Dual-Path Modeling With Memory Embedding Model for Continuous Speech Separation

Chenda Li, *Graduate Student Member, IEEE*, Zhuo Chen, *Member, IEEE*, and Yanmin Qian, *Senior Member, IEEE*

***Abstract*—Continuous speech separation (CSS) aims at separating overlap-free targets from a long, partially-overlapped recording. Though it has shown promising results, the origin CSS framework does not consider cross-window information and long-span dependency. To alleviate these limitations, this work introduces two novel methods to implicitly and explicitly capture the long-span knowledge, respectively. We firstly apply the dual-path (DP) modeling architecture for the CSS framework, where the within and across window information are jointly modeled by alternating stacked local-global processing modules. Secondly, to further capture the long-span dependency, we introduce a memory-based model for CSS. An additional memory pool is designed to extract embedding from each small window, and the inter-window commutation is established above the memory embedding pool through an attention mechanism. This memory-based model can precisely control what information needs to be transferred across the windows, thus leading to both improved modeling capacity and interpretability. The experimental results on the LibriCSS dataset show that both strategies can well capture the long-span information of the continuous speech and significantly improve system performance. Moreover, further improvements are observed with the integration of these two methods.**

***Index Terms*—Continuous speech separation, dual-path modeling, memory pool, overlap ratio predictor.**

I. Introduction

THE "cocktail party problem" [1] depicts a special challenge in acoustic signal processing, where the signal to be processed consists of audio mixtures from various sound sources and multiple active speakers, as if it is recorded from a dinner party, and the target of the problem is to separate or recognize the content from each individual speaker. Due to its multi-speaker nature, the "cocktail party problem" often causes severe performance degradation to conventional speech processing systems such as automatic speech recognition (ASR), which usually hold the assumption that only a single active speaker exists in the input signal.

Neural network based speech separation is often considered as a remedy for this challenge. Starting from the deep clustering (DPCL) [2] and permutation invariant training (PIT) [3], [4], tremendous progresses have been made in recent years for both the speech separation [2]–[18] and multi-talker ASR [19]–[24]. With the latest methods, on benchmark dataset such as WSJ0-2mix [2], the separated speech gets more than 20 dB signal-to-distortion ratio (SDR) improvement [17], [18].

Researchers also explore speech separation in more realistic scenarios [25]–[28]. Compared with artificially generated speech mixtures (e.g., WSJ0-2mix), the recordings differ in three major aspects and poses additional challenges in realistic meeting scenarios.

- The real-world meeting recordings usually have a longer duration, in which may include multiple utterances without labeled-out boundaries.
- The overall overlap ratio is lower in nature meeting conversations, which is usually below 20% [29].
- The number of active speakers in meeting recordings varies in a wide range. Thus, assuming a fixed number of speakers is inappropriate for realistic scenarios.

In [30]–[36], different approaches for separating a large or uncertain number of speakers have been investigated on high-overlap utterance-level separation. However, directly applying these methods to long and low-overlap recordings may introduce a lot of overhead. Because assigning one output channel for each speaker may lead to a lot of sparse output channels, especially when the number of speakers is large in a long session. Serialized Output Training (SOT) [24], [37] tries to model the partially overlapped speech with serialized output token in an end-to-end ASR system, in which the separation progress is implicitly done in the ASR encoder. SOT is designed for ASR tasks and it is difficult to be applied in speech separation training. In general, an effective speech separation model for all three aspects above is still being actively studied.

Continuous speech separation (CSS) framework has been found promising to handle real world multi-speaker recordings [25], [26]. In CSS, the long recordings are segmented into length-fixed windows, each processed individually. When the window size is small (e.g. $<$3 seconds), it is reasonable to assume that there is a maximal number of active speakers in a single window [26], [29]. Thus, in the CSS framework, the

Manuscript received September 4, 2021; revised January 30, 2022; accepted March 24, 2022. Date of publication April 7, 2022; date of current version April 29, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2021ZD0201504, in part by China NSFC projects under Grants 62122050 and 62071288, and in part by Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Andy W. H. Khong. *(Corresponding author: Yanmin Qian.)*

Chenda Li and Yanmin Qian are with the X-Lance Lab, Department of Computer Science and Engineering & MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University, Shanghai 200240, P. R. China (e-mail: lichenda1996@sjtu.edu.cn; yanminqian@gmail.com).

Zhuo Chen is with Microsoft, Redmond, WA 98052 USA (e-mail: zhuc@microsoft.com).

Digital Object Identifier 10.1109/TASLP.2022.3165712

separation model only needs to handle small segments with limited active speakers. And the separated speech of each window are reorganized into continuous overlap-free speech through an additional stitching algorithm and processed by downstream tasks such as diarization and ASR, without changing their single active speaker assumption. However, though CSS has shown effectiveness in the practical dataset [26], it processes each small window independently, which may be sub-optimal especially for long recording, where cross utterance clue is usually found beneficial for the separation in overlapped region [38]. Finding a proper way to model the cross-window information might be a step for better performance.

In this paper, we employ the dual-path (DP) modeling method for CSS, which is an extension of our previous works [39], [40]. The original dual-path recurrent neural network (DPRNN) [41] is proposed to solve the long sequential features of the time-domain models. It has been shown effective for utterance-level separation. Inspired by DPRNN, we explore the DP modeling methods for the CSS task in time-frequency (T-F) domain. Similar to DPRNN, we alternately stack the *intra-* and *inter-window* processing layers for local and global modeling, respectively. Both dual-path bidirectional long-short memory (BLSTM) [42] and dual-path Transformer [43] models are investigated. We adopt an improved sampling method to reduce the computation cost and improve the separation performance for DP models. We also optimize the design of the basic single-input-multi-output (SIMO) and single-input-single-output (SISO) modules based on the findings of a recent work [44].

For better control and analysis, we propose an attention-based method for cross-window information exchange to enhance the DP model, referred as memory-based model. In this design, an additional neural net extracts memory embeddings from each individual window is introduced, and the extracted embeddings form a memory pool. Cross-window communication is then established on the memory pool. An attention mechanism accesses the memory pool and chooses the most valuable information from it for local-window processing. The memory embedding net can be jointly trained with the separation task or use a pre-trained network on other tasks to introduce external knowledge.

The contribution of this paper can be summarized as follows:

- We developed the dual-path (DP) modeling for the CSS task. Both BLSTM- and Transformer-based are explored, and the SIMO-SISO srtucture is equipped. This DP model shows stronger modeling capacity in the CSS task.
- A refined convolution-based downsampling method is further utilized on the deep DP model, which can hugely reduce the computational cost as well as improve the separation performance, and the online processing DP models are also investigated for online CSS application.
- Another memory-based model is designed for the CSS framework. It can discover, encode and utilize valuable information from the long recordings to help the local window processing. This memory-based model is easier to introduce external knowledge with a pre-trained memory embedding net. Or, it can be jointly trained with the separation task.
- The proposed DP model architecture and memory-based model can be further integrated to capture the long-span knowledge of the continuous speech, and significant improvement can be obtained for continuous speech separation.

The rest of the paper is organized as follows: Section II reviews the basic CSS framework. The proposed dual-path methods and the optimized dual-path model for CSS are introduced in Section III. Then another proposed memory-based method and three different memory models will be described in Section IV. Section V presents the experimental results and detailed analysis, and finally Section VI gives the conclusion.

II. Continuous Speech Separation

The CSS method is firstly proposed in [45], and compared to the conventional utterance-wise speech separation, it is closer to the real-world scenario. As figure 1 shows, the CSS pipeline typically consists of three steps, including *segmentation*, *separation* and *stitching*.

Firstly the *segmentation* stage splits the continuous mixture into small windows with fixed window size and hop size. The hop size is usually smaller than the window size, and thus an overlapped region can be reserved between the adjacent windows. More formally, we denote the input mixture by $\mathbf{w} \in \mathbb{R}^{T\times M}$, where M is the number of channels[1] and T is the number of sampling points. The input magnitude spectrogram derived from $\mathbf{w}$ is denoted by $\mathbf{W} \in \mathbb{R}^{L\times F}$, where F is the number of frequency bins and L is the sequence length. In the *segmentation* stage, $\mathbf{W}$ is split into B windows $\mathbf{D}_b \in \mathbb{R}^{K\times F}, b = 1, \ldots, B$, with window size K and hop size P.

Then the *separation* stage performs separation on each segmented window, and generates C streams of overlap-free speech, where C is the number of output channels. For the partially overlapped meeting-style recordings [29], when the window size is short enough (< 3 seconds), it is reasonable to assume that at most two speakers are talking simultaneously in each small window [26], [28], [29], [46]. Thus, in the *separation* stage of this work, we can simply let $C = 2$ for the meeting-style dataset [26]. When the input audio only involves one speaker, one channel tends to output the original input or the enhanced speech, while the other channel tends to output silence or small noise. The separator follows the advance design recently proposed in [44]. It contains a single-input-multi-output (SIMO) module and a shared-weight singe-input-single-output (SISO) module. The SIMO module is a one-to-many mapping to generate 2 streams of intermediate features, which is similar to most of the current bind source separation (BSS) models [3], [11]. The SISO module takes the output of the SIMO module as input, and the 2 streams of SIMO output are processed in a weight-shared style. The mixed SIMO-SISO design can be considered as a pre-separation and post-enhancement pipeline. Both of the SIMO and SISO modules are made up with basic sequence modeling layers, e.g. bidirectional long-short memory (BLSTM) [42] and Transformer [43] layers.

[1] In this paper, we mainly discuss the single-channel condition where M is 1.

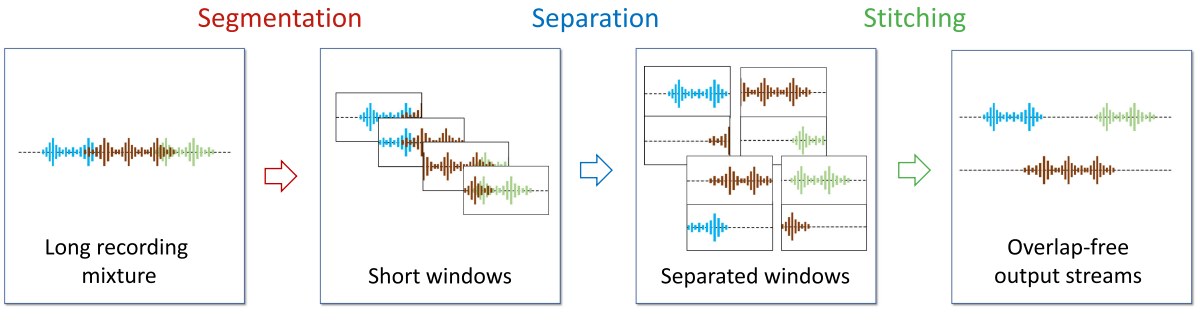


Fig. 1. The continuous speech separation (CSS) framework with three stages: segmentation, separation and stitching. The *segmentation* step splits the long recording into short windows. The *separation* step performs separation on each window. The *stitching* step concatenates the window-level separation outputs to long streams which only contains overlap-free targets.

To describe the SIMO-SISO separator more formally, we denote $f^{SIMO}(\cdot)$ and $f^{SISO}(\cdot)$ as the mapping function of the SIMO and SISO module, respectively. The SIMO operation can be presented as:

$$\{\mathbf{E}_b^c\} = f^{SIMO}(\mathbf{D}_b), \tag{1}$$

$$\hat{\mathbf{E}}_b^c = f^{SISO}(\mathbf{E}_\mathbf{b}^\mathbf{c}), \tag{2}$$

$$\hat{\mathbf{M}}_b^c = \mathrm{ReLU}(\mathrm{FC}(\hat{\mathbf{E}}_b^c)), \tag{3}$$

$$\hat{\mathbf{s}}_b^c = \mathrm{iSTFT}(\hat{\mathbf{M}}_b^c \odot \mathbf{D}_b, \Phi_b), \tag{4}$$

where $c \in \{1, 2\}$ denotes the 2 separated streams, Φ_b is the original phase spectrum of the b-th window. The output of the SISO module is then passed to a fully connected (FC) layer and ReLU activation function to estimate the time-frequency (T-F) masks $\hat{\mathbf{M}}_b^c$ for the magnitude spectrum $\mathbf{D}_b$ of each window. Then $\mathbf{D}_b$ is element-wisely multiply by $\hat{\mathbf{M}}_b^c$, and inverse STFT (iSTFT) is applied to obtain the predicted window-level time-domain signals $\hat{\mathbf{s}}_b^c$. The objective function $\mathcal{L}_{separator}$ is the window-level signal-to-noise ratio (SNR), and the permutation invariant training (PIT) is applied on each window:

$$\mathcal{L}_{separator} = -\mathrm{SNR}(\mathbf{s}_b^c, \hat{\mathbf{s}}_b^c) = -20log_{10}\frac{\|\hat{\mathbf{s}}_b^c\|}{\|\hat{\mathbf{s}}_b^c - \mathbf{s}_b^c\|}. \tag{5}$$

Finally, in the *stitching* stage, the adjacent separated windows are concatenated together to generate the continuous output streams for each speaker. Since the *separation* stage is single-input-multi-output (SIMO) and usually trained with PIT, the *stitching* stage has to keep the permutation consistency for adjacent windows. This can be solved by computing the similarity on the overlap region of adjacent windows as Algorithm.1 shows. In our implementation, the similarity is the inverse of the Euclidean distance on the T-F-masks.

Algorithm 1: Stitching Algorithm.

Input: The separated audio signals of each window: $\hat{\mathbf{s}}_b^c$ and their corresponding T-F masks, $\hat{\mathbf{M}}_b^c \in \mathbb{R}^{K\times F}$. Where $c = 1, 2$ denotes c-th output channel; $b = 1, \ldots, B$ denotes the b-th window.

Initialization: let $\mathbb{C}^1 = [\hat{\mathbf{s}}_1^1]$, $\mathbb{C}^2 = [\hat{\mathbf{s}}_1^2]$, $\pi = [1, 2]$
for b in $1, \ldots, B-1$ **do**
  $p^1 = \hat{\mathbf{M}}_b^1[K/2 : K]$; $p^2 = \hat{\mathbf{M}}_b^2[K/2 : K]$
  $n^1 = \hat{\mathbf{M}}_{b+1}^1[0 : K/2]$; $n^2 = \hat{\mathbf{M}}_{b+1}^2[0 : K/2]$
  sim.^{1} $= similarity(p^1, n^1) + similarity(p^2, n^2)$
  sim.^{2} $= similarity(p^1, n^2) + similarity(p^2, n^1)$
  if sim.$^1 \leq$ sim.^{2} **then**
    Swap π (i.e. $[1, 2] \rightarrow [2, 1]$ or $[2, 1] \rightarrow [1, 2]$)
  end if
  Append $\hat{\mathbf{s}}_{b+1}^1$ to $\mathbb{C}^{\pi(1)}$, append $\hat{\mathbf{s}}_{b+1}^2$ to $\mathbb{C}^{\pi(2)}$
end for
Perform overlap-add on $\mathbb{C}^1$ and $\mathbb{C}^2$ to obtain continuous output stream $\hat{\mathbf{s}}^1$ and $\hat{\mathbf{s}}^2$.

Output: The continuous output stream $\hat{\mathbf{s}}^1$ and $\hat{\mathbf{s}}^2$.

III. Dual-Path Modeling for Continuous Speech Separation

A. Dual-Path Modeling for Long Recording

The dual-path recurrent neural network (DPRNN) is proposed in [41], which achieved state-of-the-art performance on utterance-level speech separation. The authors use a small stride in the time-domain feature encoder. That leads to a very long feature sequence for a single utterance and DPRNN is designed to effectively model the very long sequence. We do not adopt the time-domain feature because we find that the magnitude T-F masking approaches are more robust to the downstream ASR task than the time-domain approaches [11], [41]. There are also many better T-F-domain features for speech enhancement and separation [14], [47], [48], which is beyond the scope of this article. Though the frame rate of T-F methods is usually much smaller than that in time-domain methods, in the CSS task, the length of the input feature is still very large due to the long audio duration. To the best of our knowledge, we are the first to apply the dual-path (DP) modeling methods for long sequence modeling in the CSS task [39].

Fig. 2(a) illustrates the backbone and the details of the DP separator. Similar to the baseline separator, it also follows the SIMO-SISO design. In the DP separator, both the SIMO and the SISO consist of DP modeling blocks. A DP modeling block is made up of two sequence modeling layers, which are named local modeling layer and global modeling layer. The local modeling layer focuses on the short-term modeling in a small window, while the global modeling layer captures the long-span information across all the windows in the long sequence. The magnitude

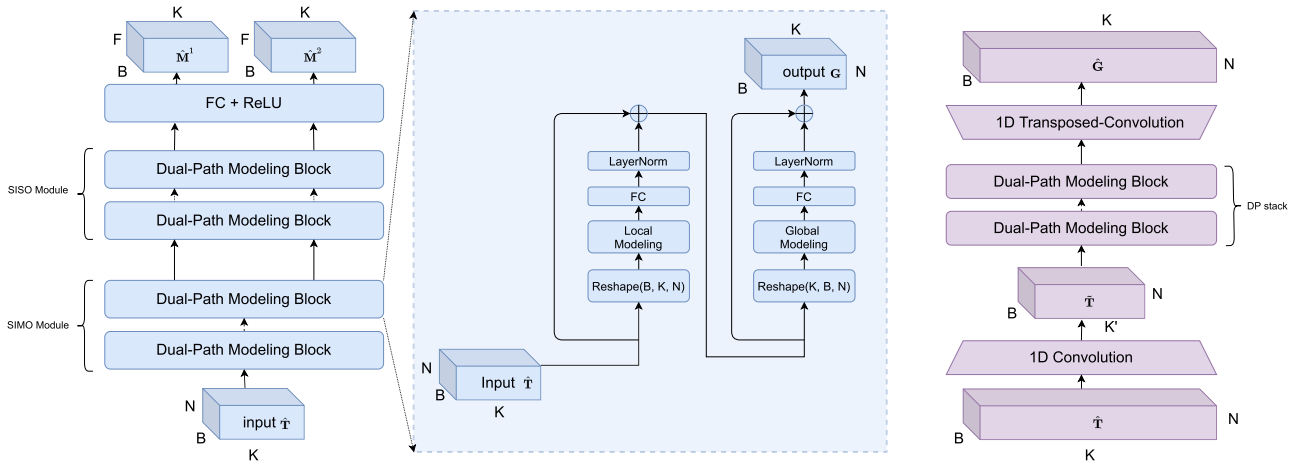


Fig. 2. (a) An illustration of the proposed dual-path modeling for long recording speech separation. Two sequential modeling are applied to the input feature. The sequential modeling can be performed with RNNs or transformers. The first model processes the input feature along the intra-window ($K \times N$) for the local processing, and the second model processes the received feature along the inter-window ($B \times N$) to capture the long-term dependency. The repeating of the DP block builds up the separation network. (b) The more refined dual-path modeling. The 1-D convolution layer downsamples the feature on the dimension K, and the size-reduced feature is then processed by the following DP blocks. Before the last DP block, a transposed 1-D convolution layer upsamples the feature to the original length.

spectrum of each window $\mathbf{D}_b \in \mathbb{R}^{F\times K}$ is firstly processed by a fully connected (FC) bottleneck layer to obtain the bottleneck feature $\hat{\mathbf{D}}_b \in \mathbb{R}^{K\times N}$. Denote the bottleneck input feature of the whole sequence as $\hat{\mathbf{T}} = [\hat{\mathbf{D}}_1, \ldots, \hat{\mathbf{D}}_B] \in \mathbb{R}^{B\times K\times N}$, the DP block models the 3-D tensor $\hat{\mathbf{T}}$ in dual-path style, while the baseline model that we discussed in section II processes each window $\hat{\mathbf{D}}_b$ independently. By alternating the local and global modeling layer in a deep DP network, the long span information can be passed across the different windows. Thus, the DP model is able to be optimized for the entire long sequence, while the baseline model is only optimized in each local window.

More formally, the local modeling layer for each window $\hat{\mathbf{D}}_b \in \mathbb{R}^{K\times N}$ be presented as:

$$\mathbf{E}_b = f_{\text{local}}\left(\hat{\mathbf{D}}_b\right), \tag{6}$$

where $f_{\text{local}}(\cdot)$ is the mapping function of the local modeling layer, $\mathbf{E}_b \in \mathbb{R}^{K\times H}$ is the processed feature and H is the hidden dimension. $\mathbf{E}_b$ is then processed by a bottleneck fully connect (FC) layer and a layer-norm (LN) [49] to build the residual connection [50]:

$$\mathbf{L}_b = \hat{\mathbf{D}}_b + \text{LN}(\text{FC}(\mathbf{E}_b)), \tag{7}$$

where $\mathbf{L}_b \in \mathbb{R}^{K\times N}$ is the output of the local modeling. Let $\mathbf{L} = [\mathbf{L}_1, \ldots, \mathbf{L}_B] \in \mathbb{R}^{B\times K\times N}$ be the 3-D tensor formed by all the windows' local modeling output. The 3-D tensor is then reshaped and indexed as $\mathbf{L}_k = \mathbf{L}[:, k, :] \in \mathbb{R}^{B\times N}, k = 1, \ldots, K$ before the global modeling. The global modeling is performed on each $\mathbf{L}_k$ along the dimension B:

$$\mathbf{Q}_k = f_{\text{global}}\left(\mathbf{L}_k\right), \tag{8}$$

where $f_{\text{global}}(\cdot)$ is the mapping function of global modeling, and $\mathbf{Q}_k \in \mathbb{R}^{B\times H}$ is the processed feature by global modeling layer. The bottleneck FC, layer-norm and residual connection are applied similar to that in the local modeling:

$$\mathbf{G}_k = \mathbf{L}_k + \text{LN}(\text{FC}(\mathbf{Q}_k)), \tag{9}$$

where $\mathbf{G}_k \in \mathbb{R}^{B\times N}$ is the output of the global modeling procedure. The output $\mathbf{G}_k$ is rearranged as $\mathbf{G} = [\mathbf{G}_1, \ldots, \mathbf{G}_K] \in \mathbb{R}^{B\times K\times N}$, which is also the input of the next DP block.

In theory, the local and global modeling layer can be any sequential modeling neural network layers. In this work, we have explored the BLSTM [42] and the Transformer encoder layer [43] as the sequential modeling layers. Respectively, the two kinds of CSS separators can be referred to as *DP-BLSTM* and *DP-Transformer*. The T-F mask estimation, separated waveform prediction, the training criterion, and the *stitching* procedure are the same as those in the basic CSS framework.

B. The More Refined Dual-Path Modeling

A refined strategy is designed to further improve the dual-path modeling. The method is illustrated in Fig. 2(b). Before feeding the 3-D intermediate feature $\hat{T} \in \mathbb{R}^{B\times K\times N}$ into the DP blocks, a 1-D convolution is performed on the dimension K. The convolution downsamples the intermediate feature $\hat{T}$ into smaller size $\tilde{T} \in \mathbb{R}^{B\times K'\times N}$, where $K = \lambda K'$ and λ is the sampling factor. The processed intermediate feature by DP stack is then passed by a transposed 1-D convolution and upsampled back to the tensor $\hat{G} \in \mathbb{R}^{B\times K\times N}$ which has the same shape as the input. The similar structure can be found in some previous works [51], [52]. There are two motivations for this convolution-based resampling in the DP model. First, it can effectively reduce the computation cost, especially when the number of DP blocks becomes large and a proper λ is chosen. Second, the convolution kernel makes the local information better presented in a single frame of one local window, which may benefit the global information interaction.

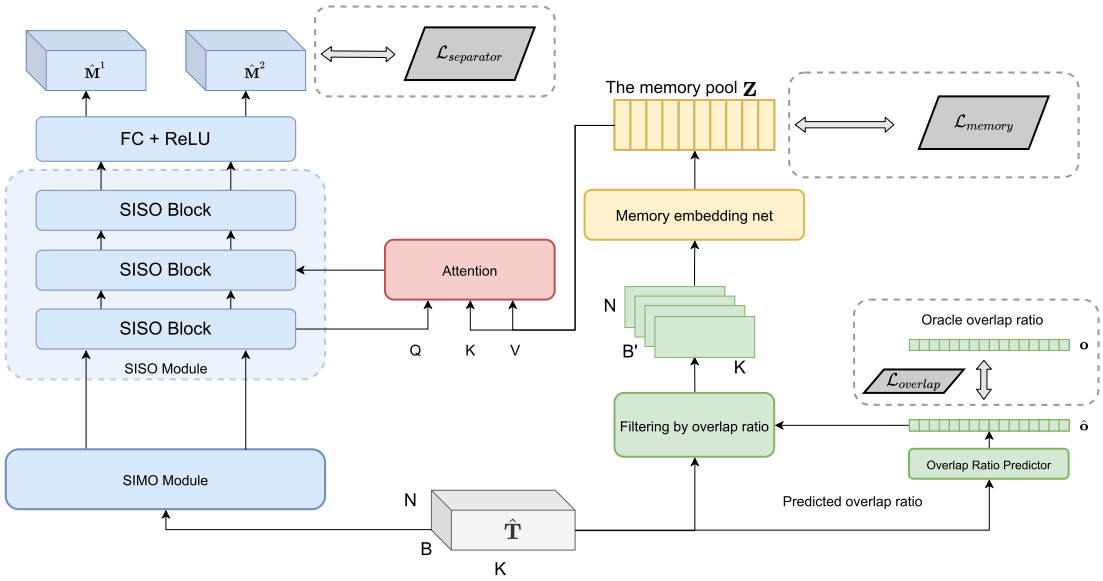


Fig. 3. The proposed memory-based continuous speech separation architecture. The memory embeddings are extracted from the windows with low overlap ratio, to form the memory pool for the long speech session. The SIMO module performs post-separation for the input feature. In the shared-weight SISO module, the hidden embeddings attend to the memory pool and get the global information from the whole speech session.

IV. Memory-Based Model for Long Recording Separation

The above DP method allows the model to capture the long-span information as well as the local information. Despite the strong power for the long sequence modeling and its delicate design for easy implementation, it still has some disadvantages. On one hand, there is no explicit control method for the global information in the DP models. How to utilize the high-value knowledge and ignore the ineffective information from the long span for the local window processing is implicitly learned by the neural network. On the other hand, the DP model is fundamentally an advanced sequential model, how to extend the model to better utilize external knowledge may not be considered when the model was being designed.

The proposed memory-based methods model the long-span information explicitly. Fig. 3 shows the entire architecture of the system. It mainly consists of four components: the SIMO-SISO-mixed separator, the window-level overlap ratio predictor, the memory embedding net to form the memory pool, and the attention-based memory reader to get access to the memory pool. The whole model is jointly trained with the objective function $\mathcal{L}$:

$$\mathcal{L} = \mathcal{L}_{separator} + \mathcal{L}_{overlap} + \mathcal{L}_{memory}, \tag{10}$$

where $\mathcal{L}_{separator}$ is the same as (5) that introduced in the basic CSS framework, $\mathcal{L}_{overlap}$ is the loss for overlap ratio predictor, which will be introduced in Section IV-B, and $\mathcal{L}_{memory}$ is the loss for memory pool construction, which is optional based on whether the memory neural net is pre-trained. $\mathcal{L}_{memory}$ will be introduced in Section IV-C.

A. SIMO-SISO-Mixed Separator

The backbone of the SIMO-SISO-mixed separator is similar to the systems that we introduced in previous sections. Both the regular BLSTM separator and DP-BLSTM separator will be compared as the baselines in this paper. The interaction of the separator and the memory pool happens in the SISO module. We believe that in the SISO module, the hidden embeddings of input mixture has been pre-separated by the SIMO module, and the disentangled feature is better than the mixed feature when we use it as the *query* in the attention mechanism.

B. Overlap Ratio Predictor

The overlap ratio predictor takes the input bottleneck feature $\hat{\mathbf{T}} \in \mathbb{R}^{B\times K\times N}$ as input, and predicts the overlap ratio for each window $b \in \{1, \ldots, B\}$. We borrow temporal convolutional network (TCN) [53], [54] block from the Conv-Tasnet [11] as the basic block to build the overlap ratio predictor. The network contains 4 TCN blocks and the dilation factors are $1, 2, 4, 8$. After the last TCN block, a mean pooling layer is used to output the predicted overlap ratios $\hat{\mathbf{o}} \in \mathbb{R}^B$. The overlap ratio predictor is jointly trained with the separation task. The loss function of the overlap ratio prediction is the mean square error (MSE):

$$\mathcal{L}_{overlap} = \mathrm{MSE}(\hat{\mathbf{o}}, \mathbf{o}), \tag{11}$$

where $\mathbf{o}$ is the oracle overlap ratio label of the training data.

C. Memory Embedding Net

The real conversations usually consists mostly of single talker speech, with a small part of multiple people talking simultaneously. For example, in [29], the authors show that in the meeting scenario, the overlap ratio is usually lower than 20%.

To make sure that the memory embedding net generates better and more representative memory embeddings, we perform filtering to the input bottleneck features $\hat{\mathbf{T}} = [\hat{\mathbf{D}}_1, \ldots, \hat{\mathbf{D}}_B] \in \mathbb{R}^{B\times K\times N}$ according to the predicted overlap ratios $\hat{\mathbf{o}}$ with a fixed threshold ϵ:

$$\tilde{\mathbf{T}} = \{\hat{\mathbf{D}}_b | \hat{\mathbf{o}}_b < \epsilon\}, \tag{12}$$

where $\tilde{\mathbf{T}} \in \mathbb{R}^{B'\times K\times N}$ is the filtered input features and B' is the number of "clean" windows. ϵ is set to 0.3 in our experiments. Then the memory embedding net takes $\tilde{\mathbf{T}}$ as input, and output the encoded memory pool $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_{B'}] \in \mathbb{R}^{B'\times N}$ for the current speech session.

The memory embedding net for CSS task can be constructed with different strategies, and three methods are proposed in this work. The first one is using pre-trained models, and the other two are learning from the data automatically and training jointly with the other modules.

1) Strategy#1. Pre-Trained Models: The memory embedding network can be pre-trained networks to encode specific knowledge. For example, when using a pre-trained speaker identify net, it encodes the speakers embeddings, and when using the encoder from end-to-end ASR, it may encodes some contextual-related embeddings. Furthermore, in the multi-channel application, it can also be used to encode spatial information. No matter which kind of the pre-trained model is utilized, the memory embedding net is initialized from these pre-trained models and the parameters are then fixed during the following model training. For example, some researches have shown that the separation can be guided and get better performance with speaker identification information [13], [38], [55]–[57]. In the meeting scenario, plenty of the small segmented windows are non-overlapped, and the speaker in those windows may overlap with others in another window. Thus, when the memory is a pre-trained speaker identification neural net, the speaker embeddings extracted from the non-overlapped region can ideally help the separation of the overlapped windows.

2) Strategy#2. Supervised Joint-Learning: In this strategy, the model parameters are learned with the other model modules. We assume that for each segmented window, the active speakers are labeled, i.e. for the filtered low-overlap window, the dominant speaker can be known. In this condition, we can design the speaker supervised objective functions for the memory embedding net:

$$\mathcal{L}_{intra\_spk} = \sum_{s}^{S}\sum_{i}^{\mathcal{P}(s)}\sum_{j}^{i}\gamma(\mathbf{z}_i^s, \mathbf{z}_j^s), \tag{13}$$

$$\mathcal{L}_{inter\_spk} = \sum_{s_1}^{S}\sum_{s_2\neq s_1}^{S}\omega(\bar{\mathbf{z}}^{s_1}, \bar{\mathbf{z}}^{s_1}), \tag{14}$$

$$\mathcal{L}_{memory} = \mathcal{L}_{intra\_speaker} + \lambda\mathcal{L}_{inter\_speaker}, \tag{15}$$

where $s = 1, \ldots, S$ denotes the s-th speaker in the session, which includes totally S active speakers. $\mathcal{P}(s)$ is the number of windows, in which speaker s is the dominant speaker. $\mathbf{z}_i^s$ is the i-th memory vector extracted from the s-th speaker's windows. $\bar{\mathbf{z}}^s$ is the mean vector of the s-th speaker's memory vectors. $\mathcal{L}_{memory}$ consists of two components with weight λ. $\mathcal{L}_{intra\_spk}$ constraints the memory vectors from the same speakers as close as possible in the embedding space. $\gamma(\cdot,\cdot)$ is the paired constraint that combines the MSE and cosine similarity:

$$\gamma(\mathbf{z}_1, \mathbf{z}_2) = \alpha_\gamma \,\mathrm{MSE}(\mathbf{z}_1, \mathbf{z}_2) - \beta_\gamma \,\mathrm{Cos\_sim}(\mathbf{z}_1, \mathbf{z}_2), \tag{16}$$

where α_γ and β_γ are manually set weights. $\mathcal{L}_{inter\_spk}$ pushes the memory centers from different speakers apart in the embedding space. The paired constraint $\omega(\cdot,\cdot)$ is defined as:

$$\omega(\mathbf{z}_1, \mathbf{z}_2) = \alpha_\omega \frac{1}{1+\mathrm{MSE}(\mathbf{z}_1, \mathbf{z}_2)} + \beta_\omega \,\mathrm{Cos\_sim}(\mathbf{z}_1, \mathbf{z}_2), \tag{17}$$

where α_ω and β_ω are manually set weights.

This memory embedding network is then trained jointly with all the other modules according to the (10).

3) Strategy#3. Unsupervised Joint-Learning: In contrast to the above supervised joint-learning, another unsupervised joint-leanring strategy is designed. In this condition, we assume that there is no speaker label in the dataset. In the training stage, the extracted memory embeddings are firstly clustered by K-means algorithm. Then, the objective functions are applied for the memory net training:

$$\mathcal{L}_{intra\_class} = \sum_{k}^{K}\sum_{i}^{\mathcal{P}(k)}\sum_{j}^{i}\gamma(\mathbf{z}_i^k, \mathbf{z}_j^k), \tag{18}$$

$$\mathcal{L}_{inter\_class} = \sum_{k_1}^{K}\sum_{k_2\neq k_1}^{K}\omega(\bar{\mathbf{z}}^{k_1}, \bar{\mathbf{z}}^{k_1}), \tag{19}$$

$$\mathcal{L}_{memory} = \mathcal{L}_{intra\_class} + \lambda\mathcal{L}_{inter\_class}, \tag{20}$$

where $\mathcal{L}_{intra\_class}$ and $\mathcal{L}_{inter\_class}$ are the constraints to pull the embeddings from the same clustering class closer and push the K clustering centers apart, respectively. The pair constraint $\gamma(\cdot,\cdot)$ and $\omega(\cdot,\cdot)$ are same as that described in (16) and (17). This memory embedding network is then trained jointly with all the other modules according to the (10).

D. Attention Based Memory Reader

We adopt the attention mechanism to design the memory reader, and access to the memory vectors for the CSS task. The attention mechanism can be formulated as a mapping function [43], which maps *query*, *key*, *value* to the weighted sum of *value*. In this task, the hidden embeddings $\hat{\mathbf{E}}_b \in \mathbb{R}^{K\times N}$ in the separator perform as the attention *query*, and the memory embeddings $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_{B'}] \in \mathbb{R}^{B'\times N}$ are the *key* and *value:*

$$\mathbf{A} = softmax\left(\frac{f_{key}(\hat{\mathbf{E}}_b)\cdot f_{query}(\mathbf{Z})^T}{\sqrt{N}}\right), \tag{21}$$

$$\mathbf{I}_b = \mathbf{A}\cdot f_{value}(\mathbf{Z}), \tag{22}$$

where f_{key}, f_{query}, f_{value} are the mapping functions of linear projection layer, and the operation $\{\cdot\}$ is the matrix multiplication. $\mathbf{A} \in \mathbb{R}^{K\times B'}$ is the attention weight matrix, and $\mathbf{I}_b \in \mathbb{R}^{K\times N}$ is the weighted sum of the mapped memory vectors. $\mathbf{I}_b$ is then

concatenated with the origin hidden embeddings $\hat{\mathbf{E}}_b$:

$$\tilde{\mathbf{E}}_b = \mathrm{FC}([\hat{\mathbf{E}}_b; \mathrm{LN}(\mathbf{I}_b)]), \tag{23}$$

where FC is the bottleneck layer which maps the hidden dimension to N, and LN is the layer-norm operation. $\tilde{\mathbf{E}}_b$ is the outputted hidden embedding. $\tilde{\mathbf{E}}_b$ is then processed by the down-stream SISO layers of the separator, and so the interaction of the separator and the memory pool is completed.

V. Experiments

A. Dataset

1) Evalutaion: In order to better evaluate the performance of the proposed CSS systems in real environment, we adopt the *LibriCSS* [26] as the testing set. It is a meeting-room-replay dataset derived from *Librispeech* [58]. The dataset was recorded in a real meeting room, with loud-speakers placed around a microphone array. *LibriCSS* is made up of *mini-sessions*, and there are 8 active speakers in each *mini-session*. The overlap ratio of *mini-sessions* ranges from 0 to 40%. In our evaluation, we firstly apply the separation to the recordings with our CSS systems, and then evaluate the word error rate (WER) for the processed recordings with an ASR system. The ASR model and the evaluation pipeline is provided by the toolkit[2] that released with *LibriCSS*. We also evaluate the DNSMOS score [59] on the separated speech. After applying CSS, we use the oracle utterance boundary to clip the utterances out from the continuous overlap-free output, and then perform the DNSMOS evaluation on these separated utterances.

2) Training Data: We simulated a 750-hour noisy and reverberant long-duration dataset based on *Librispeech* 16 kHz for CSS model training. The simulation aims to be as close to the *LibriCSS* as possible. The image method [60] is used to created the room impulse response (RIR), which is the same as our previous works [39], [40]. To simulated the meeting scene, we create 3000 and 300 virtual rooms for training and validation respectively. The length and width of the rooms are randomly sampled between 5 and 12 meters, and the height is between 2.5 to 4.5 meters. The simulated microphone is randomly placed in the 2×2 m^2 center area of the room, and the height of the microphone is between 1 and 2 meters. We randomly set 10 candidate locations in the simulated room, and one speaker from the *Librispeech* is assigned for each location. The locations of these speakers are at least 0.5 meters away from the wall, and the height is between 1 to 2 meters. The reverberation time is uniformly sampled in 0.1 to 0.5 seconds. After the RIR simulation for each room, we randomly simulate 10 *mini-sessions* in each room. The duration of each *mini-session* is 90 seconds or a little bit longer than it. We randomly choose 3 to 5 speakers from the 10 as the active speakers, and their corresponding utterances in the *Librispeech train-clean-100* and *train-clean-360* are convolved with the RIR and are used to assemble the *mini-session*. The overall overlap ratio of each *mini-session* is between 50 to 80%. It is noted that, the overlap ratio in the training set is higher than that in the *LibriCSS* testing set. That is because we found that a higher overlap rate is more conducive to separation model training, while a lower overlap rate may lead to local optimum in the training stage.

The overlap region in *mini-session* contains up to 2 speakers. During training, a Gaussian noise with SNR randomly from 0 to 30 dB is added to the pre-generated *mini-sessions* on the fly.

[2][Online]. Available: https://github.com/chenzhuo1011/libri_css

B. Model Configurations

1) General Configurations: We adopt the T-F masking method for speech separation. The size of short-time Fourier transformation (STFT) is 512-point, and the hop length is 256. In our previous works [39], [40], we have compared different window size K in the *segmentation* stage. Considering the balance of performance and window-online latency, we chose $K = 150$ (2.4 seconds) as the default window size of *segmentation*, and the overlap of the adjacent window is 1.2 seconds. The bottleneck fully connected (FC) layers mapping the magnitude spectrum into $N = 256$ dimension. In our experiments, all BLSTM layers contain 512 forward and 512 backward hidden units, and following each BLSTM layer, another linear projection layer maps the output into the bottleneck dimension $N = 256$. The experiments are carried out with ESPNet-SE toolkit [61].

2) Dual-Path Models: We firstly compare DP models on the SIMO structure. We adopt BLSTM-SIMO and Transformer-SIMO as baselines, and compare DP-BLSTM-SIMO and DP-Transformer-SIMO with their baselines, respectively. The first baseline is a 4-layer BLSTM, while the DP-BLSTM contains 2 DP blocks and each DP block consists of 2 BLSTM layers for local and global processing. The Transformer baseline contains 10 encoder layers, the attention dimension is 256 and 4 heads are used for multi-head attention [43]. The feed-forward layers has 1024 units. To keep the number of parameters comparable, the DP-Transformer is made up of 5 DP blocks.

We also compare DP models on the SIMO-SISO structure. In this setup, we have no constraint on the number of parameters. The number of DP blocks in the DP models is same as the number of layers in their corresponding baselines. Each DP block consists of 2 basic layers, the one for local modeling and the other one for global modeling. Thus, the DP models have twice the number of parameters as their baseline models. The BLSTM-SIMO-SISO baseline contains 4 BLSTM layers, 1 in which for SIMO and 3 for SISO processing; The Transformer-SIMO-SISO baseline consists of 12 transformer encoder layers, 3 in which for SIMO and 9 for SISO processing.

3) Memory-Based Models: We use three SIMO-SISO baseline to evaluated the memory based models. The first one is BLSTM-SIMO-SISO, in which each window are processed independently. The second stronger baseline is the DP-BLSTM-SIMO-SISO. This DP baseline can capture the long span information by itself. The last one is the refined DP-Transformer-SIMO-SISO, which is the strongest baseline in this paper.

The overlap ratio predictor is a 4-layer TCN with dilation $[1, 2, 4, 8]$. The number of 1-D convolution kernels in the TCN is 256, and the kernel length is 5. For the memory model, the pre-trained speaker embedding model is a ResNet-based [50],

TABLE I
WER (%) EVALUATION ON LIBRICSS FOR CONTINUOUS SPEECH SEPARATION WITH THE PROPOSED DP MODELS AND THE BASELINES. 0S&0 L: 0% OVERLAP RATIO WITH SHORT&LONG SILENCE

Systems	Model size (M)	MACs of 60s (G)	Overlap ratio in % 0S	0L	10	20	30	40	AVG
Mixture [26]	-	-	15.4	11.5	21.7	27.0	34.3	40.5	26.6
BLSTM [26]	-	-	17.6	16.3	20.9	26.1	32.6	36.1	26.0
BLSTM-SIMO	13.9	101	15.3	13.6	18.6	24.9	30.4	33.9	23.9
+ DP*	13.9	101	16.0	12.1	18.6	24.1	29.1	32.7	23.3
BLSTM-SIMO-SISO	14.0	177	16.4	13.2	18.2	23.1	29.2	32.3	23.1
+ DP	27.7	351	16.4	13.1	18.5	22.7	28.4	31.0	22.7
++ Refined	29.7	209	16.2	12.8	18.3	23.2	28.7	31.7	22.8
Transformer-SIMO	8.2	59	16.0	14.4	19.0	22.6	29.5	33.5	23.5
+ DP*	8.2	59	15.6	14.7	18.8	22.8	29.1	32.3	23.2
++ Refined	10.1	40	14.2	12.3	17.4	22.4	29.1	32.5	22.4
Transformer-SIMO-SISO	13.0	178	15.4	12.3	17.6	22.2	28.2	31.4	22.0
+ DP & Refined	27.6	127	14.6	12.0	17.1	21.8	27.9	30.7	**21.7**

[62] 34-layer *r-vector* net [63], [64]. The *r-vector* net is pre-trained on *Voxceleb* [65] and *LibriSpeech* [58] dataset, and the parameters of the *r-vector* net is frozen during the training. In the end-to-end training, the memory embedding net is a 6 layer TCN with 256 convolution kernels in each layer, and the kernel length is 5. The dilations of the TCN is $[1, 2, 4, 1, 2, 4]$. Mean pooling is used after the last TCN layer, thus for each window, one memory vector will be generated.

C. Results on Dual-Path Models

We first compare the DP models with the baseline CSS framework, and the results are listed in Table I. The partially overlapped *mini-sessions* are firstly processed with the CSS systems, and then perform continuous evaluation [26] for word error rate (WER). As the table shows, our baselines are stronger than that reported in [26]. In the four different setups, all the DP models show better overall performance than their baseline, even if we limit the same number of parameters (DP*). Besides, replacing the BLSTM with the Transformer structure can bring another improvement on the higher overlap ratio segments. All these observations demonstrate that the DP models have stronger modeling capacity for the CSS tasks.

D. Results on SIMO-SISO Design

From the results in Table I, it can be further observed that the SIMO-SISO design brings another improvement compared to the original SIMO model. In the BLSTM-based models, SIMO-SISO outperforms the SIMO model with the same model size. In the Transformer-based models, although we use a little bit more parameters in the SIMO-SISO structure, the performance improvement of SIMO-SISO is large enough to show its effectiveness.

E. Results on Refined DP Models

Moreover, the refined structure on the dual-path models can obtain further WER and computational cost reduction according to the results in Table I. The computational cost is evaluated with multiplier–accumulator (MACs) on a 60 seconds input audio. On DP-BLSTM-SIMO, we did not apply the refined method because it only contains 2 DP blocks. From the last row in Table.I, we find that the computational cost of the DP-transformer with the refined method has been hugely reduced, though it has more parameters than the baseline. This refined convolutional downsampling method is beneficial for the implementation due to the substantial computational cost reduction and better performance.

F. Window-Online Processing for DP Models

The baseline CSS systems process each window independently. They can be applied as window-online systems, with the ideal latency of window length. However, in experiments of Section V-C, bidirectional modeling is used in the *inter-window* processing, and thus the DP systems are offline. Accordingly we carry out the window-online processing experiments on DP BLSTM-SIMO here. To enable the window-online processing capacity, the BLSTM in the *inter-window* processing layer is replaced with the unidirectional LSTM. Four window sizes of $50, 100, 150, 200$, which corresponding to $0.8, 1.6, 2.4, 3.2$ seconds, are compared in the window-online processing experiments. The inference latency of the window-online models is evaluated on an Intel Core i7 CPU. The latency in Table II includes the systems' inherent latency, which is the window size of the CSS system.

The WER and DNSMOS scores are listed in Table II. There are four observations from the Table I) Firstly, for all the window sizes, the DP models shows better performance than the BLSTM-SIMO baseline. The results verify the power of DP method. 2) Secondly, with the reduced window-length, the BLSTM-SIMO baselines become worse on WER and DNSMOS evaluation while the proposed DP models keep the performance at the same level. The reason should be that, the baseline BLSTM can not get enough context information for the separation with a small window, while the proposed DP models can still capture the long span information with the *inter-window* block. 3) Thirdly, the window-online processing on DP model also shows comparable performance with the window-offline DP models, although the online model has only half of the parameters in the *inter-window* processing layer compared to the

TABLE II
WER (%) AND DNSMOS EVALUATION ON LIBRICSS FOR CONTINUOUS SPEECH SEPARATION WITH DIFFERENT LOCAL PROCESSING WINDOW SIZE. THE COMPARISON IS CONDUCTED ON DP-BLSTM-SIMO AND BASELINE BLSTM-SIMO

Window Size	Dual-Path	Window Online	MACs of 60s (G)	Latency (s)	WER (%) w.r.t. overlap ratio						DNSMOS w.r.t. overlap ratio					
					0S	0L	10	20	30	40	0S	0L	10	20	30	40
0.8s	No	Yes	97.1	0.806	16.1	12.7	19.9	25.0	31.8	36.4	3.36	3.37	3.34	3.23	3.22	3.17
	Yes	No	97.1	-	15.0	12.8	18.1	22.9	28.3	31.7	3.37	3.40	3.36	3.25	3.26	3.25
	Yes	Yes	73.1	0.805	14.7	13.2	18.6	24.3	29.3	32.7	3.35	3.37	3.34	3.24	3.24	3.21
1.6s	No	Yes	98.4	1.614	16.2	14.5	20.1	25.1	31.3	34.6	3.36	3.41	3.37	3.28	3.27	3.25
	Yes	No	98.4	-	15.0	12.0	18.4	23.0	28.6	31.6	3.40	3.40	3.39	3.29	3.29	3.27
	Yes	Yes	74.1	1.611	15.8	12.9	18.5	23.6	29.9	32.9	3.40	3.41	3.38	3.27	3.28	3.25
2.4s	No	Yes	100.9	2.426	15.3	13.6	18.6	24.9	30.4	33.9	3.42	3.42	3.38	3.29	3.29	3.26
	Yes	No	100.9	-	16.0	12.1	18.6	24.1	29.1	32.7	3.41	3.43	3.39	3.29	3.30	3.26
	Yes	Yes	76.1	2.417	15.6	12.4	18.4	23.6	29.9	32.8	3.42	3.42	3.38	3.31	3.29	3.28
3.2s	No	Yes	103.5	3.235	15.5	13.4	19.4	24.7	30.7	33.7	3.41	3.42	3.38	3.29	3.28	3.26
	Yes	No	103.5	-	15.2	12.3	18.7	24.3	29.9	33.7	3.41	3.44	3.38	3.30	3.31	3.30
	Yes	Yes	78.0	3.224	15.9	12.7	18.8	23.8	29.9	33.4	3.42	3.43	3.38	3.29	3.28	3.29

TABLE III
WER (%) AND DNSMOS EVALUATION ON LIBRICSS FOR CONTINUOUS SPEECH SEPARATION WITH THE PROPOSED MEMORY-BASED MODEL ON BLSTM-SIMO-SISO BASELINES

Systems	Model Size (M)	MACs of 60s (G)	WER (%) w.r.t. overlap ratio							DNSMOS (%) w.r.t. overlap ratio						
			0S	0L	10	20	30	40	AVG	0S	0L	10	20	30	40	AVG
BLSTM-SIMO-SISO	14.0	177	16.4	13.2	18.2	23.1	29.2	32.3	23.1	3.39	3.39	3.37	3.29	3.29	3.28	3.33
+ SPK-PTRD Mem	21.5	320	15.9	13.7	18.2	22.3	28.7	31.5	22.7	3.42	3.43	3.41	3.34	3.32	3.34	**3.37**
+ SPK-CL Mem	17.1	198	15.5	12.0	17.8	22.5	27.6	30.4	22.0	3.39	3.40	3.38	3.29	3.30	3.30	3.34
+ UN-CL Mem	17.1	198	15.4	12.1	17.2	21.7	27.9	30.5	**21.8**	3.39	3.40	3.37	3.29	3.30	3.30	3.34

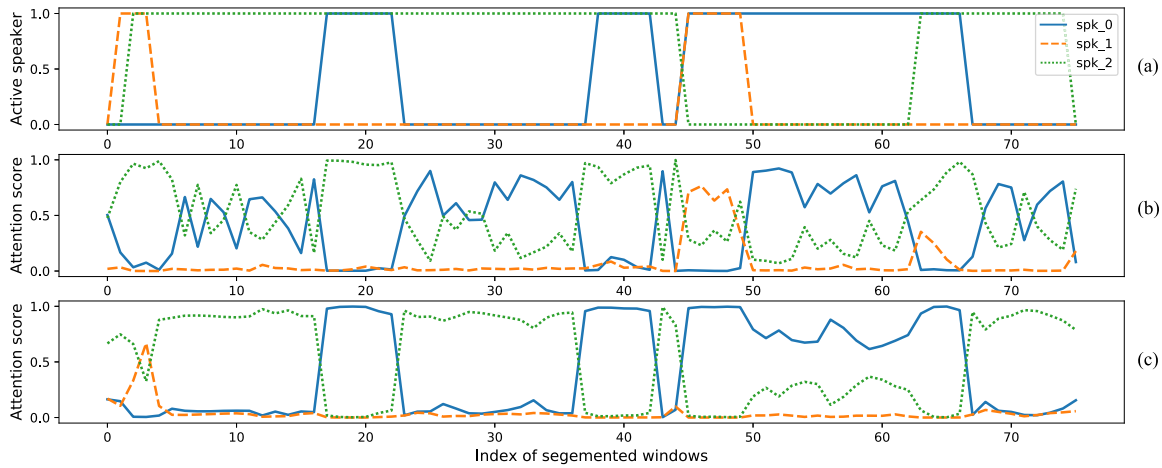


Fig. 4. The attention scores illustration of the proposed speaker-pretrained memory embedding net (SPK-PTRD-Mem). (a) The oracle active speakers in the segmented windows, 1.0 is active and 0.0 is inactive; (b), (c) The attention weights in two SISO branches, and the attention scores computed from the same speaker's active windows has been merged for clarity. The input sample is from the validation set.

offline model. 4) The last interesting observation is that, when the window size (3.2 s) leads to worse WER performance for DP models. One possible explanation is that, bigger window size leads to smaller time steps B in the *inter-window* processing and the resolution for the global modeling also becomes worse. On the contrary, smaller window size leads to more windows, and it has finer resolution for the global modeling.

G. Results on Memory-Based Models

The experiments of memory-based models are firstly performed on the BLSTM-SIMO-SISO mixed backbone. The results are listed in Table III.

1) SPK-PTRD Mem: We firstly compare the pre-trained model strategy, named *"SPK-PTRD Mem"* in this work, with the BLSTM-SIMO-SISO baseline. *"SPK-PTRD Mem"* is a memory-based CSS system, in which the memory embedding net is a pre-trained *r-vector* net [64] for speaker classification. The results in Table III show that *"SPK-PTRD Mem"* can obtain better performance in most cases, and the proposed memory embedding module is useful for CSS task. The large increment of the model size and MACs of *"SPK-PTRD Mem"* mainly comes from the pre-trained *r-vector* net.

Fig. 4 illustrates the attention scores in the *"SPK-PTRD Mem"* model. Fig. 4(a) is the oracle active speaker trace in a simulated

mini-session. Fig. 4(b) and 4(c) are the attention weight scores computed on the two SISO branches. Since the memory embedding net encodes speakers identity information (the *key* and *value* in attention), and the embeddings from the same speaker should be very close, we simply merge the components from the same speakers in the attention weight matrix for a clearer analysis.

Three interesting points can be observed from the figure. First, when processing the overlapped region, the model can attend to the correct memory embeddings that belong to the involved speaker (around index 20, 40, 50, 65). Second, in both SISO branches, the model can track the same speaker at most of the time (e.g. index 5∼45 in Fig. 4(c), but the high-score speaker sometimes swaps between two SISO branches (e.g. around index 20, 40) when there is an overlap. This should be caused by the window-level PIT objective, and the permutation will rearrange to the correct position in the *stitching* stage. Third, in the non-overlapped region where only exist one active speaker (e.g. index 5∼15, 25∼35), one SISO branch can choose the correct memory embedding while the other SISO branch which should output silence, usually chooses random memory embeddings. A better modeling method for silence output may need to be explored in the future.

2) SPK-CL Mem: The second memory model we evaluate is trained with a supervised joint-learning strategy, named *"SPK-CL Mem"*. In *"SPK-CL Mem"*, the speaker label of the simulated training data is utilized for memory embedding net training. As (13) to (15) shows, the loss function is design to be close to the clustering criterion. During the inference stage, a 5-centroid K-means clustering method is performed and the set of clustering centroids are used as the memory pool. The experiments in Table III show that the *"SPK-CL Mem"* method not only outperforms the baseline model, but also shows obviously better performance than the pre-trained model strategy *"SPK-PTRD Mem"*. Compared to *"SPK-PTRD Mem"*, the benefits of *"SPK-CL Mem"* may come from the joint-training with the other modules on the CSS task.

3) UN-CL Mem: The third strategy is unsupervised joint-learning, named *"UN-CL Mem"*. In this setup, the objective function is that described in 18 to 20, which is an unsupervised objective for the memory embedding net. In the inference stage, the memory pool is made up of 5 clustering centroids of the K-means algorithm. The experimental results in Table III show that the unsupervised memory embedding method significantly outperforms the baseline on WER evaluation, and is even better than the other two memory embedding construction strategies. Compared to the speaker-supervised joint-learning *"SPK-CL Mem"*, this fully unsupervised joint-learning *"UN-CL Mem"* can learn more knowledge not only the speaker information, so the other useful knowledge can guide the clustering more accurate and appropriate for continuous speech separation. Moreover, this unsupervised joint-learning strategy is more flexible for the data usage for the model training, and those corpora without labels can also be used. In contrast, the speaker labels are necessary for the supervised joint-learning in *"SPK-CL Mem"*, therefore the unsupervised joint-learning strategy *"UN-CL Mem"* is more practical for real applications.

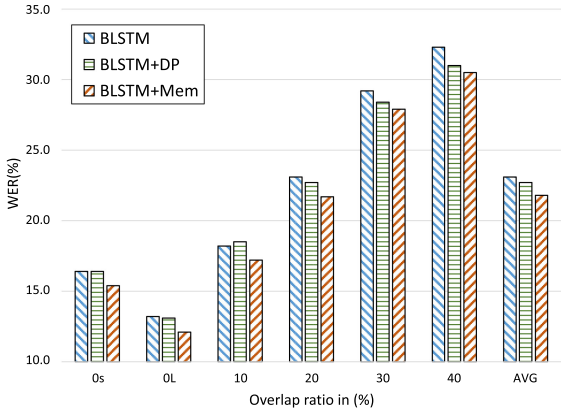


Fig. 5. Comparison between the dual-path model and the memory model. *BLSTM+Mem* is the best *UN-CL Mem* in Tabel III. The score is word error rate (WER), lower is better.

TABLE IV
WER (%) AND DNSMOS EVALUATION ON LIBRICSS FOR CONTINUOUS SPEECH SEPARATION WITH THE PROPOSED DUAL-PATH ARCHITECTURE WITH MEMORY-BASED MODEL

Systems	Model Size (M)	MACs of 60s (G)	WER AVG	DNSMOS AVG
DP-BLSTM-SIMO-SISO	27.7	351	22.7	3.34
+ SPK-PTRD Mem	35.1	494	22.5	**3.35**
+ SPK-CL Mem	30.7	371	**21.9**	3.34
+ UN-CL Mem	30.7	371	22.2	3.34
DP-Trans.-SIMO-SISO	27.6	117	21.7	3.29
+ SPK-PTRD Mem	34.2	254	21.5	**3.32**
+ SPK-CL Mem	28.7	123	21.6	3.29
+ UN-CL Mem	28.7	123	**21.2**	3.28

H. Results on Integrated Dual-Path Architecture With Memory Models

We first make a comparison between the DP model versus the best memory model. The comparison is based on the BLSTM-SIMO-SISO model. From Fig. 5, we can find that the best memory-based model consistently outperform the dual-path model in WER evaluation.

We further combine the proposed dual-path architecture with the new memory-based model to form an integrated framework for the CSS task, and the results are illustrated in Table IV. It is observed that the new memory-based model can also work well on the proposed DP-BLSTM-SIMO-SISO and DP-Transformer-SIMO-SISO systems with the DP architecture, and the memory-based model can obtain additional improvement upon the proposed DP architecture for CSS. We find that the improvement of memory-based method on DP model is relatively smaller than that of BLSTM baseline, it is due to that both the proposed DP architecture and the memory pool model can capture the long span information. To some extent, the effect of one method may have been well expressed by the other one. The system applied with both proposed dual-path architecture and memory-based model achieves the best system performance on the continuous WER evaluation.

TABLE V
WER (%) AND DNSMOS EVALUATION ON LIBRICSS FOR CONTINUOUS SPEECH SEPARATION WITH THE ABLATION STUDY ON MEMORY-BASED MODELS

Systems	Model Size (M)	MACs of 60s (G)	WER AVG	DNSMOS AVG
DP-Trans.-SIMO-SISO	27.6	117	21.7	**3.29**
+ UN-CL Mem	28.7	123	**21.2**	3.28
+ Pseudo Mem	28.7	123	22.7	3.27
+ 1 DP SISO Block	33.9	136	21.5	3.29

I. Ablation Study on Memory-Based Models

The analysis of Fig. 4 in Section V-G1 shows that the attention mechanism could properly control the reading procedure of the memory pool, i.e, choose the most valuable memory embeddings for the local-window processing.

Moreover, we also replace the memory embedding pool with random vectors in both training and evaluation, and keep other modules the same as the memory model. The corresponding result is named as *Pseudo Mem* in Table V. It is observed that without the correct and meaningful memory embedding pool, the memory-based model shows even worse performance. In another ablation study, we simply stack one more SISO DP transformer block in the baseline DP model, which has more parameters and computation cost than the proposed memory-based models. From the last row in Table V, we find that simply stacking more DP blocks is not as effective as the proposed memory-based model. This ablation study shows that the proposed memory-based model with a suitable optimization strategy is useful for continuous speech separation.

VI. CONCLUSION AND DISCUSSION

In this paper, we introduced the dual-path (DP) architecture and the memory-based model for continuous speech separation (CSS). These two kinds of methods can model the long sequences in CSS in two different ways. The DP models utilize the long-span information for local processing in an implicit way, and the memory-based models selectively encode and make use of the memory embeddings from the long recording for local processing. Both of the methods show strong power in the CSS task, and the integration of these two approaches also brings further improvement.

The memory-based framework is a flexible extension to CSS. The memory embedding net can be jointly trained with the CSS task, or various pre-trained networks can also be easily used to introduce external information. In this work, we basically explored the pre-trained speaker identify net on the single-channel data. In a real application, more external knowledge can be used, e.g., spatial information in multi-channel data, visual modality of speakers, or context of the conversation. These kinds of knowledge can be extracted with the corresponding pre-trained networks and selectively accessed in the memory pool, and these will be explored in our future work.

ACKNOWLEDGMENT

Experiments have been carried out on the PI supercomputers at Shanghai Jiao Tong University. Some of the initial experiments were started at JSALT 2020 at JHU, and the authors would like to thank Yi Luo, Cong Han, Tianyan Zhou, Keisuke Kinoshita, Marc Delcroix, and Shinji Watanabe for discussion.

REFERENCES

[1] E. C. Cherry, "Some experiments on the recognition of speech, with one and with two ears," *J. Acoustical Soc. Amer.*, vol. 25, no. 5, pp. 975–979, Sep. 1953.
[2] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. IEEE Int. Conf. Acoust. Speech and Signal Process.*, 2016, pp. 31–35.
[3] 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. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2017, pp. 241–245.
[4] 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 Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 10, pp. 1901–1913, Jul. 2017.
[5] Y. Isik, J. Le Roux, Z. Chen, S. Watanabe, and J. R. Hershey, "Single-channel multi-speaker separation using deep clustering," in *Proc. Int. Speech Commun. Assoc. Interspeech*, pp. 545–549, 2016.
[6] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2017, pp. 246–250.
[7] Y. Luo, Z. Chen, and N. Mesgarani, "Speaker-independent speech separation with deep attractor network," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 4, pp. 787–796, Jan. 2018.
[8] Y. Luo *et al.*, "Deep clustering and conventional networks for music separation: Stronger together," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2017, pp. 61–65.
[9] Z.-Q. Wang, J. Le Roux, and J. R. Hershey, "Alternative objective functions for deep clustering," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 686–690.
[10] Y. Luo and N. Mesgarani, "Tasnet: Time-domain audio separation network for real-time, single-channel speech separation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 696–700.
[11] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing ideal time-frequency magnitude masking for speech separation," *IEEE/ACM Trans. Audio, Speech, and Lang. Process.*, vol. 27, no. 8, pp. 1256–1266, May 2019.
[12] C. Xu, W. Rao, E. S. Chng, and H. Li, "Time-domain speaker extraction network," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2019, pp. 327–334.
[13] P. Wang *et al.*, "Speech separation using speaker inventory," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2019, pp. 230–236.
[14] M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, "Filterbank design for end-to-end speech separation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 6364–6368.
[15] N. Zeghidour and D. Grangier, "Wavesplit: End-to-end speech separation by speaker clustering," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 2840–2849, Jul. 2021.
[16] Y. Luo, Z. Chen, N. Mesgarani, and T. Yoshioka, "End-to-end microphone permutation and number invariant multi-channel speech separation," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 2020, pp. 6394–6398.
[17] J. Chen, Q. Mao, and D. Liu, "Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation," in *Proc. Int. Speech Commun. Assoc. Interspeech*, 2020, pp. 2642–2646.
[18] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 21–25.
[19] D. Yu, X. Chang, and Y. Qian, "Recognizing multi-talker speech with permutation invariant training," in *Proc. Int. Speech Commun. Assoc. Interspeech*, 2017, pp. 2456–2460.
[20] S. Settle *et al.*, "End-to-end multi-speaker speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 4819–4823.
[21] X. Chang *et al.*, "Mimo-speech: End-to-end multi-channel multi-speaker speech recognition," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2019, pp. 237–244.

[22] W. Zhang, X. Chang, Y. Qian, and S. Watanabe, "Improving end-to-end single-channel multi-talker speech recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 1385–1394, Apr. 2020.

[23] T. von Neumann *et al.*, "Multi-talker ASR for an unknown number of sources: Joint training of source counting, separation and ASR," in *Proc. Int. Speech Commun. Assoc. Interspeech*, 2020, pp. 3097–3101.

[24] N. Kanda *et al.*, "Investigation of end-to-end speaker-attributed ASR for continuous multi-talker recordings," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2021, pp. 809–816.

[25] T. Yoshioka, H. Erdogan, Z. Chen, and F. Alleva, "Multi-microphone Neural speech separation for far-field multi-talker speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 5739–5743.

[26] Z. Chen *et al.*, "Continuous speech separation: Dataset and analysis," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 7284–7288.

[27] S. Watanabe *et al.*, "Chime-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings," 2020, pp. 1–7.

[28] S. Chen *et al.*, "Continuous speech separation with conformer," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 5749–5753.

[29] Ö. Çetin and E. Shriberg, "Analysis of overlaps in meetings by dialog factors, hot spots, speakers, and collection site: Insights for automatic speech recognition," in *Proc. 9th Int. Conf. Spoken Lang. Process.*, 2006, pp. 293–296.

[30] K. Kinoshita, L. Drude, M. Delcroix, and T. Nakatani, "Listening to each speaker one by one with recurrent selective hearing networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 5064–5068.

[31] N. Takahashi, S. Parthasaarathy, N. Goswami, and Y. Mitsufuji, "Recursive speech separation for unknown number of speakers," in *Proc. Int. Speech Commun. Assoc. Interspeech*, pp. 1348–1352, 2019.

[32] E. Nachmani, Y. Adi, and L. Wolf, "Voice separation with an unknown number of multiple speakers," in *Proc. Int. Conf. Mach. Learn., Proc. Mach. Learn. Research*, 2020, pp. 7164–7175.

[33] Y. Luo and N. Mesgarani, "Separating varying numbers of sources with auxiliary autoencoding loss," in *Proc. Interspeech 2020, 21st Annu. Conf. Int. Speech Commun. Assoc.*, Virtual Event, Shanghai, China, International Symposium on Computer Architecture, 2020, pp. 2622–2626.

[34] J. Shi *et al.*, "Sequence to multi-sequence learning via conditional chain mapping for mixture signals," in *Proc. Neural Info. Process. Syst.*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020, pp. 3735–3747.

[35] S. Dovrat, E. Nachmani, and L. Wolf, "Many-speakers single channel speech separation with optimal permutation training," in *Proc. Int. Speech Commun. Assoc. Interspeech*, 2021, pp. 3890–3894.

[36] H. Tachibana, "Towards listening to 10 people simultaneously: An efficient permutation invariant training of audio source separation using Sinkhorn's algorithm," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 491–495.

[37] N. Kanda, Y. Gaur, X. Wang, Z. Meng, and T. Yoshioka, "Serialized output training for end-to-end overlapped speech recognition," in *Proc. Int. Speech Commun. Assoc. Interspeech.*, 2020, pp. 2797–2801.

[38] C. Han *et al.*, "Continuous speech separation using speaker inventory for long recording," in *Proc. Int. Speech Commun. Assoc. Interspeech*, 2021, pp. 3036–3040.

[39] C. Li *et al.*, "Dual-path RNN for long recording speech separation," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2021, pp. 865–872.

[40] C. Li *et al.*, "Dual-path modeling for long recording speech separation in meetings," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 5739–5743.

[41] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path RNN: Efficient long sequence modeling for time-domain single-channel speech separation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 46–50.

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

[43] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[44] Y. Luo, Z. Chen, C. Han, C. Li, T. Zhou, and N. Mesgarani, "Rethinking the separation layers in speech separation networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 1–5.

[45] T. Yoshioka, H. Erdogan, Z. Chen, and F. Alleva, "Multi-microphone neural speech separation for far-field multi-talker speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 5739–5743.

[46] Z.-Q. Wang, P. Wang, and D. Wang, "Multi-microphone complex spectral mapping for utterance-wise and continuous speech separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 2001–2014, May 2021.

[47] Y. Hu *et al.*, "DCCRN: Deep complex convolution recurrent network for phase-aware speech enhancement," in *Proc. Int. Speech Commun. Assoc. Interspeech.*, 2020, pp. 2472–2476.

[48] Z.-Q. Wang, P. Wang, and D. Wang, "Complex spectral mapping for single- and multi-channel speech enhancement and robust ASR," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 1778–1787, May 2020.

[49] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2016, pp. 630–645.

[51] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., Cham, Switzerland: Springer, 2015, pp. 234–241.

[52] D. Stoller, S. Ewert, and S. Dixon, "Wave-u-Net: A multi-scale neural network for end-to-end audio source separation," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2018, pp. 334–340.

[53] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, "Temporal Convolutional Networks: A Unified Approach to Action Segmentation," in *Proc Eur. Conf. Comput. Vis. 2016 Workshops*, ser. Lecture Notes in Computer Science, G. Hua and H. Jégou, Eds., Cham, Switzerland, Springer, 2016, pp. 47–54.

[54] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 156–165.

[55] M. Delcroix *et al.*, "Single channel target speaker extraction and recognition with speaker beam," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 5554–5558.

[56] Q. Wang *et al.*, "VoiceFilter: Targeted voice separation by speaker-conditioned spectrogram masking," in *Proc. Interspeech*, 2019, pp. 2728–2732.

[57] K. Žmolíková *et al.*, "Speakerbeam: BUT System Description to VoxCeleb Speaker Recognition Challenge in speech mixtures," *IEEE J. Sel. Top. Signal Process.*, vol. 13, no. 4, pp. 800–814, Oct. 2019.

[58] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2015, pp. 5206–5210.

[59] C. K. A. Reddy, V. Gopal, and R. Cutler, "Dnsmos: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 6493–6497.

[60] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *J. Acoustical Soc. Amer.*, vol. 65, no. 4, pp. 943–950, 1979.

[61] C. Li *et al.*, "ESPNet-SE: End-to-end speech enhancement and separation toolkit designed for ASR integration," in *Proc. IEEE Spoken Lang. Technol.*, 2021, pp. 785–792.

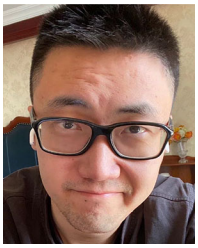
[62] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[63] H. Zeinali, S. Wang, A. Silnova, P. Matějka, and O. Plchot, "BUT system description to VoxCeleb speaker recognition challenge 2019," Oct. 2019, *arXiv:1910.12592*.

[64] S. Wang, Y. Yang, Z. Wu, Y. Qian, and K. Yu, "Data augmentation using deep generative models for embedding based speaker recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 2598–2609, Aug. 2020.

[65] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in *Proc. Int. Speech Commun. Assoc. Interspeech.*, 2017, pp. 2616–2620.

Chenda Li (Graduate Student Member, IEEE) received the B.Eng. degree from the Department of Electronic and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2018, and the M.Sc. degree in 2020 from the Department of Computer Science, Shanghai Jiao Tong University, Shanghai, China, where he is currently working toward the Ph.D. degree with the X-LANCE Lab, Department of Computer Science and Engineering, under the supervision of Yanmin Qian. His current research interests include speech enhancement and speech separation.



Zhuo Chen (Member, IEEE) received the Ph.D. degree from Columbia University, New York, NY, USA, in 2017. He is currently a Principal Applied Data Scientist with Microsoft. He has authored or coauthored more than 80 papers in peer-reviewed journals and conferences and is a reviewer or technical committee member for more than ten journals and conferences. His research interests include automatic conversation recognition, speech separation, diarisation, and speaker information extraction. He actively participated in the academic events and challenges, and won several awards. Meanwhile, he contributed to open-sourced datasets, such as WSJ0-2mix, LibriCSS, and AI-SHELL4, that have been main benchmark datasets for multi-speaker processing research. In 2020, he was the Team Leader in 2020 Jelinek workshop, leading more than 30 researchers and students to push the state of the art in conversation transcription.

Yanmin Qian (Senior Member, IEEE) received the B.S. degree from the Department of Electronic and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2007, and the Ph.D. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2012. Since 2013, he has been with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, where he is currently an Associate Professor. From 2015 to 2016, he was also an Associate Research with Speech Group, Cambridge University Engineering Department, Cambridge, U.K. His current research interests include the acoustic and language modeling in speech recognition, speaker and language recognition, speech enhancement and separation, key word spotting, and multimedia signal processing.