Utterance-level Permutation Invariant Training with Latency-controlled BLSTM for Single-channel Multi-talker Speech Separation

Lu Huang* and Gaofeng Cheng†‡ and Pengyuan Zhang†‡ and Yi Yang* and Shumin Xu§ and Jiasong Sun*
* Department of Electronic Engineering, Tsinghua University, Beijing, China
E-mail: hl17@mails.tsinghua.edu.cn, {yangyy, sunjiasong}@tsinghua.edu.cn
† University of Chinese Academy of Sciences, Beijing, China
‡ Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, Chinese Academy of Sciences, China
E-mail: {chenggaofeng, zhangpengyuan}@hccl.ioa.ac.cn
§ North China Power Engineering CO., LTD. of China Power Engineering Consulting Group
Email: xushumin81@tom.com

***Abstract*—Utterance-level permutation invariant training (uPIT) has achieved promising progress on single-channel multi-talker speech separation task. Long short-term memory (LSTM) and bidirectional LSTM (BLSTM) are widely used as the separation networks of uPIT, i.e. uPIT-LSTM and uPIT-BLSTM. uPIT-LSTM has lower latency but worse performance, while uPIT-BLSTM has better performance but higher latency. In this paper, we propose using latency-controlled BLSTM (LC-BLSTM) during inference to fulfill low-latency and good-performance speech separation. To find a better training strategy for BLSTM-based separation network, chunk-level PIT (cPIT) and uPIT are compared. The experimental results show that uPIT outperforms cPIT when LC-BLSTM is used during inference. It is also found that the inter-chunk speaker tracing (ST) can further improve the separation performance of uPIT-LC-BLSTM. Evaluated on the WSJ0 two-talker mixed-speech separation task, the absolute gap of signal-to-distortion ratio (SDR) between uPIT-BLSTM and uPIT-LC-BLSTM is reduced to within 0.7 dB.**

Index Terms: multi-talker speech separation, permutation invariant training, latency-controlled BLSTM, speaker tracing

I. Introduction

Many advancements have been observed for monaural multi-talker speech separation [1], [2], [3], [4], [5], [6], [7], [8], [9], known as cocktail party problem [10], which is meaningful to many practical applications, such as human-machine interaction, automatic meeting transcription etc. With the development of deep learning[11], a lot of innovations have been proposed, such as deep clustering [3], [4], deep attractor network [5], time-domain audio separation network [6], [9] and permutation invariant training (PIT) [7], [8].

Deep clustering [3], [4] projects the time-frequency (TF) units into an embedding space, with a clustering algorithm to generate a partition of TF units, which assumes that each bin belongs to only one speaker. However, the separation under the embedding space may be not the optimal technique.

Deep attractor network [5] also learns a high-dimensional representation of the mixed speech with some attractor points in the embedding space to attract all the TF units corresponding to the target speaker. However, the estimation of attractor points has a high computational cost.

PIT [7] is an end-to-end speech separation method, which gives an elegant solution to the training label permutation problem [5], [7]. It is extended to utterance-level PIT (uPIT) [8] with an utterance-level cost function to further improve the performance. Because uPIT is simple and well-performed, it draws a lot of attention [6], [9], [12], [13], [14], [15], [16], [17], [18], [19]. LSTM [20], [21], [22] and BLSTM [23], [24] are widely used for uPIT to exploit utterance-level long time dependency. Although uPIT-BLSTM outperforms uPIT-LSTM, its inference latency is as long as the utterance, which hampers its applications in many scenarios.

To reduce the latency of BLSTM-based acoustic model on automatic speech recognition (ASR) tasks, context-sensitive chunk (CSC) [25], which is the chunk with appended contextual frames, is proposed for both training and decoding. In [26], CSC-BLSTM is extended to latency-controlled BLSTM (LC-BLSTM), which directly carries over the left contextual information from previous chunk of the same utterance to reduce the computational cost and improve the recognition accuracy.

In this paper, inspired by LC-BLSTM-based acoustic model on ASR tasks, uPIT-LC-BLSTM for low-latency speech separation is proposed, which splits an utterance into non-overlapping chunks with future contextual frames during inference to reduce the latency from utterance-level to chunk-level. The chunk-level PIT (cPIT) of BLSTM is also proposed, but the preliminary experiments indicate that cPIT is inferior to uPIT. uPIT-LC-BLSTM propagates BLSTM's forward hidden states across chunks, which helps keep the speaker consistency across chunks. Meanwhile, an inter-chunk speaker tracing (ST) algorithm is proposed to further improve the performance of uPIT-LC-BLSTM. Experiments evaluated on the WSJ0 two-talker mixed-speech separation task show that uPIT-LC-

BLSTM with ST only loses a little when compared to uPIT-BLSTM.

The paper starts by briefly describing prior work in Section II. The cPIT, uPIT-LC-BLSTM and speaker tracing algorithm are described in Section III. The experimental setup and results are discussed in Section IV. Section V presents the conclusions.

II. Prior Work

A. Monaural Speech Separation

The goal of single-channel multi-talker speech separation is to separate the individual source signals from the mixed audio. Let us denote S source signals as $\mathbf{x}_s(t), s = 1, ..., S$ and the microphone receives mixed audio $\mathbf{y}(t) = \sum_{s=1}^{S} \mathbf{x}_s(t)$. The separation is often carried out in the time-frequency (TF) domain, where the task is to reconstruct the short-time Fourier transform (STFT) of each individual source signal. The STFT of the mixed signal is $\mathbf{Y}(t,f) = \sum_{s=1}^{S} \mathbf{X}_s(t,f)$, where $\mathbf{Y}(t,f)$ is the TF unit at frame t and frequency f.

The STFT reconstruction of each source can be done by estimating S masks $\hat{\mathbf{M}}_s(t,f), s = 1, ..., S$. We use phase sensitive mask (PSM) here: $\mathbf{M}_s(t,f) = \frac{|\mathbf{X}_s(t,f)|}{|\mathbf{Y}(t,f)|} \cos(\theta_{\mathbf{Y}}(t,f) - \theta_{\mathbf{X}_s}(t,f))$, where $|\mathbf{Y}|$ and $\theta_{\mathbf{Y}}$ are the magnitude and phase of $\mathbf{Y}$ respectively. With an estimated mask $\hat{\mathbf{M}}_s(t,f)$ and the mixed STFT, the STFT of source s is $\hat{\mathbf{X}}_s(t,f) = \hat{\mathbf{M}}_s(t,f) \cdot |\mathbf{Y}(t,f)| \cdot e^{j\theta_{\mathbf{Y}}(t,f)}$, where j is imaginary unit.

The straightforward mask-based separation methods based on deep learning are to use neural network to estimate masks for S source signals and then minimize the mean square error (MSE) between estimated and target magnitudes. For PSM, the cost function is as follows:

$$\mathcal{J}_{psm} = \frac{1}{B} \sum_{s=1}^{S} ||\hat{\mathbf{M}}_s \circ |\mathbf{Y}| - |\mathbf{X}|_s \circ \cos(\theta_{\mathbf{Y}} - \theta_{\mathbf{X}_s})||_F^2 \quad (1)$$

where $B = T \times F \times S$ is the total number of TF units, $\circ$ is the element-wise product and $||\cdot||_F$ is the Frobenius norm.

B. Utterance-level Permutation Invariant Training

The cost function mentioned above is a good way for some simple cases. For example, when a priori convention can be learned, we can force the speakers with higher energy (or male speakers) to be the first output, and those with lower energy (or female speakers) to be the second output. However, when the energy difference is small or two speakers have the same gender, a problem named label permutation [5], [7] is introduced, where the permutation of two output streams is unknown.

PIT [7] has eliminated the label permutation problem, while it faces another problem named speaker tracing, which is solved by extending PIT with an utterance-level cost function, i.e. uPIT [8], to force the separation of the same speaker into the same output stream. The cost function of uPIT is as follows:

$$\mathcal{J} = \frac{1}{B} \sum_{s=1}^{S} ||\hat{\mathbf{M}}_s \circ |\mathbf{Y}| - |\mathbf{X}|_{\phi^*(s)} \circ \cos(\theta_{\mathbf{Y}} - \theta_{\mathbf{X}_{\phi^*(s)}})||_F^2 \quad (2)$$

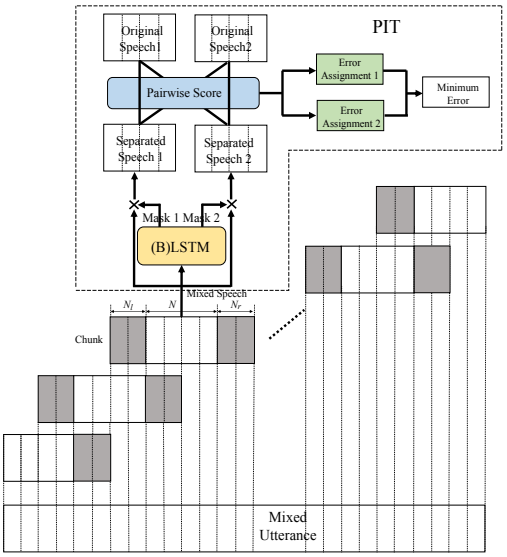

Fig. 1. The architecture of cPIT, whose main idea is to split an utterance into chunks. The main chunk has N frames, with appended N_l left and N_r right contextual frames. For the first/last chunk of each utterance, no left/right contextual frames are appended. The appended frames are only used to provide context information and do not generate error signals during training. LC-BLSTM does not need left contextual frames.

where ϕ^* is the permutation that minimizes the separation error:

$$\phi^* = \arg\min_{\phi \in \mathcal{P}} \sum_{s=1}^{S} ||\hat{\mathbf{M}}_s \circ |\mathbf{Y}| - |\mathbf{X}|_{\phi} \circ \cos(\theta_{\mathbf{Y}} - \theta_{\mathbf{X}_{\phi}})||_F^2 \quad (3)$$

where $\mathcal{P}$ is the set of all $S!$ permutations. As illustrated in the area surrounded by dotted lines in Figure 1, PIT computes MSE between estimated and target magnitudes using all possible permutations, and the minimum error is used for back propagation.

C. CSC-BLSTM and LC-BLSTM

BLSTM is often used in uPIT-based speech separation systems for its capacity of modeling long time dependency in forward and backward directions [8], [13], [12], [14], [15], [16], [17], [18]. BLSTM has a high latency as long as the utterance. Since BLSTM is one of the state-of-the-art acoustic models on ASR tasks [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], there have been some relative works to address the latency problem [25], [26], [33].

In [25], context-sensitive chunk (CSC) with left and right contextual frames to initialize the forward and backward LSTM is used for both training and decoding, which reduces the decoding latency from utterance-level to chunk-level. CSC-BLSTM is extended to LC-BLSTM by directly carrying over the left contextual information from previous chunk of the same utterance into current chunk [26], where the latency can be determined by the number of right contextual frames and modified by users to get a trade-off between performance and latency.

TABLE I
FOR SIMPLICITY AND CLARITY, SOME DENOTATIONS ARE LISTED.

Denotation	Model	Training Strategy	Inferring Method
uPIT-LSTM	LSTM	utterance-level PIT	utterance-level
uPIT-BLSTM	BLSTM	utterance-level PIT	utterance-level
uPIT-CSC-BLSTM			chunk-level (CSC)
uPIT-LC-BLSTM			chunk-level (LC)
cPIT-BLSTM	BLSTM	chunk-level PIT	utterance-level
cPIT-CSC-BLSTM			chunk-level (CSC)
cPIT-LC-BLSTM			chunk-level (LC)

TABLE II
SDR IMPROVEMENTS (DB) FOR ORIGINAL MIXTURES AND UPIT-(B)LSTM BASELINES. M/F STANDS FOR MALE/FEMALE.

PIT Model	Average	M-F	F-F	M-M
Mixtures	0.06	0.06	0.07	0.06
uPIT-LSTM [8]	7.0	-	-	-
uPIT-BLSTM [8]	9.4	-	-	-
Our uPIT-LSTM	7.16	9.02	3.80	5.77
Our uPIT-BLSTM	9.46	10.90	7.61	8.11

III. PROPOSED METHODS

A. Chunk-level PIT

As illustrated in Figure 1, the proposed cPIT splits an utterance into context-sensitive chunks, where main chunks (without contextual frames) do not overlap. Since the lengths of chunks are very close (no longer than $N_l + N + N_r$), we do not need to do zero padding frequently during training, so the training can be sped up significantly when compared to uPIT. Besides, we evaluate whether cPIT is beneficial for chunk-level inference.

B. cPIT-LC-BLSTM and uPIT-LC-BLSTM

Inference can also be done at the utterance level or chunk level. If we simply infer at the chunk level, i.e. use CSC-BLSTM, the output streams of main chunks in the same utterance are spliced to compose utterance-level separated results. However, permutation may change across neighboring chunks. For instance, in two-speaker case, the output permutation may be 1-1 (the first output stream corresponds to the first speaker) and 2-2 in previous chunk, while it may change to 1-2 (the first output stream corresponds to the second speaker) and 2-1 in current chunk. If the output streams of these two chunks are simply spliced, the separated speech may face the speaker inconsistency problem.

The first proposed method to alleviate the problem is to replace CSC-BLSTM with LC-BLSTM. The only difference between them is that LC-BLSTM copies the forward hidden states from previous chunk directly and does not need left contextual frames, while CSC-BLSTM uses left contextual frames to initialize forward LSTM. They both need right contextual frames to initialize backward LSTM. There are two advantages in using LC-BLSTM. Firstly, computational cost is reduced by $\frac{N_l}{N_l+N+N_r}$ with the removing the left initialization operation. Secondly, it helps keep the forward hidden states continuous across neighboring chunks, which is beneficial for modeling a broader left context and to some extent alleviates the speaker inconsistency problem.

With the model trained at the chunk level or utterance level, cPIT-LC-BLSTM or uPIT-LC-BLSTM method is obtained. Besides, some other denotations are also listed in Table I.

C. Inter-chunk Speaker Tracing

In [7], there is a huge performance gap between default assign (without ST) and optimal assign (assuming that all speakers are correctly traced), which can be reduced with ST algorithms.

In this paper, a simple ST algorithm is adopted to exploit the overlapping frames between two neighboring chunks. Let us denote $\mathbf{O}^1_{t-1}$ and $\mathbf{O}^2_{t-1}$ as two output streams of overlapping frames in previous chunk, and $\mathbf{O}^1_t$ and $\mathbf{O}^2_t$ as those in current chunk. We compute pairwise MSE as PIT does:

$$E_1 = \text{MSE}(\mathbf{O}^1_{t-1}, \mathbf{O}^1_t) + \text{MSE}(\mathbf{O}^2_{t-1}, \mathbf{O}^2_t) \quad (4)$$

$$E_2 = \text{MSE}(\mathbf{O}^1_{t-1}, \mathbf{O}^2_t) + \text{MSE}(\mathbf{O}^2_{t-1}, \mathbf{O}^1_t) \quad (5)$$

If $E_1 > \alpha E_2$, we consider there exists a change of output permutation, where α is the penalty factor and set to 2.0 by default. There are two reasons to set α to 2.0 instead of 1.0. Firstly, we believe that the probability of permutation changing is smaller than that of the same permutation, especially when LC-BLSTM is used. Secondly, more robustness is added into the system. For example, if both speakers are silent in the overlapping frames, the two output streams are almost similar, and then setting α to 1.0 may lead to a false detection of permutation changing.

IV. EXPERIMENTS AND RESULTS

A. Experimental Setup

The dataset is the same as the two-talker mixed dataset in [3], [4], [6], [7], [8], except that the sample rate is 16 kHz. It is generated by mixing the utterances in WSJ0 corpus at various signal-to-noise ratios uniformly chosen between 0 dB and 5 dB, and has 20k, 5k and 3k mixtures for training, validation and testing respectively. The 30-hour training set and 10-hour validation set are generated from `si_tr_s` using 49 male and 51 female speakers. The 5-hour testing set is generated from `si_dt_05` and `si_et_05` using 16 speakers.

The input to the model is the magnitude of mixture's STFT, which is extracted with a frame size 32 ms and 16 ms shift, and has 257 frequency sub-band. The PIT model has a fully-connected layer, 3 (B)LSTM layers and two output layers. The dimension of LSTM cell is 640, so each BLSTM layer has 1280 units. We use ReLU [34] as the activation function of two output layers, and two output masks have the same dimension as that of input. The input mixed magnitude is multiplied by two masks respectively to get two separated magnitudes, and then use the phase of mixed speech and inverse STFT to get the separated audios. Signal-to-distortion ratio (SDR) [35] is used to evaluate the performance of separation.

Tensorflow [36] is used to build the systems. The validation set is only used for tuning the learning rate as it will be halved by 0.7 when the loss on validation set increases. The initial learning rate is 0.0005. Dropout is applied to BLSTM layers

TABLE III
AVERAGE SDR IMPROVEMENTS (DB) FOR BLSTM TRAINED WITH CPIT OR UPIT. SPEAKER TRACING (ST) IS USED TO IMPROVE THE PERFORMANCE OF CSC-BLSTM AND LC-BLSTM. THE ABSOLUTE GAP (ABS. GAP) IS COMPARED TO UPIT-BLSTM.

Method	SDR	Abs. Gap
cPIT-CSC-BLSTM	8.00	-1.46
cPIT-CSC-BLSTM + ST	8.72	-0.74
cPIT-LC-BLSTM	8.61	-0.85
cPIT-LC-BLSTM + ST	8.71	-0.75
cPIT-BLSTM	8.73	-0.73
uPIT-CSC-BLSTM	8.09	-1.37
uPIT-CSC-BLSTM + ST	9.10	-0.36
uPIT-LC-BLSTM	8.98	-0.48
uPIT-LC-BLSTM + ST	9.16	-0.30
uPIT-BLSTM	9.46	-

with a rate 0.5. For faster evaluation, all models are trained for 32 epochs. When training at the utterance level, each minibatch contains 10 random utterances. When training at the chunk level, each minibatch contains 100 random chunks.

B. uPIT Baselines

Table II presents the SDR improvements of baseline uPIT-(B)LSTM. It is obvious that uPIT-BLSTM is far better than uPIT-LSTM. It is also noticed that the same-gender separation is more difficult, especially female-female separation. Although the size of our model is smaller than that in [8] and we trained for fewer epochs, the obtained results are comparable with the baseline results in [8].

C. cPIT v.s. uPIT

As described before, the model for inference can be trained at the utterance level or chunk level. We trained one BLSTM at the chunk level with $N_l = 50, N = 100, N_r = 50$, and compared it with the baseline BLSTM trained at the utterance level. We present the SDR results in Table III. Here, we consider four inferring methods: cPIT-CSC-BLSTM, uPIT-CSC-BLSTM, cPIT-BLSTM and uPIT-BLSTM. Generally, the model trained at the utterance level performs better.

D. CSC-BLSTM v.s. LC-BLSTM

Here we compare two inferring methods: CSC-BLSTM and LC-BLSTM. As illustrated in Table III, LC-BLSTM outperforms CSC-BLSTM significantly, with improvements of 0.61 dB when using the model trained at the chunk level and 0.89 dB when using the model trained at the utterance level. Besides, uPIT-LC-BLSTM outperforms cPIT-LC-BLSTM significantly.

To prove LC-BLSTM helps alleviate the speaker inconsistency problem, an example is shown in Figure 2. As illustrated, there exists a change of permutation in the last chunk when using uPIT-CSC-BLSTM. Also, the spectrograms separated by uPIT-LC-BLSTM and uPIT-BLSTM are quite similar.

E. Inter-chunk Speaker Tracing

As illustrated in Table III, ST can further improve the performance of both CSC-BLSTM and LC-BLSTM. For the model trained at the chunk level, ST improves the cPIT-CSC-BLSTM and cPIT-LC-BLSTM by 0.72 dB and 0.1 dB respectively. For the model trained at the utterance level, ST improves the uPIT-CSC-BLSTM and uPIT-LC-BLSTM by 1.01 dB and 0.18 dB respectively, where the improvements are more obvious.

Finally, uPIT-LC-BLSTM with ST achieves the best results of chunk-level inference, which is slightly worse than that of uPIT-BLSTM with a gap 0.3 dB, but is significantly better than that of uPIT-LSTM with a gain of 2.0 dB.

TABLE IV
AVERAGE SDR IMPROVEMENTS (DB) AND LATENCY (DEFINED AS $16 \times N_r$ MS AS THAT IN [26]) FOR UPIT-LC-BLSTM.

Method	N_r	SDR	Abs. Gap	Latency (ms)
uPIT-LC-BLSTM	0	8.76	-0.70	0
uPIT-LC-BLSTM + ST	10	8.81	-0.55	160
	25	9.02	-0.44	400
	35	9.07	-0.39	560
	50	9.16	-0.30	800
	100	9.26	-0.20	1600
uPIT-BLSTM		9.46	-	utterance-level
uPIT-LSTM		7.16	-2.30	0

F. Trade-off between Latency and Performance

The latency of above chunk configuration is 50×16 ms $= 800$ ms (defined as $16 \times N_r$ ms as that in [26]), which is quite high for low-latency applications. Here, we keep N_l and N fixed (Note N_l is useless for LC-BLSTM), and change the value of N_r to evaluate the performance with different latency, and the results are illustrated in Table IV.

Generally, SDR decreases as N_r decreases. Note that when $N_r = 0$, we cannot perform ST for LC-BLSTM, since there is no overlapping frame. Even though N_r is 0, uPIT-LC-BLSTM still outperforms uPIT-LSTM with a gain of 1.6 dB, and has a gap of 0.7 dB when compared to uPIT-BLSTM.

V. CONCLUSIONS

In this paper, we explored uPIT-LC-BLSTM on single-channel multi-talker speech separation task to reduce the latency of uPIT-BLSTM from utterance-level to chunk-level. To reduce the SDR gap between uPIT-LC-BLSTM and uPIT-BLSTM, inter-chunk speaker tracing was proposed to further alleviate the permutation changing problem across neighboring chunks. Besides, a trade-off between inference latency and separation performance could be obtained according to the actual demand by setting the number of right contextual frames. In the future, we plan to combine the uPIT-LC-BLSTM with cross entropy for directly multi-talker speech recognition [12], [13], [14], [15], [16].

ACKNOWLEDGEMENTS

This work is partially supported by the National Natural Science Foundation of China (Nos. 11590774, 11590770).

REFERENCES

[1] D. Wang and G. J. Brown, *Computational auditory scene analysis: Principles, algorithms, and applications*. Wiley-IEEE press, 2006.
[2] M. N. Schmidt and R. K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," in *Ninth International Conference on Spoken Language Processing*, 2006.

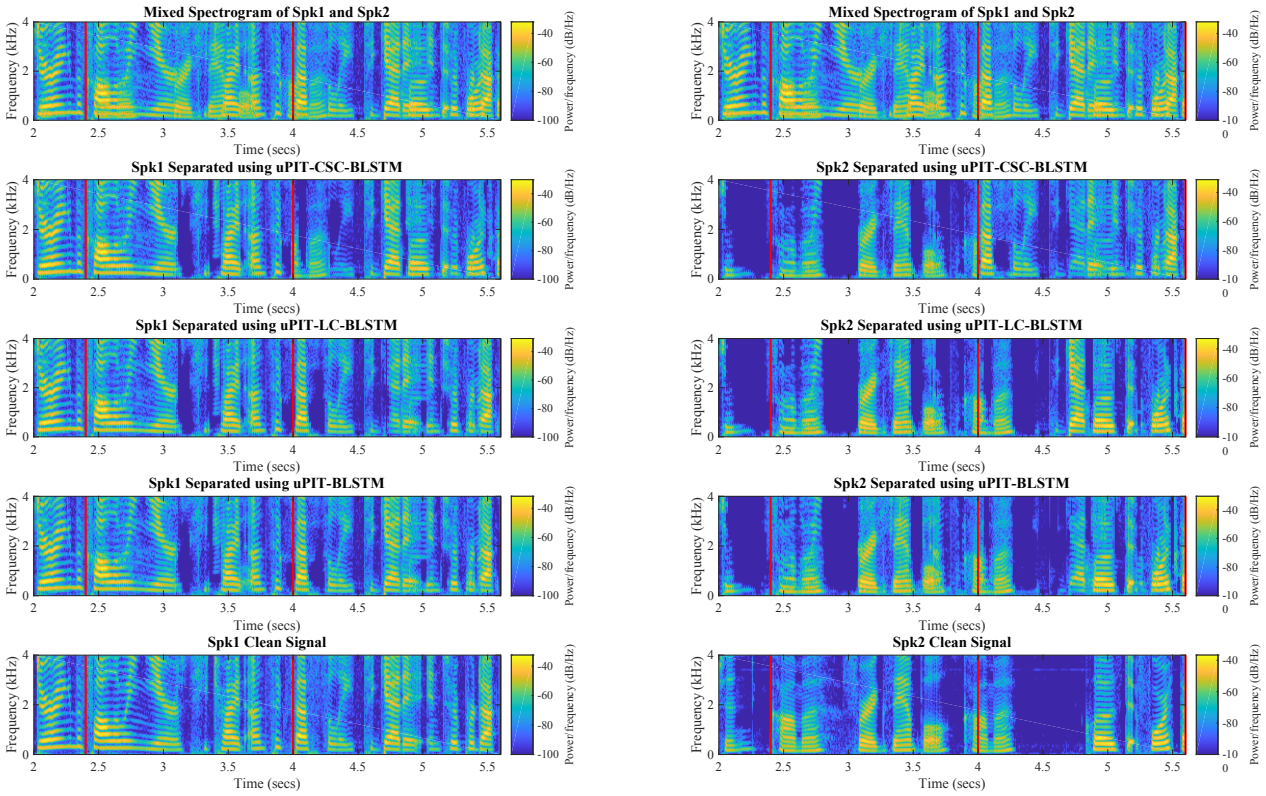

Fig. 2. Permutation changing problem is alleviated by LC-BLSTM. The mixed/clean spectrograms of two speakers are shown in the first/last row respectively. The second, third and last rows are the separated spectrograms using uPIT-CSC-BLSTM, uPIT-LC-BLSTM and uPIT-BLSTM respectively. The vertical red lines are the borders of chunks. In the second row, the speakers exchanges in the last chunk when using CSC-BLSTM, and it is solved in the third row when using LC-BLSTM.

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

[4] Y. Isik, J. Le Roux, Z. Chen, S. Watanabe, and J. R. Hershey, "Single-channel multi-speaker separation using deep clustering," in *Proc. Interspeech*, 2016, pp. 545–549.

[5] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *Proc. ICASSP*. IEEE, 2017, pp. 246–250.

[6] Y. Luo and N. Mesgarani, "Tasnet: time-domain audio separation network for real-time, single-channel speech separation," in *Proc. ICASSP*. IEEE, 2018, pp. 696–700.

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

[8] M. Kolbæk, D. Yu, Z.-H. 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, 2017.

[9] Y. Luo and N. Mesgarani, "Real-time single-channel dereverberation and separation with time-domain audio separation network," in *Proc. Interspeech*, 2018, pp. 342–346.

[10] S. Haykin and Z. Chen, "The cocktail party problem," *Neural computation*, vol. 17, no. 9, pp. 1875–1902, 2005.

[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[12] D. Yu, X. Chang, and Y. Qian, "Recognizing multi-talker speech with permutation invariant training," in *Proc. Interspeech*, 2017, pp. 2456–2430.

[13] Y. Qian, X. Chang, and D. Yu, "Single-channel multi-talker speech recognition with permutation invariant training," *Speech Communication*, vol. 104, pp. 1–11, 2018.

[14] Z. Chen, J. Droppo, J. Li, and W. Xiong, "Progressive joint modeling in unsupervised single-channel overlapped speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 1, pp. 184–196, 2018.

[15] X. Chang, Y. Qian, and D. Yu, "Adaptive permutation invariant training with auxiliary information for monaural multi-talker speech recognition," in *Proc. ICASSP*. IEEE, 2018, pp. 5974–5978.

[16] T. Tan, Y. Qian, and D. Yu, "Knowledge transfer in permutation invariant training for single-channel multi-talker speech recognition," in *Proc. ICASSP*. IEEE, 2018, pp. 5340–5344.

[17] H. Seki, T. Hori, S. Watanabe, J. Le Roux, and J. R. Hershey, "A purely end-to-end system for multi-speaker speech recognition," in *Proc. the 56th Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2018, pp. 2620–2630.

[18] X. Chang, Y. Qian, K. Yu, and S. Watanabe, "End-to-end monaural multi-speaker asr system without pretraining," in *Proc. ICASSP (Accepted)*, 2019.

[19] C. Xu, W. Rao, X. Xiao, E. S. Chng, and H. Li, "Single channel speech separation with constrained utterance level permutation invariant training using grid lstm," in *Proc. ICASSP*. IEEE, 2018, pp. 6–10.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[22] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with lstm recurrent networks," *Journal of machine learning research*, vol. 3, no. Aug, pp. 115–143, 2002.

[23] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[24] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*. IEEE, 2013, pp. 6645–6649.

[25] K. Chen and Q. Huo, "Training deep bidirectional lstm acoustic model for lvcsr by a context-sensitive-chunk bptt approach," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 7, pp. 1185–1193, 2016.

[26] Y. Zhang, G. Chen, D. Yu, K. Yaco, S. Khudanpur, and J. Glass, "Highway long short-term memory rnns for distant speech recognition," in *Proc. ICASSP*. IEEE, 2016, pp. 5755–5759.

[27] S. Xue and Z. Yan, "Improving latency-controlled blstm acoustic models for online speech recognition," in *Proc. ICASSP*. IEEE, 2017, pp. 5714–5718.

[28] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, "The microsoft 2017 conversational speech recognition system," in *Proc. ICASSP*. IEEE, 2018, pp. 5934–5938.

[29] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Achieving human parity in conversational speech recognition," *arXiv preprint arXiv:1610.05256*, 2016.

[30] G. Cheng, D. Povey, L. Huang, J. Xu, S. Khudanpur, and Y. Yan, "Output-gate projected gated recurrent unit for speech recognition," in *Proc. Interspeech*, 2018, pp. 1793–1797.

[31] W. Li, G. Cheng, F. Ge, P. Zhang, and Y. Yan, "Investigation on the combination of batch normalization and dropout in blstm-based acoustic modeling for asr," in *Proc. Interspeech*, 2018, pp. 2888–2892.

[32] K. Han, A. Chandrashekaran, J. Kim, and I. Lane, "Densely connected networks for conversational speech recognition," in *Proc. Interspeech*, 2018, pp. 796–800.

[33] V. Peddinti, Y. Wang, D. Povey, and S. Khudanpur, "Low latency acoustic modeling using temporal convolution and lstms," *IEEE Signal Processing Letters*, vol. 25, no. 3, pp. 373–377, 2018.

[34] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning*, 2010, pp. 807–814.

[35] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE transactions on audio, speech, and language processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[36] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: a system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.