SINGLE CHANNEL SPEECH SEPARATION WITH CONSTRAINED UTTERANCE LEVEL PERMUTATION INVARIANT TRAINING USING GRID LSTM

Chenglin Xu[1,2]*, Wei Rao*[2]*, Xiong Xiao*[3]*, Eng Siong Chng*[1,2]*, Haizhou Li*[2,4]

[1] School of Computer Science and Engineering, Nanyang Technological University, Singapore
[2] Temasek Laboratories@NTU, Nanyang Technological University, Singapore
[3] Microsoft Corporation , United States
[4] Department of Electrical and Computer Engineering, National University of Singapore, Singapore
{xuchenglin,raowei,aseschng}@ntu.edu.sg, xioxiao@microsoft.com, haizhou.li@nus.edu.sg

ABSTRACT

Utterance level permutation invariant training (uPIT) technique is a state-of-the-art deep learning architecture for speaker independent multi-talker separation. uPIT solves the label ambiguity problem by minimizing the mean square error (MSE) over all permutations between outputs and targets. However, uPIT may be sub-optimal at segmental level because the optimization is not calculated over the individual frames. In this paper, we propose a constrained uPIT (cuPIT) to solve this problem by computing a weighted MSE loss using dynamic information (i.e., delta and acceleration). The weighted loss ensures the temporal continuity of output frames with the same speaker. Inspired by the heuristics (i.e., vocal tract continuity) in computational auditory scene analysis, we then extend the model by adding a Grid LSTM layer, that we name it as cuPIT-Grid LSTM, to automatically learn both temporal and spectral patterns over the input magnitude spectrum simultaneously. The experimental results show 9.6% and 8.5% relative improvements on WSJ0-2mix dataset under both closed and open conditions comparing with the uPIT baseline.

Index Terms— Constrained Permutation Invariant Training, Grid LSTM, Single Channel Speech Separation.

1. INTRODUCTION

Human listeners with normal hearing have the ability to focus their auditory attention on a specific signal in a complex acoustic environment, i.e., a conversation with background noise and competing speech. However, such a problem, known as the cocktail party problem [1], is not trivial to solve by a machine. However, a machine solution is required in a large number of applications, such as automatic meeting transcription and hearing aids.

Inspired by the findings on how human listeners separate sources in an acoustic mixture, initial works of computational auditory scene analysis (CASA) [2, 3] methods were proposed based on heuristics, e.g., pitch continuity. Another non-negative matrix factorization (NMF) [4, 5, 6] techniques reconstruct each source using non-negative dictionaries and activations. However, these methods not only rely on accurate trackers (i.e., pitch tracker) but also intensive computation.

In the past years, several deep learning based techniques were proposed for single channel multi-talker speech separation. A speaker dependent system in [7] achieves good performance in closed-set speaker-dependent condition. However, speaker independent separation of two or more speakers remains an open and challenging task. The deep clustering (DC) [8] technique was proposed to solve the problem that showed competitive results. With the assumption that each time-frequency (TF) bin is dominated by a single speaker, the DC method uses bidirectional long-short term memory (BLSTM) to project the spectrogram of the mixture to an embedding space, in which TF bins of different speakers are clustered by using K-means to obtain speaker-dependent mask. The masks are then applied on the mixture to obtain the speech of individual speakers. To overcome the drawback of binary mask, the soft masks are estimated by an enhancement network stacked on the top of the DC system to further improve the performance in [9]. One shortcoming of DC is that its objective function is defined in the embedding space, which may not be optimal for the speech separation task. Then deep attractor network (DANet) [10] method was proposed to solve this limitation by separating signals directly. It creates attractor points in high dimensional embedding space to group the TF bins belonging to each source together. Unfortunately, this method adds the complexity of forming the attractor to the run-time process.

Recently, permutation invariant training (PIT) [11] and utterance level PIT (uPIT) [12] were introduced to solve the speech separation problem with end-to-end training. The PIT method solves the label ambiguity problem by minimizing the mean square error (MSE) over all permutation at frame level in the training stage, but it suffers from frame-level permutation problem during inference. The uPIT method was pro-

Xiong Xiao contributed to this work before joining Microsoft.

posed to solve the problem by forcing the separated frames belonging to the same speaker to be aligned to the same output stream using BLSTM with an utterance level training criterion. However, such utterance level training criterion may not be optimal becuase some frames belonging to speaker A may be aligned to the output stream of speaker B.

In this paper, we propose a constrained uPIT (cuPIT) to solve the aforementioned problem by using a cost function that penalizes unnatural temporal structure of the separated speech spectrum. In addition to the MSE cost function of uPIT, we also introduce the MSE between the separated spectrogram and the target spectrogram in the dynamic features domain. The dynamic features, e.g. the delta and acceleration features, were originally proposed to capture speech context information for the speech recognition task [13]. By minimizing the MSE in the dynamic feature domain, we ensure that the separated spectrogram has similar temporal structure as the target spectrogram. To capture the heuristic patterns in frequency domain, e.g., common onset/offset, this paper also proposes to add a grid LSTM [14, 15, 16] in the cuPIT model to learn corresponding temporal and spectral patterns from the magnitude spectrum both in time and frequency axis simultaneously.

Section 2 describes the monaural speech separation problem. The details of the proposed model are discussed in Section 3. Section 4 introduces the experimental setup and the results. Finally, conclusions are drawn in Section 5.

2. MONAURAL SPEECH SEPARATION BY TF MASKING

The task of monaural speech separation aims to separate a linearly mixed single channel microphone signal $y(n)$ into individual source signals $x_s(n), s \in [1, S]$.

$$y[n] = \sum_{s=1}^{S} x_s[n] \tag{1}$$

The goal is to estimate $\hat{x}_s[n]$ that is close to $x_s[n]$. However, the optimization is difficult in time domain. With the assumption that speech is sparse in the spectrogram representation, the mixture is transformed to frequency domain as $Y(t, f)$ for each TF bin (t, f). According to the commonly used masking approach, the magnitude $|\hat{X}_s(t, f)|$ of individual source is estimated by

$$|\hat{X}_s(t, f)| = M_s(t, f) \odot |Y(t, f)| \tag{2}$$

where $\odot$ indicates element-wise multiply. Then the estimated magnitude $|\hat{X}_s(t, f)|$ and the phase of mixed speech $\angle Y(t, f)$ are used to reconstruct the time domain waveform $\hat{x}_s[n]$ by an inverse discrete Fourier transform and an overlap and add operation. Since the phase estimation is still an open problem in speech separation or speech enhancement, the phase of mixed speech is directly used.

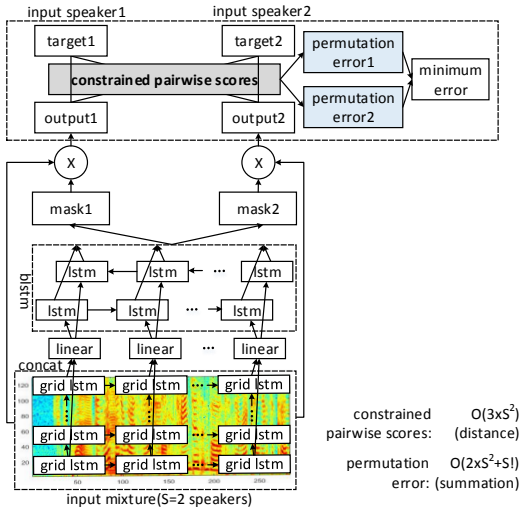

Fig. 1: The proposed cuPIT-Grid LSTM system architecture for training. During run-time testing, the upper dotted box is not necessary. The systems takes input mixture and outputs output1 and output2.

3. CUPIT-GRID LSTM SYSTEM

In speech separation task, the state-of-the-art methods (i.e., DC [8, 9] [1], DANet [10], PIT [11], uPIT [12]) commonly estimate a mask for each individual source. In this paper, we propose to estimate magnitude spectrum approximation masks with a constrained uPIT in the cost function and a grid LSTM network to explore the temporal-spectral patterns, as shown in Figure 1.

3.1. Grid LSTM

The grid LSTM[14, 15, 16] features separate LSTMs that move in both time and frequency axis. The grid frequency LSTM (gF-LSTM) and grid time LSTM (gT-LSTM) communicate through activations of previous time step and previous frequency step. When the peephole operation is applied, the cell memories of previous time and frequency steps are used to computing the weights for controlling the gates. In the frequency axis, we use a sliding window of F (i.e., 29) with a stride of A (i.e., 10) on the frequency features ($N = 129$). The gF-LSTM is thus unrolled over frequency by an amount of $B = (N - F)/A + 1$. For each time frequency step $(t, k), t \in [1, T], k \in [1, B]$, the grid LSTM in dimension $j, j \in \{t, k\}$ is defined as,

$$H_{u,t,k} = W_{uh}^{(t)} h_{t-1,k}^{(t)} + W_{uh}^{(k)} h_{t,k-1}^{(k)}, \quad u \in \{i, f, c, o\} \tag{3}$$

$$C_{u,t,k} = W_{uc}^{(t)} c_{t-1,k}^{(t)} + W_{uc}^{(k)} c_{t,k-1}^{(k)}, \quad u \in \{i, f\} \tag{4}$$

$$i_{t,k}^{(j)} = \sigma(W_{ix}^{(j)} y_{t,k} + H_{i,t,k} + C_{i,t,k} + b_i^{(j)}) \tag{5}$$

$$f_{t,k}^{(j)} = \sigma(W_{fx}^{(j)} y_{t,k} + H_{f,t,k} + C_{f,t,k} + b_f^{(j)}) \tag{6}$$

[1] The masks are obtained by clustering the embeddings of each TF bin using K-means.

$$c_{t,k}^{(t)} = f_{t,k}^{(t)} \odot c_{t-1,k}^{(t)} + i_{t,k}^{(t)} \odot g(W_{cx}^{(t)} y_{t,k} + H_{c,t,k} + b_c^{(t)}) \quad (7)$$

$$c_{t,k}^{(k)} = f_{t,k}^{(k)} \odot c_{t-1,k}^{(k)} + i_{t,k}^{(k)} \odot g(W_{cx}^{(k)} y_{t,k} + H_{c,t,k} + b_c^{(k)}) \quad (8)$$

$$o_{t,k}^{(j)} = \sigma(W_{ox}^{(j)} y_{t,k} + H_{o,t,k} + W_{oc}^{(t)} c_{t,k}^{(t)} + W_{oc}^{(k)} c_{t,k}^{(k)} + b_o^{(j)}) \quad (9)$$

$$h_{t,k}^{(j)} = o_{t,k}^{(j)} \odot \sigma(c_{t,k}^{(j)}) \quad (10)$$

In this paper, the weight matrices are shared among the time and frequency cells to reduce the computation. Finally, at each time step t, the outputs of the gF-LSTM $\{h_{t,1}^{(k)}, ..., h_{t,B}^{(k)}\}$ and gT-LSTM $\{h_{t,1}^{(t)}, ..., h_{t,B}^{(t)}\}$ are concatenated and given to the linear dimensionality reduction layer, followed by BLSTM.

3.2. Constrained uPIT

In the mask estimation layer, we use the ideal phase sensitive mask (IPSM) [17] that considers the phase differences between the mixture and individual sources. For each TF bin (t,f), the IPSM is estimated as

$$M_s(t,f) = \frac{|X_s(t,f)| cos(\theta_y(t,f) - \theta_s(t,f))}{|Y(t,f)|} \quad (11)$$

Previous work has shown that the masks trained with the magnitude spectrum approximation loss outperform those with the least square loss between the estimated and ideal masks [18]. When the magnitude spectrum approximation loss is applied, the training criterion will be the MSE between the estimated magnitude and true magnitude with phase difference.

$$J = \frac{1}{T} \sum_{s=1}^{S} ||\hat{M}_s \odot |Y| - |X_s| \odot cos(\theta_y - \theta_s)||_F^2 \quad (12)$$

where $||\cdot||_F$ is the Frobenius norm.

In this paper, we propose a constrained cost function by adding weighted MSE of dynamic features between estimated magnitude and target magnitude with phase difference.

$$\begin{aligned} J_{c,\phi_p(s)} = \frac{1}{T} \sum_{s=1}^{S} (&||\hat{M}_s \odot |Y| - |X_{\phi_p(s)}| \odot cos(\theta_y - \theta_{\phi_p(s)})||_F^2 \\ &+ w_D ||f_D(\hat{M}_s \odot |Y|) - f_D(|X_{\phi_p(s)}| \odot cos(\theta_y - \theta_{\phi_p(s)}))||_F^2 \\ &+ w_A ||f_A(\hat{M}_s \odot |Y|) - f_A(|X_{\phi_p(s)}| \odot cos(\theta_y - \theta_{\phi_p(s)}))||_F^2) \end{aligned} \quad (13)$$

where $\phi_p(s), p \in [1, P]$ is an assignment of target source (s) to an output, and $P = S!$ is the number of all permutations. w_D, w_A are the weights for delta and acceleration. The constrain function is defined as

$$f_D(v(t)) = \frac{\sum_{l=1}^{L} l \times (v(t+l) - v(t-l))}{\sum_{l=1}^{L} 2l^2} \quad (14)$$

where L is the order and sets as 2 in this study. $f_A(\cdot)$ represent the computations of $f_D(\cdot)$ twice.

Since the dynamic features are computed from a context window, we prevent speakers from switching in the same output streams. This is actually a novel application of dynamic features in the cost function, although it has been used for speech enhancement [19], where there is no speaker switching problem.

To solve the label ambiguity problem, the optimal assignment is done by choosing the minimal constrained cost among all permutations (P). For instance, the costs of $2! = 2$ permutations (error1, error2 when p=1 and 2) are considered in Figure 1.

$$\hat{p} = \arg\min_{p \in P} J_{c,\phi_p(s)} \quad (15)$$

And the final constrained cost used to optimize the network is obtained with the optimal assignment.

$$J = J_{c,\phi_{\hat{p}}(s)} \quad (16)$$

4. EXPERIMENTS AND DISCUSSION

4.1. Experimental Setup

We evaluated the methods on the WSJ0-2mix dataset [2] [8], which was mixed by randomly choosing utterances of two speakers from the WSJ0 corpus [20]. In this paper, the WSJ0-2mix (two-speaker mixed) dataset was divided into three sets: training set ($20,000$ utterances $\approx 30h$), development set ($5,000$ utterances $\approx 8h$), and test set ($3,000$ utterances $\approx 5h$). Specifically, the training and development set were generated by randomly selecting utterances from 50 male and 51 female speakers in the WSJ0 training set (si_tr_s) at various signal-to-noise (SNR) ratios uniformly chosen between 0dB and 5dB. Similar as generating training and development set, the test set was created by mixing the utterances from 10 male and 8 female speakers in the WSJ0 development set (si_dt_05) and evaluation set (si_et_05). Because the speakers in the development set were the same as those in the training set, we use the development set in closed condition (CC) to tune parameters. Moreover, as the speakers in the test set were different from those in the training and development sets, the test set was considered as open condition (OC) evaluation.

The sampling rate of all generated data is 8kHz. The input 129-dim spectral magnitude features of the mixed speech were computed by a STFT with the normalized square root of the 32ms length hamming window and 16ms window shift. The magnitudes of two targets were obtained in the same way.

In this work, 3 BLSTM layers with 896 units in each layer were deployed to keep the network configuration same as the baseline in [12]. The mask estimation layer used ReLU as the activation function. The units of the Grid LSTM cell were set to 64. Then the outputs from the Grid LSTM layer were given to a linear layer, which reduced the dimension of the outputs from 1408 to 896. A random dropout with a dropout rate of

[2] Available at: http://www.merl.com/demos/deep-clustering

0.5 was used in the Grid LSTM and BLSTM [3]. The weight w_D and w_A were tuned to be 4.5 and 10. The learning rate was initialized as 0.0005 and scaled down by 0.7 when the training loss increased on the development set. Each mini-batch had 16 randomly selected utterances. The number of minimum epoch was set to 30 and the early stopping criterion was that the relative loss improvement was lower than 0.01. The model was optimized with Adam algorithm [21] and implemented using Tensorflow [4].

In this work, performance was evaluated with global normalized signal-to-distortion ratio (GNSDR, same as "SDR improvement" in [8, 9, 11, 12]) using the toolbox in [22].

4.2. Experimental Results

4.2.1. Constrained uPIT vs. Conventional uPIT

Table 1 shows the GNSDR comparisons between our proposed approach and other state-of-the-art methods on the WSJ0-2mix database. The similar performances of the DANet, DC+ and uPIT-BLSTM method are observed. We consider uPIT-BLSTM as a benchmarking reference, therefore, we re-implement uPIT-BLSTM with our experiment setup and obtain a similar result to [12]. In our implementation, we add a dense layer between the input layer and the first BLSTM layer. The ideal ratio mask (IRM) and IPSM are also evaluated to show the upper bounds.

With the proposed constraints added in the cost function, our constrained uPIT-BLSTM (cuPIT-BLSTM) method achieves better performance than uPIT-BLSTM using the same network configuration. The constrained cost function not only ensures the estimated magnitude of the frames is close to its target, but also their delta and acceleration. Since the dynamic information is computed with a context window, it provides a constraint to ensure that the output frames of the same speaker do not jump to the other speaker.

4.2.2. Effect of Grid LSTM

Since the magnitude patterns (i.e., onset, harmonic and formant) of a speaker are corresponding in frequency domain, the handcrafted rules are created according to these patterns in CASA methods. Inspired by these rules, we proposed to use a Grid LSTM to learn such rules from those patterns in time and frequency automatically. Since the Grid LSTM can learn temporal and spectral patterns simultaneously, we replace the dense layer in cuPIT-BLSTM by a Grid LSTM layer and a linear dimension reduction layer, named as cuPIT-Grid LSTM. It achieves 10.2dB and 10.1dB GNSDR under closed and open conditions as shown in Table 1. Compared with cuPIT-BLSTM, a further 3% relative improvement is obtained under the open condition. It shows the effectiveness of the Grid LSTM in learning temporal and spectral patterns. By additional training epochs with a reduced dropout rate of 0.3, the performance is further improved (-RD model in Table 1). Finally, the proposed approach achieves an accumulated 9.6% and 8.5% relative improvement over the uPIT-BLSTM [12] under closed and open conditions, respectively.

[3]The dropout was not applied across time steps, although it was known to be effective and used in [9].

[4]https://www.tensorflow.org/

Table 1: GNSDR (dB) in a comparative study of different separation methods on the WSJ0-2mix dataset with optimal frame level assignment or default assignment on closed (CC) and open (OC) conditions. * indicates our re-implementation of the work in [12].

Method	Opt Assign		Def Assign	
	CC	OC	CC	OC
DC [8]	-	-	5.9	5.8
DC+ [9]	-	-	-	9.4
DANet [10]	-	-	-	9.6
PIT-DNN [11]	7.3	7.2	5.7	5.2
PIT-CNN [11]	8.4	8.6	7.7	7.8
uPIT-BLSTM [12]	10.9	10.8	9.4	9.4
uPIT-BLSTM*	10.8	10.7	9.6	9.5
cuPIT-BLSTM	11.1	11.0	10.0	9.8
cuPIT-Grid LSTM	11.2	11.2	10.2	10.1
cuPIT-Grid LSTM-RD	**11.3**	**11.3**	**10.3**	**10.2**
IRM	12.4	12.7	12.4	12.7
IPSM	14.9	15.1	14.9	15.1

4.2.3. Different vs. Same Gender

The performances of the mixtures with speakers under different and same gender conditions are reported in Table 2. The proposed techniques improve the performance consistently in both different and same gender conditions and achieve 12.0dB and 8.2dB GNSDR under open condition. Since the characteristics of same gender speakers are closer than those of different gender (i.e., pitch), the GNSDR of same gender is lower than that of different gender. We note that same gender mixed speech separation remains a difficult task. However, we are encouraged by the fact that our proposed system achieves a 12.3% and a 4.3% relative improvement for same and different gender over the uPIT-BLSTM baseline under open condition.

Table 2: GNSDR (dB) in a comparative study of same and different gender combinations on WSJ0-2mix development and test sets (Def Assign.).

Method	CC		OC	
	Same	Diff	Same	Diff
uPIT-BLSTM*	7.8	11.5	7.3	11.5
cuPIT-BLSTM	8.3	11.8	7.7	11.7
cuPIT-Grid LSTM	8.5	11.9	8.1	11.8
cuPIT-Grid LSTM-RD	**8.6**	**12.0**	**8.2**	**12.0**
IRM	12.2	12.7	12.4	12.9
IPSM	14.6	15.1	14.9	15.3

5. CONCLUSION

In this paper, we proposed a constrained cost function in uPIT with Grid LSTM to separate speakers in a single channel mixed speech signal. Experimental results show that our proposed method not only achieves the better performance than the conventional uPIT model, but also outperforms the current state-of-the-art DANet and DC. Moreover, we also find the effectiveness of our proposed method on the same gender mixed speech separation task.

6. REFERENCES

[1] Adelbert W Bronkhorst, "The cocktail party phenomenon: A review of research on speech intelligibility in multiple-talker conditions," *Acta Acustica united with Acustica*, vol. 86, no. 1, pp. 117–128, 2000.

[2] Daniel Patrick Whittlesey Ellis, *Prediction-driven computational auditory scene analysis*, Ph.D. thesis, Massachusetts Institute of Technology, 1996.

[3] DeLiang Wang and Guy J Brown, *Computational auditory scene analysis: Principles, algorithms, and applications*, Wiley-IEEE press, 2006.

[4] Patrik O Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, no. Nov, pp. 1457–1469, 2004.

[5] Mikkel N Schmidt and Rasmus Kongsgaard Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," in *Proceedings of INTERSPEECH*, 2006.

[6] Paris Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 1–12, 2007.

[7] Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 12, pp. 2136–2147, 2015.

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

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

[10] Zhuo Chen, Yi Luo, and Nima Mesgarani, "Deep attractor network for single-microphone speaker separation," in *Proceedings of ICASSP*. IEEE, 2017, pp. 246–250.

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

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

[13] Sadaoki Furui, "Speaker-independent isolated word recognition using dynamic features of speech spectrum," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 1, pp. 52–59, 1986.

[14] Nal Kalchbrenner, Ivo Danihelka, and Alex Graves, "Grid long short-term memory," *arXiv preprint arXiv:1507.01526*, 2015.

[15] Tara N Sainath and Bo Li, "Modeling time-frequency patterns with lstm vs. convolutional architectures for lvcsr tasks.," in *Proceedings of INTERSPEECH*, 2016, pp. 813–817.

[16] Shuo-Yiin Chang, Bo Li, Tara N Sainath, Gabor Simko, and Carolina Parada, "Endpoint detection using grid long short-term memory networks for streaming speech recognition," *Proceedings of Interspeech 2017*, pp. 3812–3816, 2017.

[17] Hakan Erdogan, John R Hershey, Shinji Watanabe, and Jonathan Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *Proceedings of ICASSP*. IEEE, 2015, pp. 708–712.

[18] Yuxuan Wang, Arun Narayanan, and DeLiang Wang, "On training targets for supervised speech separation," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 12, pp. 1849–1858, 2014.

[19] Xiong Xiao, Shengkui Zhao, Duc Hoang Ha Nguyen, Xionghu Zhong, Douglas L Jones, Eng Siong Chng, and Haizhou Li, "Speech dereverberation for enhancement and recognition using dynamic features constrained deep neural networks and feature adaptation," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, pp. 4, 2016.

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

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

[22] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.