LISTENING TO EACH SPEAKER ONE BY ONE WITH RECURRENT SELECTIVE HEARING NETWORKS

Keisuke Kinoshita[1], *Lukas Drude*[1,2], *Marc Delcroix*[1], *Tomohiro Nakatani*[1]

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

ABSTRACT

Deep learning-based single-channel source separation algorithms are currently being actively investigated. Among them, Deep Clustering (DC) and Deep Attractor Networks (DANs) have made it possible to separate an arbitrary number of speakers. In particular, they cleverly combine a neural network and a K-means clustering algorithm to obtain source separation masks with the assumption that the correct number of speakers at the test time is known in advance. Unlike DC and DAN, Permutation Invariant Training (PIT) was proposed as a purely neural network-based mask estimator. Essentially, however, PIT can deal with only a fixed number of speakers, given the strong relationship between the dimensions of the output nodes and the assumed number of sources. Considering these limitations and merits of such conventional methods, this paper proposes a purely neural-network based mask estimator that can handle an arbitrary number of sources, and simultaneously estimate the number of sources in the test signal. To accomplish this, while the conventional methods deal with the source separation problem as a one-pass problem, we cast the problem as a recursive multi-pass source *extraction* problem based on a recurrent neural network (RNN) that can learn and determine how many computational steps/iterations have to be performed depending on the input signals. In this paper, we describe our proposed method in detail, and experimentally show its efficacy in terms of source separation and source counting performance.

Index Terms— Blind source separation, neural network, arbitrary number of sources, attention, source counting.

1. INTRODUCTION

Recently, ASR technologies have progressed greatly [1, 2] and are being used increasingly in our daily lives, and so it is becoming more and more important to handle realistic tasks such as meeting recognition with distant microphones. In such tasks, target speech signals recorded with distant microphones are often covered by noise and speech signals concurrently spoken by other speakers. To overcome this problem, much research has been undertaken on blind source separation (BSS) algorithms.

In general, there are two different ways of addressing the BSS problem, i.e. multichannel methods [3–5] leveraging spatial characteristics, and single channel methods exploiting spectral characteristics [6–10]. This paper focuses on the latter.

Recent single-channel source separation research has tended to utilize deep neural networks (NNs) because of its higher performance compared to conventional approaches based on e.g. non-negative matrix factorization [6]. For example, Deep Clustering [7] (DC) and the Deep Attractor Network (DAN) [8] are recently proposed effective single-channel BSS algorithms. They can be seen as two-stage algorithms. These algorithms first encode an input spectrogram into an embedding space based on a pretrained NN, and output embedding vectors for each time-frequency (T-F) bin. Then, to obtain source separation masks, these embedding vectors are clustered by means of e.g. K-means clustering or a mixture model, given the correct number of clusters equal to the true number of speakers. These approaches are able to generalize to unseen speakers, and importantly do not explicitly assume a fixed number of speakers in the encoding stage. However, there are still some unresolved issues related to these methods, such as (a) how to estimate the correct number of speakers/clusters at the clustering stage [11], and (b) how to select an appropriate clustering algorithm that can model the distribution of the embedding vectors in an optimal sense.

Permutation Invariant Training (PIT) was proposed in [10] as an alternative to DC and DAN for solving the single-channel BSS problem. Unlike DC and DAN, PIT is designed to directly output source separation masks without an explicit clustering step. On the assumption that the number of sources to be separated during a test is known in advance when training NNs, the PIT network has output nodes corresponding to the dimension of the separation mask times the number of sources to be separated. In other words, the current PIT framework is not capable of dealing with an arbitrary number of sources, unlike DC and DAN.

Considering these limitations and merits of the conventional methods [7–10], this paper proposes an NN-based mask estimator that can handle an arbitrary number of speakers and adaptively change the number of output masks depending on the input signal. The proposed network is predicated on a recurrent neural network (RNN), which can learn and determine how many computational steps/iterations have to be performed [12]. By the nature of such networks, at the first iteration, the proposed method can output a mask for a certain speaker/source in the observed spectrogram. Then, in the next iteration it automatically attends to the remaining part of the observed signal and outputs another mask for a different speaker/source. This process is repeated until all sources in the observed mixture have been extracted. In this paper, we call this network a recurrent selective hearing network. While the conventional methods [7–10] tend to deal with the BSS problem as a one-pass separation problem, we cast it as a *recursive multi-pass source extraction* problem. In the remainder of this paper, we first formulate our proposed method, and then evaluate its performance in comparison with a variant of PIT [9].

2. PROPOSED METHOD

This section provides a detailed explanation of our proposed method. We start Section 2.1 by explaining how the proposed method estimates the source separation masks of an arbitrary number of sources. Then, in Section 2.2, we describe the network training procedures.

978-1-5386-4658-8/18/$31.00 ©2018 IEEE

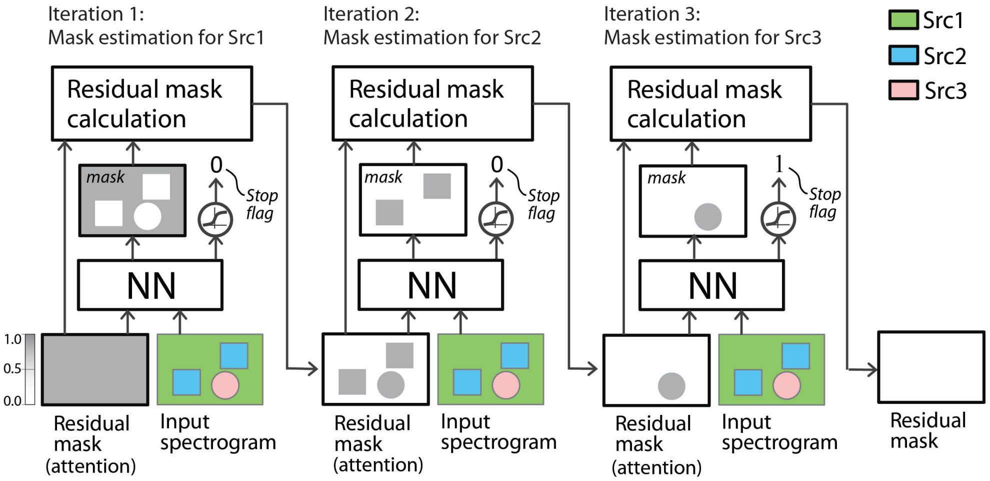

Fig. 1. Overview of the proposed framework. The residual mask calculation block subtracts the estimated mask from the previous residual mask. Naturally, the system simply attends to the remaining portion of the input spectrogram.

2.1. Test (decoding) step of proposed method

Figure 1 summarizes the overall structure, inputs and outputs of the proposed method namely, the recurrent selective hearing network. It runs multiple mask estimation steps iteratively, while judging at each iteration whether or not it should proceed to the next iteration by monitoring statistics provided in the mask estimation process.

At the first iteration, the network receives two inputs, namely the amplitude spectrogram of the observed signal $\mathbf{Y}$ and a residual mask filled with ones ($\mathbf{R}_1 = \mathbf{1}$). This residual mask can be seen as an *attention map* since it controls where to attend. Initially, the network pays attention to all the regions of the input spectrogram. However, as iterations proceed, it focuses more on particular regions of the input spectrogram. Given the input signals, the network decides on its own which source to extract, and estimates a source separation mask $\hat{\mathbf{M}}_1$ for that source. Fig. 1 shows a situation where Src1 is extracted at the first iteration. At the same time, the network can optionally output a stop flag $\hat{z}_1$ indicating whether the iteration process should stop ($\hat{z}_1 = 1$) or not ($\hat{z}_1 = 0$). Then, finally, the first iteration ends with the generation of another residual mask for the next iteration by subtracting the estimated mask from the input residual mask as $\mathbf{R}_2 = \mathbf{R}_1 - \hat{\mathbf{M}}_1$. The residual mask is used to steer the attention of the NN to the remaining part of the input spectrogram to extract the other sources that were not extracted by the previous iterations.

At the second iteration, the network employs the residual mask $\mathbf{R}_2$ and an input spectrogram $\mathbf{Y}$ as its input to estimate a mask for another source. Then, it follows exactly the same procedure as the first iteration. Note that we use the same network for all iterations.

At the third iteration where the network is assumed to stop its iteration, the network decides to output the stop flag of one and calculates the final residual mask. During the test, judgement concerning whether it should stop further iterations can be made by (a) threshold processing on statistics (e.g. mean or median) of the residual mask $\mathbf{R}_{3+1}$, or (b) threshold processing on the stop flag $\hat{z}_3$. As a result, if the network functions as expected as in Fig. 1, the number of performed iterations corresponds to the number of sources. Note that, by counting background noise as one of the target sources to be extracted, the proposed network can naturally handle both 0 speaker scenarios and 1 or more speaker scenarios.

2.2. Training step of proposed method

This subsection explains the training procedure for the proposed network. We first introduce an overall cost function, and then describe its elements.

2.2.1. *Overall cost function for network training*

To train the network through back-propagation, we can use the following multi-task cost function:

$$J = J^{(\text{mse})} + \alpha J^{(\text{flag})} + \beta J^{(\text{res-mask})}, \tag{1}$$

where $J^{(\text{mse})}$ is the main cost function of the proposed network, which controls the mask estimation accuracy. $J^{(\text{flag})}$ is a cost function related to the stop flag. $J^{(\text{res-mask})}$ is a cost function for the network to ensure that all the T-F bins are covered by the estimated masks, and all the sources in the input spectrogram are extracted.

2.2.2. *Definition of $J^{(mse)}$*

Although the network is required to output a mask for a certain source in the input spectrogram at each iteration, we are not able to know in advance in which order the network will extract the sources. Since this label permutation problem is essentially similar to that addressed by PIT, we will adopt the same or a similar training procedure to that of PIT.

We here assume that the network processes the input utterance with a Bidirectional Long-Short Term Memory (BLSTM) RNN. Therefore, the following utterance-level mean square error (MSE) criterion proposed in [9] is a natural choice.

$$J^{(\text{mse})} = \frac{1}{B} \sum_{i=1}^{S} \|\hat{\mathbf{M}}_i \circ \mathbf{Y} - \mathbf{A}_{\phi^*}\|_F^2, \tag{2}$$

where $\hat{\mathbf{M}}_i$ is a mask estimated at the i-th iteration, and $\mathbf{A}_{\phi^*}$ is the amplitude spectrum of an appropriate target source. $\| \cdot \|_F$ is the Frobenius norm. $B = T \times N \times S$ is the total number of T-F bins over all target sources. T, N and S correspond to the number of time frames, the number of frequency bins and the total number of target sources, respectively. ϕ^* is the permutation that minimizes the

```
Algorithm 1: An algorithm for calculating $J^{(\text{mse})}$
Define a set $\mathcal{S} = \{1, 2, \ldots, S\}$, which stores a set of source
indices, and $\phi^* = [\,]$.
for i = 1 : S do
    1. Calculate $\hat{\mathbf{M}}_i$ with NN based on $\mathbf{Y}$ and $\mathbf{R}_i$.
    2. Obtain an index of a source in $\mathcal{S}$, $s_i^*$, which produces
       the minimum MSE with the estimated mask, i.e.
       $s_i^* = \mathrm{argmin}_{s_i \in \mathcal{S}} \|\hat{\mathbf{M}}_i \circ \mathbf{Y} - \mathbf{A}_{s_i}\|_F^2$
    3. Remove $s_i^*$ from the set $\mathcal{S}$ and form a new set $\mathcal{S}$ with
       the remaining indices.
    4. $\phi^*[i] \leftarrow s_i^*$ .
    5. Calculate the residual mask for the next iteration as
       (a) $\mathbf{R}_{i+1} = \max(\mathbf{R}_i - \mathbf{M}_{s_i^*}, 0)$ (w/ oracle mask), or
       (b) $\mathbf{R}_{i+1} = \max(\mathbf{R}_i - \hat{\mathbf{M}}_i, 0)$ (w/ estimated mask).
end
```

following utterance-level separation error:

$$\phi^* = \underset{\phi \in \mathcal{P}}{\mathrm{argmin}} \sum_{i=1}^{S} \|\hat{\mathbf{M}}_i \circ \mathbf{Y} - \mathbf{A}_\phi\|_F^2. \tag{3}$$

Here, $\mathcal{P}$ is the set of all permutations. It should be noted that in our case eq. (3) can be evaluated only after performing all the iterations, which may not be convenient as we will show below.

Alternatively, we can determine the permutation ϕ^* at each iteration recursively as in Alg. 1, by taking the recurrent structure of the proposed method into account. In Alg. 1, $\mathbf{M}_{s_i^*}$ is an ideal mask that is estimated at the i-th iteration. The advantage of this way of calculating permutation is that, if we use 5-(a) of Alg. 1, it is guaranteed that the mask estimation process is always performed based on an ideal input signal, i.e. an ideal residual mask at each iteration, and hence the gradient we obtain eventually becomes more reliable. By having such ideal input signals, the training converges very quickly to a better local minimum. Moreover, by switching the residual mask calculation from 5-(a) to 5-(b) (where we use an estimated mask to calculate a residual mask) after a certain epoch, we can make the network robust to errors in masks estimated in preceding iterations. In our experiments, we will use the procedure summarized in Alg. 1.

2.2.3. *Definition of $J^{(\mathit{flag})}$*

$J^{(\text{flag})}$ is the cost regarding the stop flag, which can tell us whether or not the iteration should be stopped. Let $\mathbf{z}$ be a vector of the stop flags formed of $S-1$ zeros followed by one, i.e. $\mathbf{z} = (z_1, z_2, \ldots, z_{S-1}, z_S) = (0, 0, \ldots, 0, 1)$. Then, $J^{(\text{flag})}$ can be formulated as a cross-entropy loss as follows:

$$J^{(\text{flag})} = -\sum_{i=1}^{S} z_i \ln \hat{z}_i, \tag{4}$$

where z_i and $\hat{z}_i$, respectively, correspond to the true and estimated stop flags at the i-th iteration.

2.2.4. *Definition of $J^{(\mathit{res\text{-}mask})}$*

Finally let us define $J^{(\text{res-mask})}$, which encourages the network to cover all the T-F bins of the input spectrogram. To impose such a constraint, we propose that $J^{(\text{res-mask})}$ be:

$$J^{(\text{res-mask})} = \max\left(\mathbf{1} - \sum_{i=1}^{S} \hat{\mathbf{M}}_i, 0\right). \tag{5}$$

In our implementation, we empirically applied the max function as above to handle negative values in the residual mask.

3. EXPERIMENTS

We carried out two experiments to investigate the effectiveness and characteristics of the proposed method.

In the first experiment, based on a standard source separation task, we evaluated the source separation performance of the proposed method compared with that of a conventional PIT with an utterance-level cost function, i.e. uPIT [9]. As an evaluation metric, we used the signal-to-distortion ratio (SDR) of BSSeval [13]. Hereafter, this first experiment will be referred to as clean matched condition.

We used the second experiment mainly to evaluate the source counting accuracy. Since the test data generated for this experiment comprises 0 speaker, 1 speaker and 2 simultaneous speaker scenarios, the network is required to change the number of iterations/masks adaptively while performing the source separation. Since we added a moderate level of noise to the training and test data, we refer to this experiment as the noisy mixed condition.

3.1. Experiment 1: clean matched condition

3.1.1. *Experimental conditions*

We evaluate the proposed method in comparison with an uPIT model [9] on the WSJ0-2mix dataset. The WSJ0-2mix dataset was introduced in [7] and was derived from the WSJ0 corpus [14]. The 30h training set and the 10h validation set contain two-speaker mixtures generated by randomly selecting speakers and utterances from the WSJ0 training set si_tr_s, and mixing them at various signal-to-noise ratios (SNRs) uniformly chosen between 0 dB and 5 dB. The 5h open-speaker test set was similarly generated using utterances from 16 speakers from the WSJ0 validation set si_dt_05 and evaluation set si_et_05. The sampling frequency was 8 kHz.

3.1.2. *Network details*

One of the inputs to the proposed models $\mathbf{Y}$ is a spectrogram consisting of the 257-dimensional short-time Fourier transform (STFT) spectral magnitude of the speech mixture. The size and shift of the STFT were 512 and 128, respectively. The other input is a residual mask $\mathbf{R}_i$ with the same dimension as $\mathbf{Y}$. These two inputs are first concatenated and fed to the network. Then, based on the input, the network outputs an amplitude mask estimated with the sigmoid activation function [9, 15], assuming that the amplitude mask falls in the [0,1] range. We employed a 2-layer BLSTM network with 600 LSTM units (i.e., 600 cells each for backward and forward RNNs) at each layer, followed by 1 fully connected layer. To control the training process, we used an Adam optimizer with an initial learning rate of 0.001. The maximum epoch was set at 200. With preliminary experiments, we confirmed that the performance can be improved by using e.g. larger models and more epochs, but we opted for this configuration to speed up the experiments.

Two types of models were trained, which differ as regards the cost function. One model, which we refer to as "Res-mask-model", uses $J = J^{(\text{mse})} + \beta J^{(\text{res-mask})}$, while the other model, which we refer to as "Stop-flag-model", employed $J = J^{(\text{mse})} + \alpha J^{(\text{flag})}$, which is the subset of eq. (1). We used an α value of 0.05 and a β value of 1e$-$5. We prepared three different training recipes for each model. The first recipe (train1) corresponds to the recipe 5-(b) (i.e. the estimated mask for the calculation of residual mask) of Alg. 1 all through the training process. The second (train2) uses a recipe corresponding to 5-(a) (i.e. oracle mask for the calculation of a residual mask) of Alg. 1. The third recipe (train2') carries out the

Table 1. SDR improvement (in dB) for different separation methods in clean matched condition

uPIT [9]	Res-mask-model			Stop-flag-model		
	train1	train2	train2’	train1	train2	train2’
7.2	7.8	7.9	8.6	7.3	8.8	8.1

Table 2. Source counting accuracy (in %) of each model in mixed noisy conditions

Exp. condition	0 speaker	1 speaker	2 speaker
Res-mask-model	100.0	100.0	99.9
Stop-flag-model	100.0	99.9	96.5

first 40 epochs with 5-(a), and then switches to 5-(b) for the remaining epochs.

3.1.3. Results

As in table 1, while uPIT achieves an SDR improvement of 7.2 dB [1], our models with a similar configuration, namely the Res-mask-model and Stop-flag-model achieved higher SDR improvements of 8.6 dB and 8.8 dB, respectively. We expect to realize further gains by improving the configuration (i.e. type of masks, activation functions) as in [9], which will be a part of our future work. In this experiment, since the NNs were trained only on 2 speaker mixtures, the Stop-flag-model always output the stop flag of 1 at the second iteration (and 0 at the first iteration), and the Res-mask-model output a very empty residual mask $\mathbf{R}_3$. Therefore, the decision to stop the iteration could be made with 100 % accuracy by using simple threshold processing.

3.2. Experiment 2: noisy mixed condition

3.2.1. Experimental conditions

In this experiment, we mainly evaluated the source counting performance of the proposed methods. The evaluation was based on a database comprising 3 noisy conditions, namely a 2 speaker mixture in noisy conditions, 1 speaker in noisy conditions and 0 speakers in noisy conditions. We chose this setting, because, for example, AMI meeting recordings rarely contain 3 simultaneous speaker regions (only 0.3 % of an entire meeting) and are dominated by 0, 1 and 2 speaker regions. As in the previous experiment, all speech data used to generate training, validation and test data were taken from the WSJ0 database. Background noise data were taken from the CHiME real noise dataset, and added to the mixture with an SNR of 20 dB relative to the first speaker in the mixture. We generated a 30h training set, which covered the above 3 noisy conditions equally. In other words, the training data for a 2 speaker mixture in this experiment were only about 1/3 of those used in the previous experiment, which implies inferior separation performance in the 2 speaker mixture condition. However, we employed this database to speed up our experiments. As in the previous experiment, we generated a 10h validation set containing the same 3 types of noisy conditions, and a 5h test set for each of the noisy conditions.

3.2.2. Network details

We used the same network configuration, except that this time we forced the network to always output a mask for noise at the first iteration. In this experiment, we used the Res-mask-model with train2’, and the Stop-flag-model with train2’. For the source counting, i.e. to stop the iteration during the test, we used a threshold of 0.9 for the stop flag of the Stop-flag-model, and 0.1 for the median of the residual mask for the Res-mask-model.

3.2.3. Results

Table 2 shows the accuracy of source counting under each condition with each model. The Res-mask-model worked particularly well, achieving an accuracy exceeding 99% for all tested conditions. It should be noted that the values of the stop flag and the mean and median values of the residual mask were fortunately quite discriminative, and thus threshold adjustment was an easy task. Concerning the source separation accuracy, the SDR improvement we obtained e.g. for a noisy 2 mixture case, was not as good as in the previous experiment, i.e. it was about 6.6 dB. This is mainly because the training data for each condition amount to about 1/3 of the data used in the previous experiment. According to our preliminary investigation, we are sure that this performance can be greatly improved simply by increasing the number of training data, which is also a part of our future work.

4. RELATION TO PRIOR WORKS

Recently, PIT was extended to handle an unknown number of sources [16], in which the maximum number of sources to be extracted during the test is assumed to be known in advance. When the network is trained with fewer sources, they interestingly assume the existence of silent speaker(s), and output a mask for them. It was found that, by training PIT in this way, we can efficiently handle 2- and 3-speaker scenarios with the same network having a fixed output node dimension for 3 speakers. A major difference lies in the structure of the network; while their network has to fix the maximum number of speakers, our model can theoretically handle an arbitrary number of speakers. In this paper, we compared our source separation performance with theirs in Section 3.2. Furthermore, we showed the efficacy of our model in 0-, 1- and 2-speaker scenarios, which we believe to be very important in real meeting scenarios.

Another related study [17] originates in the image processing field, where the authors use a similar network to iteratively identify an object in an image. Interestingly their network was trained in an unsupervised manner and was shown to generalize well for unseen conditions. The difference lies in the fact that, in our case, (a) the network is trained in a strongly supervised manner, (b) the network structure was appropriately modified to handle sequence data like speech, and (c) the target objects to be extracted are not continuous like an image object, but discrete, meaning that speech is an intermittent object.

5. CONCLUSIONS

In this paper, we proposed a neural network-based mask estimator that can handle an arbitrary number of speakers and adaptively change the number of output masks depending on the input signal. Thanks to the nature of the employed recurrent neural networks (RNN) that can learn how many computational steps/iterations to perform, the proposed method attends to one speaker at a time, and estimates a mask for each speaker individually at each iteration until all the sources of interest are extracted from the input spectrogram. We presented the training procedure of this network and confirmed experimentally that it showed promising performance.

[1] The value corresponds to the result of [9] in Table II, which was obtained with a similar configuration to ours. It is a result obtained with a model having 3 BLSTM layers each with 896 units and by estimating amplitude masks using a sigmoid activation function.

6. REFERENCES

[1] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, “The Microsoft 2016 conversational speech recognition system,” in *ICASSP*, 2017, pp. 5255–5259.

[2] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L. Lim, B. Roomi, and P. Hall, “English conversational telephone speech recognition by humans and machines,” in *Interspech*, 2017, pp. 132–136.

[3] O. Yilmaz and S. Rickard, “Blind separation of speech mixture via time-frequency masking,” *IEEE Trans. Signal Processing*, vol. 52, no. 7, pp. 1830–1847, 2004.

[4] H. Sawada, S. Araki, and S. Makino, “Underdetermined convolutive blind source separation via frequency bin-wise clustering and permutation alignment,” *IEEE Trans. Audio, Speech, and Language Processing*, vol. 19(3), pp. 516–627, 2011.

[5] S. Makino, T. Lee, and H. Sawada, *Blind source separation*, Springer, New York, NY, 2007.

[6] C. Fevotte an N. Bertin and J. Durrieu, “Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis,” *Neural computation*, vol. 21(3), pp. 793 – 830, 2009.

[7] J. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, “Deep clustering: Discriminative embeddings for segmentation and separation,” in *ICASSP*, 2016, pp. 31–35.

[8] Z. Chen, Y. Luo, and N. Mesgarani, “Deep attractor network for single-microphone speaker separation,” 2016, arXiv:1611.08930.

[9] M. Kolbæk, D. Yu, Z. Tan, and J. Jensen, “Multi-talker speech separation with utterance-level permutation invariant training of deep recurrent neural networks,” 2017, arXiv:1703.06284.

[10] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, “Permutation invariant training of deep models for speaker-independent multi-talker speech separation,” in *ICASSP*, 2017, pp. 241 – 245.

[11] T. Higuchi, K. Kinoshita, and T. Nakatani, “Deep clustering-based beamforming for separation with unknown number of sources,” in *Interspech*, 2017, pp. 1183–11876.

[12] Alex Graves, “Adaptive computation time for recurrent neural networks,” 2016, arXiv:1603.08983.

[13] E. Vincent, R. Gribonval, and C. Fevotte, “Performance measurement in blind audio source separation,” *IEEE Trans. Audio, Speech and Lang. Process.*, vol. 14, pp. 1462–1469, 2006.

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

[15] H. Erdogan, J. R. Hershey, S. Watanabe, and J. L. Roux, “Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks,” in *ICASSP*, 2015, pp. 708–712.

[16] M. Kolbæk, D. Yu, Z. Tan, and J. Jensen, “Joint separation and denoising of noisy multi-talker speech using recurrent neural networks and permutation invariant training,” in *International Workshop on Machine Learning for Signal Processing (MLSP)*, 2017, pp. 31–35.

[17] S. M. A. Eslami, N. Heess, T. Weber, Y. Tassa, D. Szepesvari, K. Kavukcuoglu, and G. E. Hinton, “Attend, infer, repeat: Fast scene understanding with generative models,” 2016, arXiv:1603.08575.