DUAL-PATH RNN FOR LONG RECORDING SPEECH SEPARATION

Chenda Li[1], *Yi Luo*[2], *Cong Han*[2], *Jinyu Li*[3], *Takuya Yoshioka*[3], *Tianyan Zhou*[3],
Marc Delcroix[4], *Keisuke Kinoshita*[4], *Christoph Boeddeker*[5],
Yanmin Qian[1], *Shinji Watanabe*[6], *Zhuo Chen*[3]

[1]Shanghai Jiao Tong University, [2]Columbia University, [3]Microsoft Corporation,
[4]NTT Corporation, [5]Paderborn University, [6]Johns Hopkins University

ABSTRACT

Continuous speech separation (CSS) is an arising task in speech separation aiming at separating overlap-free targets from a long, partially-overlapped recording. A straightforward extension of previously proposed sentence-level separation models to this task is to segment the long recording into fixed-length blocks and perform separation on them independently. However, such simple extension does not fully address the cross-block dependencies and the separation performance may not be satisfactory. In this paper, we focus on how the block-level separation performance can be improved by exploring methods to utilize the cross-block information. Based on the recently proposed dual-path RNN (DPRNN) architecture, we investigate how DPRNN can help the block-level separation by the interleaved intra- and inter-block modules. Experiment results show that DPRNN is able to significantly outperform the baseline block-level model in both offline and block-online configurations under certain settings.

Index Terms— Continuous speech separation, long recording speech separation, dual-path RNN

1. INTRODUCTION

The task of speech separation has long been an active research topic for speech processing. The permutation problem of label assignment is one of the fundamental problem in the supervised learning [1] for speech separation. Deep clustering (DPCL) was proposed in [2] to tackle the label permutation problem by using an affinity-based objective function, which is invariant to the permutation of speakers. Following the DPCL, the deep attractor net (DANet) [3, 4] enables end-to-end training and improves the performance. Permutation invariant training (PIT) [5,6] is another effective yet simple way to solve the label permutation problem. In the last five years, the performance has been significantly advanced thanks to the progress in neural-based speech separation [6–23].

Most existing systems consider the problem configuration where a short, segmented mixture utterance containing at least two sources is provided as the system input. However, such configuration is often invalid in two aspects in real-world applications. First, the assumption that the input mixture is always well-segmented in short segments is unrealistic since daily conversations are continuous and can last for a relatively long time. Second, the assumption that there are at least two speakers in the mixture is typically not true, as in scenarios such as group meetings where the overlap ratio is in general less than 30% [24], that there are often long segments with a single speaker speaking. This means that in real-world communications where the sessions can be long and the overlap ratio can be small, existing systems need to be properly modified to match the new data distribution.

There are several possible ways to process long recordings. One straightforward approach would be to apply separation approach to the whole recording. However, this would require knowing the total number of speakers in the mixture and using separation approach that can handle a potentially large number of speakers. This is challenging for most neural-network-based approaches. Iterative separation frameworks have recently been proposed to address an arbitrary number of speakers [25–27]. In [28], iterative separation has been extended to long recording by using a block-online processing and making use of speaker embeddings estimated from previous blocks to connect local processing blocks of a same speaker. This separation scheme is realized using an iterative extraction approach [25,26] combined with target speaker extraction capabilities [29,30]. Although this approach can handle an arbitrary number of speakers, the computational complexity might significantly increase when the total numbers of active speakers in the recording is high. Moreover, when there are rapid speaker changes within a short period, the bias information might be inaccurate and the number of extraction iterations might be large. Recently, the continuous separation scheme (CSS) [31, 32] proposed to handle long recordings by splitting them into fixed-length blocks and perform block-level separation independently. After the block-level separation finishes, the outputs from different blocks are concatenated, or *stitched*, into long output streams where each stream only contains non-overlapping speech. By using short enough blocks, given the overlapping characteristics of real recordings, it is reasonable to assume the number of speak-

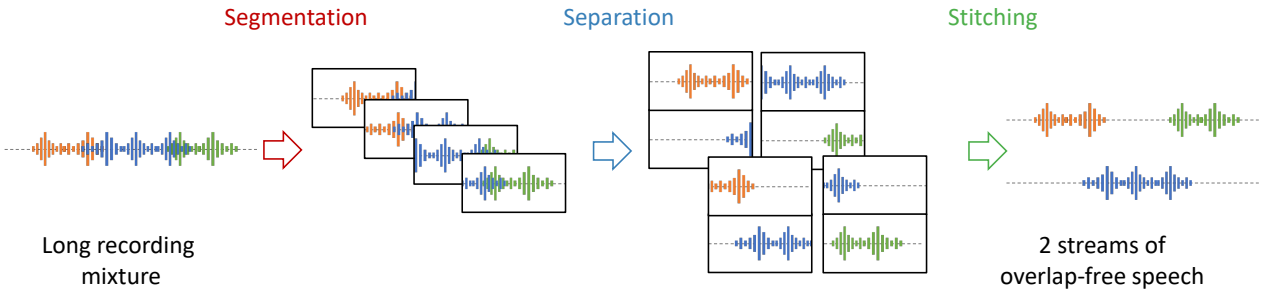

Fig. 1. A typical pipeline for continuous speech separation systems. The *segmentation* step splits the long recording into short blocks. The *separation* step performs separation on the blocks. The *stitching* step concatenates the block-level separation outputs to long streams which only contains nonoverlapped targets.

ers to be less than 2 or 3, and thus the existing sentence-level separation systems trained for a small number of overlapping speakers can be applied. Diarization performed on the separated output can handle the speaker counting and association problem. However, though shown effective for practical datasets [32, 33], CSS perform separation on a block-level, and does not utilize the cross-block dependencies and thus may have limited separation performance. For example, the speaker activity information can be helpful in deciding the number of active sources in adjacent blocks - one separation output from a single-speaker block should be a silent signal, and such silent signal can further be utilized by the next block for both speaker activity detection and separation.

A recently proposed neural network architecture, the dual-path RNN (DPRNN) [34], tried to address the long-sequence modeling problem by using interleaved RNN layers in different time scales. DPRNN splits a long sequence into shorter, fixed-length blocks and applies intra- and inter-block RNNs to perform local- and global-processing iteratively. Such segmentation allows each of the RNNs to only receive a small number of time steps in the entire sequence, and alleviates the optimization difficulty during training. It is easy to find that the segmentation applied in DPRNN is identical to the segmentation of long recording in the local-level separation systems described above, and the inter-block RNN is a good candidate for modeling the cross-block dependencies.

In this paper, we investigate how DPRNN can be applied in the CSS problem to help improve the block-level separation performance. Instead of modeling the recording in time-domain as with the original DPRNN, we use a conventional time-frequency (T-F) masking framework with a longer short-time Fourier transform (STFT) window size for the consideration of computational complexity and memory usage. We use the same configuration in DPRNN where a *LocalRNN* is used for intra-block processing, a *GlobalRNN* is used to capture the inter-block information, and the two RNNs are applied in an interleaved way for a larger model capacity. We also explore the block-online configuration where the inter-block RNN is unidirectional instead of bidirectional, which allows the system to be deployed into real-time applications such as real-time meeting transcription systems. Note that a similar approach has been proposed in a very recent work [35], however the initial experiments in [35] only considered a maximum number of two active speakers in the entire recording that consists of close talk utterances, and performed recording-level training with aligned output permutations across all the blocks. Here we consider the more realistic case where the number of active speakers in the recordings can be large. Moreover, we focus on block-level training and do not assume the same output permutations in different blocks. Experiment results show that the DPRNN architecture is able to significantly improve the separation performance compared with the block-level baselines in both offline and block-online configurations, proving the effectiveness of DPRNN in the task of CSS.

The rest of the paper is organized as follows. Section 2 introduces the problem definition of the CSS and the configuration of the baseline models. Section 3 describes the DPRNN architecture for CSS. Section 4 presents the detailed experiment setups. The results are analyzed and discussed in Section 5. Section 6 concludes the paper.

2. CONTINUOUS SPEECH SEPARATION: PROBLEM AND BASELINE

Figure 1 shows the overall pipeline for a conventional continuous speech separation (CSS) system. It typically contains three steps: *segmentation*, *separation*, and *stitching*. The segmentation step splits the long recording into blocks, and the adjacent blocks typically contain an overlapped region. The separation step applies any separation system to the blocks and generates separated outputs. The stitching step concate-

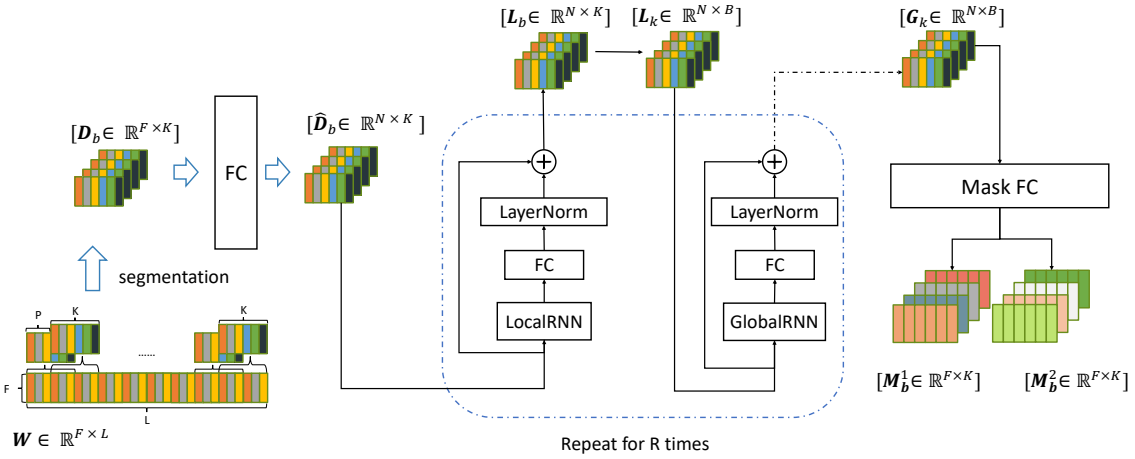

Fig. 2. Model design for the DPRNN architecture. After segmentation on the long input spectrogram, a linear bottleneck layer is first applied to transform the feature dimension. The transformed feature is then passed to the *LocalRNN* layer for intra-block processing, and then the output is passed to the *GlobalRNN* layer for inter-block processing. The procedure is repeated for multiple stacks, and the output from the last stack is sent to a output layer with ReLU activation to generate the T-F masks for the two targets.

nates the adjacent outputs based on any similarity measure on the overlapped regions, and generates long output streams within which the target sources are all nonoverlapping.

To formally describe the three steps, the input mixture is denoted by $\mathbf{w} \in \mathbb{R}^{M\times T}$ where M denotes the number of microphones and T denotes the total length. In this paper we consider the single-channel scenario where $M = 1$. In T-F masking systems, the magnitude spectrogram $\mathbf{W} \in \mathbb{R}^{F\times L}$ is calculated by the STFT on $\mathbf{w}$, where F denotes the number of frequency bins and L denotes the number of frames. The segmentation step splits $\mathbf{W}$ into B blocks $\mathbf{D}_b \in \mathbb{R}^{F\times K}, b = 1,\dots B$, with block size K and block hop size P, resulting in a 3-D tensor $\mathbf{T} = [\mathbf{D}_1,\dots,\mathbf{D}_B] \in \mathbb{R}^{F\times K\times B}$. The separation step generates C outputs from each of the blocks, denoted by $\mathbf{O}_b \in \mathbb{R}^{F\times K\times C}, b = 1,\dots B$, where C is typically assumed fixed. Given the assumption that the overlap ratio in real-world meetings is small and the number of overlapped speakers is typically less than three, C can be set to 2 to satisfy the requirement. This assumption also allows us to bypass the problem of separating too many sources in the entire recording, as in real-world meetings the total number of active speakers can be more than 10 and asking the model to generate so many outputs may cause troubles in both model complexity and optimization. The stitching step merges the outputs from adjacent blocks by comparing the similarities of the overlapped regions between the former block outputs and all permutations of the latter block outputs. The overlapped regions of the adjacent blocks are averaged, similar to the overlap-and-add processing. The final time-domain outputs after the inverse STFT operation are denoted as $\mathbf{SO}_c \in \mathbb{R}^{M\times T}, c = 1, 2$, where each output only contains separated, nonoverlapping speech signals.

3. DPRNN FOR CONTINUOUS SPEECH SEPARATION USING LONG CONTEXT

DPRNN can be easily applied to the CSS problem. A DPRNN layer contains an intra-block RNN and an inter-block RNN, where the intra-block RNN, which we refer to as the *LocalRNN*, is applied on each of the blocks independently, and the inter-block RNN, which we refer to as the *GlobalRNN*, receives the outputs from the LocalRNNs and perform cross-block processing. Multiple DPRNN layers can be stacked to increase the depth of the entire network. Figure 2 shows the flowchart of the DPRNN architecture.

We follow the original model design of DPRNN. Before the first DPRNN layer, a bottleneck fully-connected (FC) layer is applied on the segmented blocks $\mathbf{D}_b$ and maps them to $\hat{\mathbf{D}}_b \in \mathbb{R}^{N\times K}$, where N is the dimension of bottleneck feature. This operation is mainly for consideration on the computational complexity. The transformed spectrograms are then fed into the LocalRNN for intra-block processing:

$$\mathbf{E}_b = f_l(\hat{\mathbf{D}}_b) \tag{1}$$

where $\mathbf{E}_b \in \mathbb{R}^{H\times K}$ is the output of the LocalRNN and $f_l(\cdot)$ is the mapping function defined by the LocalRNN, H is the hidden dimension of RNN. The output $\mathbf{E}_b \in \mathbb{R}^{H\times K}$ is then passed to another FC layer to generate $\hat{\mathbf{E}}_b \in \mathbb{R}^{N\times K}$, which matches the feature dimension of $\hat{\mathbf{D}}_b$, and a residual connec-

tion is added between $\hat{\mathbf{D}}_b$ and the layer-normalized (LN) output $\hat{\mathbf{E}}_b$:

$$\mathbf{L}_b = \hat{\mathbf{D}}_b + LN(\hat{\mathbf{E}}_b) \tag{2}$$

where $\mathbf{L}_b \in \mathbb{R}^{N\times K}$ is the final output of the LocalRNN. All outputs from all the blocks form another 3-D tensor $\mathbf{L} = [\mathbf{L}_1, \ldots, \mathbf{L}_B] \in \mathbb{R}^{N\times K\times B}$, and the GlobalRNN is applied on $\mathbf{L}$ across the third (block index) dimension. By rewriting $\mathbf{L}$ into $\mathbf{L}_k = \mathbf{L}[:, k, :] \in \mathbb{R}^{N\times B}, k = 1, \ldots, K$, the transform of GlobalRNN can be written as:

$$\mathbf{Q}_k = f_g(\mathbf{L}_k) \tag{3}$$

where $\mathbf{Q}_k \in \mathbb{R}^{H\times B}$ is the output of the GlobalRNN and $f_g(\cdot)$ is the mapping function defined by the GlobalRNN. Similarly, a FC layer is applied on $\mathbf{Q}_k$ to generate $\hat{\mathbf{Q}}_k$ that matches the dimension of $\mathbf{L}_k$, and LN is applied before the residual connection:

$$\mathbf{G}_k = \hat{\mathbf{L}}_k + LN(\hat{\mathbf{Q}}_k) \tag{4}$$

where $\mathbf{G}_k \in \mathbb{R}^{N\times B}$ is the final output of the GlobalRNN, and can be fed into the next DPRNN layer for further processing. The output of the last DPRNN layer, denoted by $\hat{\mathbf{G}} \in \mathbb{R}^{N\times K\times B}$, is passed to a FC layer with ReLU activation to generate two T-F masks for each block $\mathbf{M}_b^1, \mathbf{M}_b^2 \in \mathbb{R}^{F\times K}, b = 1, \ldots, B$. The masks are then applied to $\mathbf{D}_b$ to generate the spectrograms of the separated outputs $\mathbf{S}_b^1, \mathbf{S}_b^2 \in \mathbb{R}^{F\times K}$.

Since we focus on the block-level separation performance during training, we optimize the model by calculating the signal-quality measures between the block-level outputs and references. Note that this implies that the output permutation in different blocks can be different. A commonly-used training objective in many recent systems is the scale-invariant signal-to-distortion ratio (SI-SDR) [36]. However, as there are many single-speaker blocks in the long recordings and SI-SDR cannot take an all-zero signal as the reference, we use the signal-to-noise ratio (SNR) as our training objective:

$$\mathrm{SNR}(\mathbf{s}, \hat{\mathbf{s}}) = 10\log_{10}\frac{\|\hat{\mathbf{s}}\|^2}{\|\hat{\mathbf{s}} - \mathbf{s}\|^2} \tag{5}$$

where $\mathbf{s}, \hat{\mathbf{s}}$ are the estimated and reference waveforms, respectively. SNR has been used as the training objective in sentence-level reverberant separation tasks and has shown at least on par performance as SI-SDR [37].

For the optional stitching step, the stitching can be applied on either the output spectrograms or waveforms. Empirically we find that using the output spectrograms leads to a better stitching accuracy than the waveforms.

4. EXPERIMENT DETAILS

4.1. Data simulation

We simulate a noisy reverberant dataset for all our experiments. We randomly generate 3000 and 300 rooms for training and development, respectively. The sample rate for all recordings is 16 kHz. The length and width of the rooms are randomly sampled between 5 and 12 meters, and the height is randomly sampled between 2.5 and 4.5 meters. A microphone is randomly placed in the room, and its location is constrained to be within 2 meters of the room center. The height of the microphone is randomly sampled between 0.4 and 1.2 meters. We randomly sample 10 candidate speaker locations with the constraint that the locations are at least 0.5 meters away from the room walls, and the height of the speakers are between 1 and 2 meters. The reverberation time is uniformly sampled between 0.1 and 0.5 seconds. Each of the room configuration is used for 3 times, where 3-5 speakers from the LibriSpeech corpus [38] are randomly sampled and placed at randomly sampled locations from the 10 candidate speaker locations. Multiple sentences are sampled from the selected speakers and mixed at a uniformly sampled overlap ratio between between 30% and 60%. A simulated Gaussian noise signal is then added to the mixture at a random SNR of 0 to 20 dB. The total length of the mixture is 90 seconds. This leads to a total of 9000 training recordings and 900 development recordings.

To further evaluate the generalization ability of the models with respect to the recording length, we simulate three extra test sets containing 2, 5, and 8 speakers with duration of 60 seconds, 150 seconds, and 240 seconds, respectively. The overlap ratio of these test sets is all 30%.

4.2. Model configurations

For feature extraction, we use a 512-point STFT and 256-point hop size in STFT to extract the spectrograms. The block size K in the segmentation step is selected from $\{50, 100, 200\}$, corresponding to $\{0.8, 1.6, 3.2\}$ seconds, respectively, to investigate the effect of the block size on the separation performance. The baseline model we use applies a deep RNN architecture where we stack multiple LocalRNN layers. The LocalRNNs in DPRNN models are all bidirectional LSTM (BLSTM) layers with 512 hidden units in each direction. For the offline configuration in DPRNN, the configuration of GlobalRNN is identical as that of the LocalRNN, while for block-online configuration the GlobalRNN is a unidirectional LSTM with 512 hidden units. All DPRNN models contain 2 LocalRNNs and 2 GlobalRNNs which are interleaved with each other. The LocalRNNs in the small baseline models are also bidirectional LSTM (BLSTM) layers with 512 hidden units in each direction, while in large baseline models they contain 768 hidden units in each direction. This is to match the total model sizes of the large baseline and DPRNN models for a fair comparison. All baseline models contain 2 LocalRNNs.

We use the *PaderTorch*[1] framework as the toolkit for the experiments. The Adam optimizer [39] is used with the initial

[1] https://github.com/fgnt/padertorch

Table 1. Pre-stitching SNR (dB) on blocks with different overlap ratios for different models.

Models	Model size (M)	Block size (seconds)	Block-online	Overlap ratio (%)				
				0	0-25	25-50	50-75	75-100
Local	7.0	3.2	Yes	16.8	**8.8**	9.6	8.5	7.8
Local	13.6		Yes	16.8	8.6	9.4	8.5	7.7
Global	13.9		No	**17.0**	8.3	9.7	**8.7**	**8.0**
Global	10.4		Yes	16.9	8.6	**9.8**	**8.7**	**8.0**
Local	7.0	1.6	Yes	17.1	**8.5**	9.1	9.1	7.9
Local	13.6		Yes	17.1	8.4	9.1	9.1	8.0
Global	13.9		No	**17.2**	8.3	**9.6**	**9.7**	**8.4**
Global	10.4		Yes	17.1	8.3	9.4	9.6	8.3
Local	7.0	0.8	Yes	16.1	**8.0**	8.3	8.5	7.8
Local	13.6		Yes	16.1	**8.0**	8.2	8.4	7.6
Global	13.9		No	**16.2**	7.9	**8.4**	**8.9**	**8.4**
Global	10.4		Yes	**16.2**	7.8	8.2	8.8	**8.4**

learning rate of 0.001. The learning rate is decayed by 0.95 for every two epochs. We train all the models for 100 epochs with batch size of 2. Note that each sample in the batch contains a single long recording, which further contains B blocks of length K.

4.3. Evaluation

We evaluate the models in both pre-stitching and post-stitching measures. The pre-stitching evaluation directly calculates the block-level SNR scores between the outputs and references. In this case, we report the results for different overlap ratios independently. The post-stitching evaluation stitches the outputs from all the blocks into recording-level streams, and segments the streams by oracle segmentation information (i.e. onset and offset information for each sentence). The segmented streams are then compared with the segmented reference signals.

5. RESULTS AND DISCUSSIONS

We start with the experiment results on the pre-stitching evaluation. Table 1 presents the SNR scores on the baseline (Local) and DPRNN (Global) models. We first notice that the performance of the models on the single-speaker blocks (0 overlap ratio) and low-overlap-ratio blocks (0-25% overlap ratio) are all comparable, and it indicates that cross-block information is not really necessary when the speaker activation is already sparse within a block. The performance of Global models are consistently better in high-overlap-ratio blocks. Note that although the overall overlap ratio in real-world long recordings can be small, e.g. below 30%, such segmentation on the recordings will always lead to blocks with very high overlap ratios. As the block size decreases, the number of single-speaker blocks and high-overlap-ratio blocks will increase, thus the performance improvement on such high-overlap-ratio blocks is important. Moreover, the results of the block-online configurations of the Global models are always on par or better than the offline configurations across all settings and overlap conditions. This shows that the block-online DPRNN model is a good option for streaming separation applications which require a lower system latency. We also observe that increasing the model size of the Local models does not lead to a better performance. The reason behind this observation it yet to explore, but one possible explanation is that increasing the depth of the model might be more beneficial than increasing the width. We leave this topic as the future work.

We then consider the effect of stitching and perform post-stitching evaluation. Table 2 provides the results on the matched development set (3-5 speakers, 90-second long) and unmatched extra test sets (2, 5, 8 speakers, 60, 150, 240-second long, respectively). Note that following the existing evaluation pipeline in *PaderTorch*, we use signal-to-distortion ratio (SDR) in the *mir_eval* toolbox as the metric [40]. For the matched development set, we observe that the Global models are always better than the Local models especially when the block size becomes smaller. The performance of the Global models are consistent across 3.2 and 1.6-second long blocks, while the performance of Local models has an obvious degradation. This shows that the cross-block modeling module is able to assist the stitching step to achieve a higher accuracy. For the unmatched test set, we first find that the performance of the Local models are on par or better than the Global models on the 3.2-second long blocks. It indicates that when the block size is long enough, block-level separation systems can already achieve a satisfactory performance. When the block size decreases, the Global models become better than the Local models, and the improvement is significant in 0.8-second long blocks. As the block-online configuration of the Global models achieves on par performance as the offline configuration, we can conclude that the block-online DPRNN

Table 2. Post-stitching SDR (dB) for different models.

Models	Model size (M)	Block size (seconds)	Block-online	Number of speakers			
				3-5	2	5	8
Local	7.0	3.2	Yes	11.3	14.0	**13.0**	**12.9**
Local	13.6		Yes	11.2	13.9	**13.0**	12.7
Global	13.9		No	11.3	14.1	12.8	12.4
Global	10.4		Yes	**11.4**	**14.3**	12.9	12.0
Local	7.0	1.6	Yes	10.8	13.6	12.7	**12.5**
Local	13.6		Yes	10.8	13.6	12.6	12.4
Global	13.9		No	**11.4**	**14.0**	**12.9**	12.2
Global	10.4		Yes	11.2	13.8	12.7	11.8
Local	7.0	0.8	Yes	8.9	11.5	10.6	10.3
Local	13.6		Yes	8.8	11.5	10.3	9.9
Global	13.9		No	**10.1**	**12.7**	**11.5**	**10.8**
Global	10.4		Yes	9.7	12.3	11.2	10.5

models are better designs especially in applications where a low system latency is required.

Another observation on the 8-speaker recordings is that the performance of the Global models are worse than the Local models in both 3.2 and 1.6-second long blocks. One possible explanation is that as the Global models only received up to 5 speakers during training, the models may not be able to properly capture the cross-block dependencies with more speakers. The Local models do not have such issue and have a consistent performance. Moreover, the speaker activation in 8-speaker recordings might be sparser, e.g. each speaker may only contain two or three sentences, which makes the model hard to generalize. More experiments are necessary to correctly identify the problem.

On the other hand, the absolute performance of both Local and Global models becomes much worse for 0.8-second block size across all datasets. One reason for this observation is that as the stitching step relies on the overlapped regions between adjacent blocks, a small block size contains a short overlapped region and may make the stitching inaccurate. This problem can be alleviated by adjusting the length of the overlapped regions in adjacent blocks or by using the post-stitching outputs as the system outputs for optimization, and we also leave it as future work.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we investigated the problem of continuous speech separation (CSS) where multiple speakers needed to be separated from a long recording instead of a single sentence. Existing sentence-level separation systems can be directly applied to the CSS problem by segmenting the long recording into shorter blocks and performing separation on the blocks in parallel, however such pipeline cannot utilize the cross-block dependencies which can be helpful on improving the performance. Based on the recent progress on dual-path RNN (DPRNN) architecture for long-sequence modeling, we explored how DPRNN can be applied in the CSS problem to better make use of the inter-block information. We conducted experiments on simulated noisy reverberant separation datasets with various configurations, and the results showed that the DPRNN models were able to outperform the block-level baseline models in both offline and block-online configurations. The improvements were significant on small block sizes, indicating that DPRNN is especially suitable for applications where a small system latency is required.

Future works can be done in multiple aspects. From the model design perspective, a more complicated design, as the ones in [35], can be evaluated on the same datasets. Different choices on the STFT window size can also be explored, and time-domain processing is also a nature extension of the current time-frequency domain processing pipeline. From the parameter setting perspective, how to enable larger models to achieve better performance is an important topic. From the recording-level processing perspective, incorporate stitching into the training pipeline to improve the stitching accuracy is also interesting.

7. ACKNOWLEDGEMENT

The work presented here was carried out during the 2020 Jelinek Memorial Summer Workshop on Speech and Language Technologies at Johns Hopkins University, which was supported with unrestricted gifts from Microsoft (Research and Azure), Amazon (Alexa and AWS), and Google. Chenda Li and Yanmin Qian are also supported by the China NSFC project (No. 62071288 and No.U1736202).

8. REFERENCES

[1] 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.

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

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

[4] Yi Luo, Zhuo Chen, and Nima Mesgarani, "Speaker-independent speech separation with deep attractor network," *IEEE/ACM Trans. ASLP.*, vol. 26, no. 4, pp. 787–796, 2018.

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

[6] Morten Kolbæk, Dong Yu, Zheng-Hua Tan, Jesper Jensen, Morten Kolbaek, 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 (TASLP)*, vol. 25, no. 10, pp. 1901–1913, 2017.

[7] Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R Hershey, "Single-channel multi-speaker separation using deep clustering," *Interspeech 2016*, pp. 545–549, 2016.

[8] Yi Luo, Zhuo Chen, John R Hershey, Jonathan Le Roux, and Nima Mesgarani, "Deep clustering and conventional networks for music separation: Stronger together," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 61–65.

[9] Zhong-Qiu Wang, Jonathan Le Roux, and John R Hershey, "Alternative objective functions for deep clustering," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*, 2018.

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

[11] Yi Luo and Nima Mesgarani, "Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 27, no. 8, pp. 1256–1266, 2019.

[12] Yuzhou Liu and DeLiang Wang, "Divide and conquer: A deep casa approach to talker-independent monaural speaker separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 27, no. 12, pp. 2092–2102, 2019.

[13] Jonathan Le Roux, Gordon Wichern, Shinji Watanabe, Andy Sarroff, and John R Hershey, "The phasebook: Building complex masks via discrete representations for source separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2019 IEEE International Conference on*. IEEE, 2019, pp. 66–70.

[14] Chenglin Xu, Wei Rao, Eng Siong Chng, and Haizhou Li, "Time-domain speaker extraction network," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 327–334.

[15] Yi Luo, Zhuo Chen, and Takuya Yoshioka, "Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation," *arXiv preprint arXiv:1910.06379*, 2019.

[16] Eliya Nachmani, Yossi Adi, and Lior Wolf, "Voice separation with an unknown number of multiple speakers," *arXiv preprint arXiv:2003.01531*, 2020.

[17] Neil Zeghidour and David Grangier, "Wavesplit: End-to-end speech separation by speaker clustering," *arXiv preprint arXiv:2002.08933*, 2020.

[18] Rongzhi Gu, Jian Wu, Shi-Xiong Zhang, Lianwu Chen, Yong Xu, Meng Yu, Dan Su, Yuexian Zou, and Dong Yu, "End-to-end multi-channel speech separation," *arXiv preprint arXiv:1905.06286*, 2019.

[19] Peidong Wang, Zhuo Chen, Xiong Xiao, Zhong Meng, Takuya Yoshioka, Tianyan Zhou, Liang Lu, and Jinyu Li, "Speech separation using speaker inventory," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 230–236.

[20] Yi Luo, Enea Ceolini, Cong Han, Shih-Chii Liu, and Nima Mesgarani, "Fasnet: Low-latency adaptive beamforming for multi-microphone audio processing," *arXiv preprint arXiv:1909.13387*, 2019.

[21] Zhong-Qiu Wang, Scott Wisdom, Kevin Wilson, and John R Hershey, "Sequential multi-frame neural beamforming for speech separation and enhancement," *arXiv preprint arXiv:1911.07953*, 2019.

[22] Rongzhi Gu, Shi-Xiong Zhang, Lianwu Chen, Yong Xu, Meng Yu, Dan Su, Yuexian Zou, and Dong Yu, "Enhancing end-to-end multi-channel speech separation via spatial feature learning," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7319–7323.

[23] Yi Luo, Zhuo Chen, Nima Mesgarani, and Takuya Yoshioka, "End-to-end microphone permutation and number invariant multi-channel speech separation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6394–6398.

[24] Özgür Çetin and Elizabeth Shriberg, "Analysis of overlaps in meetings by dialog factors, hot spots, speakers, and collection site: Insights for automatic speech recognition," in *Ninth international conference on spoken language processing*, 2006.

[25] Keisuke Kinoshita, Lukas Drude, Marc Delcroix, and Tomohiro Nakatani, "Listening to each speaker one by one with recurrent selective hearing networks," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5064–5068.

[26] Naoya Takahashi, Sudarsanam Parthasaarathy, Nabarun Goswami, and Yuki Mitsufuji, "Recursive speech separation for unknown number of speakers," *Interspeech 2019*, pp. 1348–1352, 2019.

[27] Jing Shi, Xuankai Chang, Pengcheng Guo, Shinji Watanabe, Yusuke Fujita, Jiaming Xu, Bo Xu, and Lei Xie, "Sequence to multi-sequence learning via conditional chain mapping for mixture signals," *arXiv preprint arXiv:2006.14150*, 2020.

[28] Thilo von Neumann, Keisuke Kinoshita, Marc Delcroix, Shoko Araki, Tomohiro Nakatani, and Reinhold Haeb-Umbach, "All-neural online source separation, counting, and diarization for meeting analysis," in *Acoustics, Speech and Signal Processing (ICASSP), 2019 IEEE International Conference on*. IEEE, 2019, pp. 91–95.

[29] Marc Delcroix, Katerina Zmolikova, Keisuke Kinoshita, Atsunori Ogawa, and Tomohiro Nakatani, "Single channel target speaker extraction and recognition with speaker beam," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5554–5558.

[30] Kateřina Žmolíková, Marc Delcroix, Keisuke Kinoshita, Tsubasa Ochiai, Tomohiro Nakatani, Lukáš Burget, and Jan Černockỳ, "Speakerbeam: Speaker aware neural network for target speaker extraction in speech mixtures," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 800–814, 2019.

[31] Takuya Yoshioka, Hakan Erdogan, Zhuo Chen, and Fil Alleva, "Multi-microphone neural speech separation for far-field multi-talker speech recognition," in *Proc. IEEE ICASSP*. IEEE, 2018, pp. 5739–5743.

[32] T. Yoshioka, I. Abramovski, C. Aksoylar, Z. Chen, M. David, D. Dimitriadis, Y. Gong, I. Gurvich, X. Huang, Y. Huang, A. Hurvitz, L. Jiang, S. Koubi, E. Krupka, I. Leichter, C. Liu, P. Parthasarathy, A. Vinnikov, L. Wu, X. Xiao, W. Xiong, H. Wang, Z. Wang, J. Zhang, Y. Zhao, and T. Zhou, "Advances in online audio-visual meeting transcription," in *Proc. IEEE ASRU*. IEEE, 2019, pp. 276–283.

[33] Zhuo Chen, Takuya Yoshioka, Liang Lu, Tianyan Zhou, Zhong Meng, Yi Luo, Jian Wu, Xiong Xiao, and Jinyu Li, "Continuous speech separation: Dataset and analysis," in *Proc. IEEE ICASSP*. IEEE, 2020, pp. 7284–7288.

[34] Yi Luo, Zhuo Chen, and Takuya Yoshioka, "Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation," in *Proc. IEEE ICASSP*. IEEE, 2020, pp. 46–50.

[35] Keisuke Kinoshita, Thilo von Neumann, Marc Delcroix, Tomohiro Nakatani, and Reinhold Haeb-Umbach, "Multi-path rnn for hierarchical modeling of long sequential data and its application to speaker stream separation," *arXiv preprint arXiv:2006.13579*, 2020.

[36] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R. Hershey, "Sdr – half-baked or well done?," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 626–630.

[37] Tsubasa Ochiai, Marc Delcroix, Rintaro Ikeshita, Keisuke Kinoshita, Tomohiro Nakatani, and Shoko Araki, "Beam-tasnet: Time-domain audio separation network meets frequency-domain beamformer," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6384–6388.

[38] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5206–5210.

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

[40] Colin Raffel, Brian McFee, Eric J Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, Daniel PW Ellis, and C Colin Raffel, "mir_eval: A transparent implementation of common mir metrics," in *In Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR*. Citeseer, 2014.