Moreover, SpeakerBeam-JT can further greatly improve performance at higher input SIRs. Comparing the results of SpeakerBeam-AM with those of the AM with auxiliary input features confirms that the speaker adaptation layer is essential for speaker extraction. It is noticeable that even if SpeakerBeam-AM is a relatively simple model that does not use any BLSTM, it already significantly improves performance over the baseline and outperforms SpeakerBeam-FE. Investigation of SpeakerBeam-AM with more powerful architectures will be a part of our future works.

Performance of both SpeakerBeam-AM and SpeakerBeam-FE stopped improving significantly for input SIR above 10dBs. This may be due to a mismatch between the training and testing conditions, because the training data mostly cover input SIR around 0 dB. For SpeakerBeam-FE, the training criterion mismatch (IBM vs ASR criterion) also appears to contribute to the poor performance at higher input SIR, and SpeakerBeam-JT mitigates this issue.

For SpeakerBeam-JT, we used pre-trained modules to initialize the model parameters. Such a pre-training improves performance, however, we could still observe reasonable performance even when all modules were randomly initialized, e.g. WERs of 35.1 % for input SIRs of 0 dB.

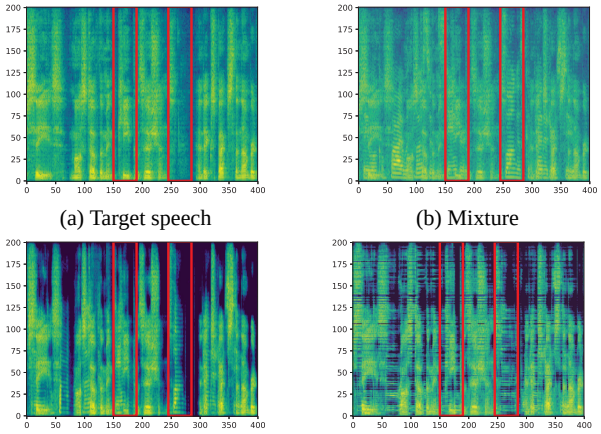
4.4. Discussions

To better appreciate the difference between SpeakerBeam-FE and SpeakerBeam-JT we compare recognition results and spectrograms of the extracted speech signals for one utterance of the eval set at input SIR of 0 dB. Using SpeakerBeam-FE the recognized sentence includes several recognition errors shown in the underline text:

> "The company has five hundred Japanese managers overseas most of *** *** *** the position expects the number to rise sixty percent in the next five years".

With SpeakerBeam-JT the utterance was correctly recognized as:

> "The company has five hundred Japanese managers overseas most of them in key positions and expects the number to rise sixty percent in the next five years".



(a) Target speech (b) Mixture

(c) SpeakerBeam-FE (d) SpeakerBeam-JT

Fig. 3. Spectrograms corresponding to utterance "440c0202" of the eval set at input SIR of 0 dB.

Figure 3 plots the spectrograms of the single target speech, the mixture and the extracted speech with SpeakerBeam-FE and SpeakerBeam-JT for the portion of the utterance around the underlined text. As shown in Fig.3-(c), SpeakerBeam-FE can reduce the interference signal and outputs a relatively smooth spectrum. However, some parts of the interference signals are still present as shown in the areas marked with red rectangles. In contrast, the spectrogram obtained with SpeakerBeam-JT is less smooth, but reduces further the interfering speaker. This appears to be better in terms of recognition performance.

5. CONCLUSION

In this paper we have investigated three SpeakerBeam configurations for target speaker extraction and recognition using a single microphone. These schemes exploit a speaker adaptation layer, which enables great control of the model parameters depending on the target speaker characteristics and therefore allows tracking only a target speaker in a mixture. We showed that a speaker adaptive acoustic model could address the problem to some extent, but that best performance was obtained when jointly training SpeakerBeam-FE with an acoustic model.

Although SpeakerBeam does not make explicit use of the number of speakers in the mixture, in this preliminary study, we have focused our experiments on mixture of two speakers. Future works will include investigations with various conditions in terms of the number of speakers in the mixture and in presence of background noise as well as extended training data. Moreover, we will also investigate further improvements by combining SpeakerBeam-FE and SpeakerBeam-AM so that both modules become speaker adaptive.

6. REFERENCES

[1] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE Trans. ASLP*, vol. 22, no. 4, pp. 745–777, 2014.

[2] S. Watanabe, M. Delcroix, F. Metze, and J. R. Hershey (Editors), *New Era for Robust Speech Recognition: Exploiting Deep Learning*, Springer (in press), 2017.

[3] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *Proc. of ICASSP'16*, 2016, pp. 196–200.

[4] H. Erdogan, J. R. Hershey, S. Watanabe, M. Mandel, and J. Le Roux, "Improved MVDR beamforming using single-channel mask prediction networks," in *Proc. of Interspeech'16*, 2016.

[5] X. Xiao, S. Watanabe, H. Erdogan, L. Lu, J. Hershey, M. L. Seltzer, G. Chen, Y. Zhang, M. Mandel, and D. Yu, "Deep beamforming networks for multi-channel speech recognition," in *Proc. of ICASSP'16*, 2016.

[6] T. N. Sainath, R. J. Weiss, K. W. Wilson, B. Li, A. Narayanan, E. Variani, M. Bacchiani, I. Shafran, A. Senior, K. Chin, A. Misra, and C. Kim, "Multichannel signal processing with deep neural networks for automatic speech recognition," *IEEE/ACM Trans. ASLP*, vol. 25, no. 5, pp. 965–979, May 2017.

[7] "The 4th ChiME Speech Separation and Recognition Challenge," `http://spandh.dcs.shef.ac.uk/chime_challenge/chime2016/results.html`, Cited Oct. 24 2017.

[8] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *arxiv*, 2017.

[9] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. of ICASSP'16*, 2016, pp. 31–35.

[10] C. Weng, D. Yu, M. L. Seltzer, and J. Droppo, "Deep neural networks for single-channel multi-talker speech recognition," *IEEE Trans. ASLP*, vol. 23, no. 10, pp. 1670–1679, 2015.

[11] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *Proc. of ICASSP'17*, 2017.

[12] Y. Dong, K. Morten, T. Zheng-Hua, and J. Jesper, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. of ICASSP'17*, 2017.

[13] Z. Chen, J. Droppo, J. Li, and W. Xiong, "Progressive joint modeling in unsupervised single-channel overlapped speech recognition," *arxiv*, vol. abs/1707.07048, 2017.

[14] M. Kolbaek, D. Yu, Z.-H. Tan, and J. Jensen, "Joint separation and denoising of noisy multi-talker speech using recurrent neural networks and permutation invariant training," in *Proc. of MLSP'17*, 2017.

[15] Y. Qian, X. Chang, and D. Yu, "Single-channel multi-talker speech recognition with permutation invariant training," *arxiv*, vol. abs/1707.06527, 2017.

[16] K. Zmolikova, M. Delcroix, K. Kinoshita, T. Higuchi, A. Ogawa, and T. Nakatani, "Speaker-aware neural network based beamformer for speaker extraction in speech mixtures," in *Proc. of Interspeech'17*, 2017.

[17] M. Delcroix, K. Kinoshita, C. Yu, A. Ogawa, T. Yoshioka, and T. Nakatani, "Context adaptive deep neural networks for fast acoustic model adaptation in noisy conditions," in *Proc. of ICASSP'16*. IEEE, 2016, pp. 5270–5274.

[18] K. Zmolikova, M. Delcroix, K. Kinoshita, T. Higuchi, A. Ogawa, and T. Nakatani, "Learning speaker representation for neural network based multichannel speaker extraction," in *Proc of ASRU'17*, Dec 2017.

[19] M. Delcroix, K. Kinoshita, T. Hori, and T. Nakatani, "Context adaptive deep neural networks for fast acoustic model adaptation," in *Proc. of ICASSP'15*, 2015, pp. 4535–4539.

[20] K. Vesely, S. Watanabe, K. Zmolikova, M. Karafiat, L. Burget, and J. H. Cernocky, "Sequence summarizing neural network for speaker adaptation," in *Proc. of ICASSP'16*, 2016, pp. 5315–5319.

[21] J. R. Hershey, S. J. Rennie, P. A. Olsen, and T. T. Kristjansson, "Super-human multi-talker speech recognition: A graphical modeling approach," *Computer Speech and Language*, vol. 24, no. 1, pp. 45 – 66, 2010.

[22] A. Narayanan and D. Wang, "Improving robustness of deep neural network acoustic models via speech separation and joint adaptive training," *IEEE/ACM Trans. ASLP*, vol. 23, no. 1, pp. 92–101, Jan 2015.

[23] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Proc. of ASRU'13*, 2013, pp. 55–59.

[24] B. King, I-F. Chen, Y. Vaizman, Y. Liu, R. Maas, S. H. K. Parthasarathi, and B. Hoffmeister, "Robust speech recognition via anchor word representations," in *Proc. of Interspeech'17*, 2017.

[25] J. Heymann, L. Drude, C. Boeddeker, P. Hanebrink, and R. Haeb-Umbach, "Beamnet: End-to-end training of a beamformer-supported multi-channel asr system," in *Proc. of ICASSP'17*, March 2017, pp. 5325–5329.

[26] K. Zmolikova, M. Delcroix, K. Kinoshita, T. Nakatani, and J. Cernocky, "Optimization of speaker-aware multichannel speaker extraction with asr criterion," in *Proc. of ICASSP'18*, 2018.

[27] J. Garofolo, "CSR-I (WSJ0) Complete LDC93S6A," `https://catalog.ldc.upenn.edu/ldc93s6a`, 1993.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arxiv*, vol. abs/1412.6980, 2014.