Alleviate Cross-chunk Permutation through Chunk-level Speaker Embedding for Blind Speech Separation

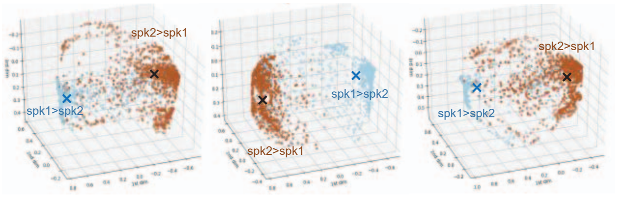
Rongzhi Gu[1], Junyi Peng[1], Yuexian Zou[1,2*], Dong Yu[3]
[1]ADSPLAB, School of ECE, Peking University, Shenzhen, China
[2]Peng Cheng Laboratory, Shenzhen, China
[3]Tencent AI LAB
[*]E-mail: zouyx@pkusz.edu.cn

***Abstract*—Speaker-independent speech separation (SI-SS) refers to recovering speech of unknown speakers from multi-speaker mixtures. The well-known deep clustering (DC) based SI-SS methods cast the speech separation problem into a clustering problem in an embedding space, where time-frequency (T-F) features are encoded as high-dimensional vectors (T-F embeddings). In training stage, the T-F embeddings from the same speaker are trained to be close to each other, otherwise far away. In prediction stage, the T-F embeddings are partitioned into clusters by K-Means, where each cluster corresponds to an unknown speaker from the mixture. To reduce the latency, the T-F embeddings are usually extracted on short speech chunks rather than utterances, which unfortunately leads to a cross-chunk permutation (CCP) problem. In this study, we focus on solving this CCP problem by using the speaker labels as the auxiliary supervision information to train a deep model to map the T-F embeddings of one cluster to one chunk-level speaker embedding (CL-SE). Therefore, in prediction stage, the generated CL-SEs are used to calculate the similarity between each cluster over consecutive chunks. As a result, the speech chunks with the more similar CL-SEs are concatenated to yield the complete utterances. The evaluation is conducted on the well-known WSJ0-2mix and the signal-to-distortion ratio (SDR) is adopted for performance evaluation. Noted that we obtain 41% SDR gain over DC baseline and up to 32% over other speaker-aware methods in open conditions.**

I. INTRODUCTION

Speaker-independent speech separation (SI-SS) or cocktail party problem [1] is defined as segregating and recovering individual and intelligible signal streams from the mixture recordings. Humans are integrated with incredible ability to distinguish speech sources from noisy background and attend to the interested source in realistic acoustic environments. Despite decades of research, machines still fail to have comparable perception capacity.

Recently, there have been two promising deep learning based approaches proposed towards solving SI-SS problem. The first one is the novel permutation invariant training (PIT) criterion proposed by Yu and Kolbæk et al [2] to solve the label permutation problem, which is a primary challenge for supervised SI-SS. This problem is that, the model's multiple outputs may change permutation across frames, resulting in difficulty in assigning correct label for each output. To solve this problem, PIT chooses to minimize the minimum frame-level source estimation error enumerating all output-speaker assignments in training. To eliminate the need for additional frame-level speaker assignment at prediction stage, later, Kolbæk and Yu extended PIT to uPIT [3]. uPIT fixes the output-speaker assignment in the whole utterance and optimizes the source estimation on the utterance-level. However, since the assignment is fixed throughout the utterance, some frames may be aligned to wrong speaker [19]. To alleviate this problem, a constrained uPIT (cuPIT) [4] further introduces speech context information to reinforce the temporal continuity of each separated stream. Preliminary results on well-known dataset WSJ0-mix for SI-SS show that uPIT and its variations achieve the state-of-the-arts at the price of increasing deep model size [2, 3, 4].



(a) chunk 1 (b) chunk 2 (c) chunk 3

Fig. 1 The visual representation of the high-dimensional T-F embeddings of three consecutive chunks (two-speaker mixture case). Each dot represents a 3d projection of a T-F embedding.

The second approach for solving SI-SS problem is called deep clustering (DC) proposed by Hershey et al. [5]. In principle, a bi-directional long short-term memory (BiLSTM) network embeds each time-frequency (T-F) feature of the mixture into a high-dimensional T-F embedding. These T-F embeddings are trained to get close to each other if their associated T-F bins dominated by the same speaker, otherwise far away. The supervision information is given by the ideal symmetric affinity matrix in which the element indicates whether each T-F bin pair belongs to the same speaker. Then, a post-clustering algorithm partitions the T-F embeddings into clusters. Each cluster corresponds to an unknown speaker from the mixture, where T-F bins belong to this cluster are dominated by this speaker. Therefore, binary mask for this speaker can be obtained by setting dominated T-F bins to 1 otherwise to 0. To achieve better clustering of the T-F embeddings in the embedding space, DANet [6] improves the deep clustering method by using a reference cluster centroid (termed as attractor), where each attractor can be regarded as a typical cluster representation. Then, the similarity between attractors and T-F embeddings are calculated to form soft masks. In the end, multiplying the soft mask to the mixture spectrogram yields the separated spectrogram of each source.

For visualization purpose, the T-F embeddings extracted from three consecutive chunks of an utterance (two-speaker mixture case) are projected in 3-D space by principal component analysis (PCA) and are shown in Fig. 1. From Fig.1, we can see that there are two attractors (blue cross for spk1 and black cross for spk2) and they are well separated in each chunk. However, the cross-chunk permutation (CCP) problem can be observed where the attractors (in same color) alter their positions across chunks. As a result, due to the CCP problem, the DC-based SI-SS approaches encounter the difficulty in organizing continuous streams with unidentified chunks during inference.

Researchers have proposed several solutions to relieve CCP problem occurring in the prediction stage. For example, researchers propose to perform clustering over the utterance-level T-F embeddings [5, 7, 8]. In these methods, chunk-level training scheme is adopted while K-means clustering is done over the utterance T-F embeddings. This mismatch leads to the estimation error and degrades the performance [5, 7]. To solve this mismatch problem, the utterance-level training is adopted in [9, 10] which reports the improved results at the price of higher computation complexity and relatively high latency. Also, some analysis show that the methods proposed in [9, 10] can successfully trace the speakers across chunks within an utterance while are not able to trace the speakers across utterances. To avoid the speaker tracing problem, instead of separating speech of all speakers from the mixture, speaker-aware methods [9, 10, 12, 13] propose to only extract target speaker's speech with the aids of target speaker related information. Obviously, these speaker-aware methods ask for the target speaker information in prediction stage, which may not be available. From a different perspective regarding to the speaker tracing problem, Drude [11] proposed to identify each cluster with the speaker identity. In [11], the attractors produced by DANet is further passed to a speaker identification network, which is trained with a cross-entropy classification loss. During prediction, the attractor is utilized to identify clusters across chunks. In principle, in the same T-F embedding space, the model proposed in [11] accomplishes the speech separation task (one vs one) and the speaker classification task (one vs all) at the same time.

In this work, following the DC-based SI-SS framework, we work on resolving the CCP problem and empower speaker tracing capability from a different perspective. Our motivation is to fully exploit all T-F embeddings in each cluster and its corresponding speaker label information in training. Instead of dealing with separation and speaker tracing in the same embedding space as in [11], we propose to train a deep model to map the T-F embeddings of a cluster to a chunk-level speaker embedding (CL-SE) in a speaker embedding space. Therefore, we split the speech separation and speaker classification task into two space, i.e., T-F embedding space and speaker embedding space. Specifically, a DC-based T-F feature encoder used to produce the chunk-level T-F embeddings is trained firstly, which is supervised by the ideal symmetric affinity matrix derived from ideal binary masks. Then a set of T-F embedding pairs is formed in each chunk to train the CL-SE generator (CL-SEG), where the training pairs are in the format of {one cluster of T-F embeddings, its associated speaker label}. As a result, the CL-SEG is trained to map the T-F embeddings to a speaker posterior by minimizing the Kullback-Leibler (K-L) divergence between the speaker label and the estimated speaker posterior. At last, the CL-SEG is cascaded to the T-F feature encoder and the whole system is trained jointly with a multi-task loss. The first loss is for speech separation task and the second loss is a pair-wise speaker verification loss used to train the CL-SEG followed an end-to-end criterion. In the end, the output of the last hidden layer of the CL-SEG is regarded as the chunk-level speaker embedding (denoted as CL-SE in this following context). It is noted that our design concept is similar to that of DC, that is, the CL-SEs are forced to be close to each other if they belong to the same speaker otherwise far away.

At prediction stage, to facilitate conceptual understanding, we take two-speaker case as an example where the chunks of two-speaker mixture are taken as inputs. The T-F features are firstly encoded to the T-F embeddings. Then in the T-F embedding space, two clusters (each for one speaker) are formed by K-means. Secondly, these two clusters are respectively mapped to the CL-SEs, which are representative feature vectors of the two speakers in the speaker embedding space. The generated CL-SEs are used to calculate the similarity between each cluster over consecutive chunks. As a result, the speech chunks with the similar CL-SEs are concatenated to yield the complete utterances. In this study, the similarity is evaluated by the cosine distance. It is worth noting that our approach does not require the identity of speakers in mixture during inference, which differs from other existing speaker-aware methods and is suitable for practical applications.

In summary, our proposed approach has the following advantages: (1) the steady CL-SEs are learned to trace speakers across speech chunks and utterances; (2) CL-SE is a new cue which can be exploited by and combined with other DC-based T-F feature encoders; (3) compared to the utterance-level processing methods [9, 10], our proposed approach works on chunks of 100 to 400 frames (≈0.8 to 3.2s in our setup) which is able to support real-time streaming speech separation and speaker tracing.

The rest of the paper is organized as follows. Section II gives the detailed description of our proposed system. Experimental procedures, results and analysis are presented in Section III. Section IV concludes the paper.

II. Proposed System

In this section, we elaborate on our research ideas and the working principle. For the clarity of the description, two-speaker speech separation task is addressed without loss of the generality. The configuration of our proposed DC-based SI-SS approach in training stage is presented in Fig. 2. Our DC-based SI-SS system mainly consists of three modules. First, the T-F feature encoder (TFE) which encodes the mixture chunk features (MCF) to the T-F embeddings (Sec II.A). Second, the T-F embedding partition module partitions the T-F embeddings into two speaker-aware T-F embedding clusters (S-TFE) using oracle speaker masks (Sec II.B). Third, the chunk-level speaker embedding generator (CL-SEG) produces a chunk-level speaker embedding (CL-SE) for each T-F embedding cluster with the supervision of corresponding speaker label (Sec II.C). Therefore， the specific training process of our proposed DC-based SI-SS has been split into three stages. First, the TFE is trained using supervision information from ideal affinity matrix, which is derived from the oracle speaker masks. Then, the CL-SEG is trained by minimizing the K-L divergence between the speaker label and the estimated speaker posterior. Last, the DC-based SI-SS system is trained jointly with a multi-task loss. The first loss is the separation loss for TFE. The second one is a pair-wise speaker verification loss, where the output from CL-SEG's last hidden layer is extracted as the CL-SE (Sec II.D). In the prediction, the well trained CL-SEs are used to alleviate the CCP problem (Sec II.E).

Considering that there are many abbreviations and symbols involved in this paper, we list them in Table I for ease of reference and understanding.

A. T-F Feature Encoder (TFE)

In virtue of speech T-F sparsity and auditory masking effect [14], speech separation has been carried out in the T-F domain, where each T-F bin is assumed to be dominated by only one speech source [20]. In this sense, the key of the speech separation task is to derive a reasonable partition on T-F bins and build the speaker-aware masks.

The TFE is designed to encode mixture chunk features (MCF) $\boldsymbol{x}$={x_i} of chunk k to the high-dimensional embeddings **V**={$\boldsymbol{v}_i$}, where i is the index of T-F bin. Here, we define the label matrix as **Y**={$\boldsymbol{y}_i$}, where $\boldsymbol{y}_i$ indicates the one-hot vector of the dominant speaker in T-F bin i. For presentation clarity, we omit the chunk index k until discussing CCP problem. The T-F embeddings belong to the same speaker are expected to be closer to each other or farther away. Under supervised learning scheme, a reference label should be given to supervise the similarity between each T-F embedding pair ($\boldsymbol{v}_i$, $\boldsymbol{v}_j$), where the similarity should be close to 1 if $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ belong to the same speaker, otherwise close to 0. This supervision information is obtained by constructing a binary affinity matrix $\mathbf{YY}^{\mathrm{T}}$={<$\mathbf{y}_i$, $\mathbf{y}_j$>} from **Y**, where element <$\mathbf{y}_i$, $\mathbf{y}_j$> denotes the cosine similarity between the speaker one-hot vector pair ($\mathbf{y}_i$, $\mathbf{y}_j$). Therefore, <$\mathbf{y}_i$, $\mathbf{y}_j$> equals to 1 if $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ are the same speaker, otherwise equals to 0. In the proposed methods [5], the partition estimation is sought by performing clustering in the T-F embedding space. The training objective is to minimize the approximation error between the affinity matrix $\mathbf{VV}^{\mathrm{T}}$ and $\mathbf{YY}^{\mathrm{T}}$:

$$L_{TFE}(\mathbf{V},\mathbf{Y}) = \left\|\mathbf{VV}^T - \mathbf{YY}^T\right\|_F^2 = \sum_{i,j}[<\mathbf{v}_i,\mathbf{v}_j> - <\mathbf{y}_i,\mathbf{y}_j>]^2 \quad (1)$$

TABLE I
DESCRIPTION OF ABBREVIATIONS AND SYMBOLS.

Abbr.	Symbol	Description
MCF	$\boldsymbol{x}^{(k)}$={x_i}	Mixture chunk features of chunk k
	$\mathbf{V}^{(k)}$={$\boldsymbol{v}_i$}	T-F embeddings of chunk k
	$\mathbf{Y}^{(k)}$={$\boldsymbol{y}_i$}	label matrix of chunk k
TFE	$f_\varphi(\mathbf{x})$	T-F feature encoder
CL-SEG	$g_\vartheta(\tilde{\mathbf{V}})$	Chunk-level speaker embedding generator
CL-SE	$\mathbf{r}$	Chunk-level speaker embedding
STFE	$\tilde{\mathbf{V}}_c^{(k)}=\{y_{c,i}\boldsymbol{v}_i\}$	A speaker-aware cluster of T-F embeddings that belongs to the c-th speaker in mixture

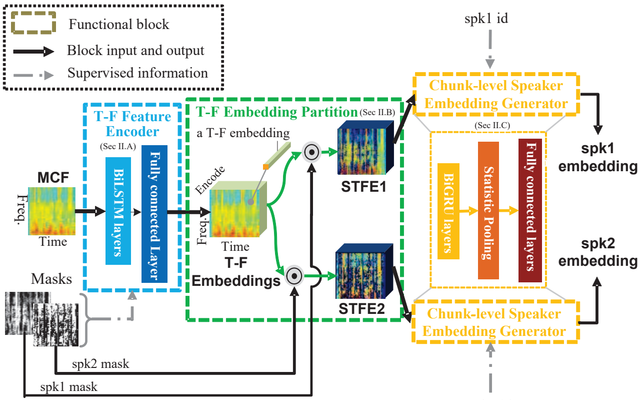


Fig. 2. The configuration of our proposed DC-based SI-SS approach in training stage with two-speaker (spk1 and spk2) mixture. ⊙ denotes element-wise multiplication.

For more efficient training, following [7], we discard some T-F bins with negligible contribution and a weighting matrix **W** is applied to force the network to focus on salient T-F regions leading to a new loss function as:

$$\begin{aligned} L_{TFE,W}(\mathbf{V},\mathbf{Y}) &= \left\|\mathbf{W}^{1/2}\left(\mathbf{VV}^T - \mathbf{YY}^T\right)\mathbf{W}^{1/2}\right\|_F^2 \\ &= \sum_{i,j} w_i w_j [<\mathbf{v}_i,\mathbf{v}_j> - <\mathbf{y}_i,\mathbf{y}_j>]^2 \end{aligned} \quad (2)$$

where w_i is a weighting factor associated to T-F bin i which reflects the importance of $\boldsymbol{v}_i$ to overall speech separation performance. Obviously, different weighting methods can be applied. For example, for the hard weighting method, the low-energy T-F bins are directly omitted as w_{hard_i} = ($sign$(x_i - (max$_j$ x_j - β)) +1) / 2, where β is a threshold parameter an $sign$ is the sign function. For the soft weighting method, the weight is calculated by w_{soft_i} = (x_i - min$_j$ x_j) / (max$_j$ x_j - min$_j$ x_j) which essentially is a min-max scaling. In this study, combining hard and soft weighting approaches, we propose a new weighting function as $w_{semi-soft_i} = w_{hard_i} w_{soft_i}$, where low-energy regions can be completely omitted, and the significant regions can be selected with high probability.

B. T-F Embedding Partition

Subsequently, T-F embeddings are partitioned by the speaker mask $\mathbf{Y}_c$ to derive a speaker-aware cluster of T-F embeddings (STFE) $\tilde{\mathbf{V}}_c \in \mathbb{R}^{TF\times D}$ that belongs to speaker c. The i-th column vector of $\tilde{\mathbf{V}}_c$ is calculated by $\tilde{\mathbf{v}}_{c,i} = y_{c,i}\mathbf{v}_i$ at T-F bin i. During the inference, K-means is performed on T-F embeddings and produce two clusters. Each cluster corresponds to an unknown speaker, where T-F bins belong to this cluster are dominated by this speaker. Therefore, binary mask for this speaker can be obtained by setting dominated T-F bins to 1 otherwise to 0.

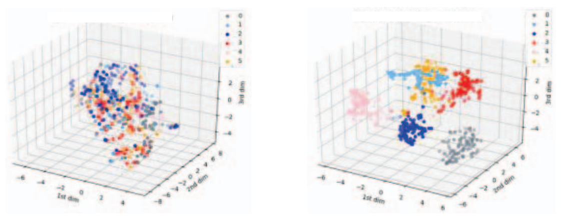
C. Chunk-level Speaker Embedding Generator (CL-SEG)

As discussed above, CCP problem occurs in the inference stage where the cluster of T-F embeddings is not assigned to its corresponding speaker appropriately. To tackle the CCP problem, in this subsection, we propose a new solution. Our main idea is to generate a chunk-level speaker-related representation $\mathbf{r} \in \mathbb{R}^{1\times H}$ for each cluster, which is termed as the chunk-level speaker embedding (CL-SE). Specifically, a deep neural network, termed as CL-SEG is designed and trained on a speaker classification task to generate CL-SEs (spk1 embedding and spk2 embedding in Fig. 2). To properly build the mapping model, CL-SEG is pre-trained by minimizing the Kullback-Leibler (K-L) divergence between $\mathbf{z}$, the one-hot vector of the speaker, and $\hat{\mathbf{z}} \in \mathbb{R}^{1\times E}$, the estimated speaker posterior probability:

$$L_{CL-SEG,KL}(\mathbf{z},\hat{\mathbf{z}},\tilde{\mathbf{V}}) = \sum_{e=1}^{E} p(\mathbf{z}_e|\tilde{\mathbf{V}}) \log \frac{p(\mathbf{z}_e|\tilde{\mathbf{V}})}{p(\hat{\mathbf{z}}_e|\tilde{\mathbf{V}})} \qquad (3)$$

where E is the total number of speakers in the training set. It should be noted that, for a speaker not in the training set, the estimated speaker posterior has no exact meaning and cannot be used as the speaker representation. Therefore, instead of the speaker posterior probability, the output of the last hidden layer is taken as the speaker representation.

To intuitively explain the transformation done by TFE and CL-SEG, Figure 3 visualizes the attractors of chunks in the T-F embedding space and the CL-SEs in the speaker embedding space. It is easy to see that the attractors with same color in (a) mix up with others, which implies that the attractors cannot be utilized as a reliable speaker representation to assign the clusters to speakers. On the other hand, CL-SEs in (b) show much better speaker-discriminative property since dots with same color go closer and dots with different colors go apart. This property indicates similar CL-SEs are more likely to belong to the same speaker. As a result, during the inference, chunks with more similar CL-SEs should be assigned to the same speaker.



(a) cluster centroids (T-SNE) (b) CL-SE (T-SNE)

Fig. 3. Experimental data: 500 two-speaker mixture chunks from 6 speakers (0, 1, 2 are male, 3, 4, 5 are female); Each color indicates a speaker and each dot represents a 3d projection of (a) attractor of a chunk in T-F embedding space; (b) chunk-level speaker embedding in speaker embedding space.

D. Joint Training

In part A and C, TFE and CL-SEG are pre-trained with speech separation and speaker classification task, separately. Motivated by the advances achieved by multi-task learning [21], which uses the correlation between tasks for mutual promotion, we cascade the CL-SEG to TFE as a joint network, as shown in Fig. 2. The motivation for joint training this network is that, when better spectrogram partition can be produced by TFE, there will be less residual information from interference speaker for CL-SEG input. Then, more precise speaker representation can be generated for each cluster. Targeting at enhancing both the spectrogram partitioning and speaker embedding generation, the joint network is fine-tuned with:

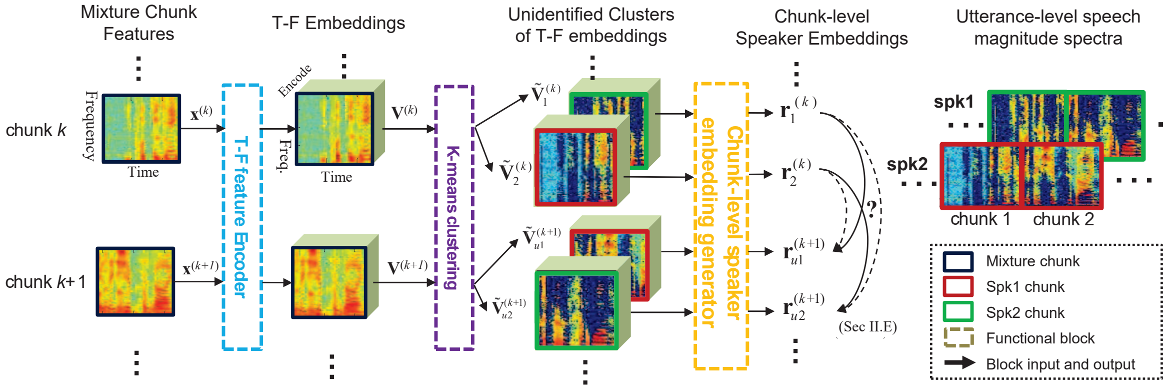


Fig. 4. The illustration of the inference stage in implementing speech separation and cluster assignment (two-speaker mixture case: spk1 and spk2). $x^{(k)}$ and $\mathbf{V}^{(k)}$ respectively denote MCF and T-F embeddings of chunk k. K-means is performed on *encode* dimension of $\mathbf{V}^{(k)}$ to obtain estimated masks, $u1$ and $u2$ denote the cluster index. As in solid line and dotted line, two possible permutations between $\mathbf{r}_u^{(k)}$ and $\mathbf{r}_u^{(k+1)}$ are measured. Then, the best output order is chosen and the separated speech features are then concatenated accordingly.

$$L = L_{TFE,W} + \alpha L_{CL-SEG} \quad (4)$$

where α is a hyper-parameter that controls the relative importance of the two losses, and $L_{CL-SEG}(\mathbf{r},\mathbf{z}) = \left\| \mathbf{r}\mathbf{r}^T - \mathbf{z}\mathbf{z}^T \right\|_F^2$ is defined directly on the CL-SE $\mathbf{r}$ for optimization. The idea of designing loss L_{CL-SEG} lies in that the CL-SE pair ($\mathbf{r}_i$, $\mathbf{r}_j$) belongs to the same speaker should be pushed together, and otherwise pushed away. Also, the speaker pair-wise loss is served for chunk assignment in the prediction stage, which will be discussed in part E.

E. Cross-Chunk Permutation Problem

For DC-based SI-SS task, the CCP comes in the inference stage. Our idea is that chunks belong to the same speaker should have more similarity among their CL-SEs than those belong to different speakers. Fig. 4 depicts our solution to CCP in our proposed system. As described in Fig. 4, a mixture chunk *k+1* followed chunk *k*, is separated into unidentified chunks u_1, u_2. where each separated chunk corresponds to an unknown speaker in the mixture. The CL-SE generated for u_c in chunk *k+1* is denoted as $\mathbf{r}_{u_c}^{(k+1)}$, which can be viewed as a speaker representation to identify this separated chunk. Then, we enumerate all possible permutations of $\{u_1, u_2\}$ to find a best match to speaker $\{1, 2\}$. In this study, the permutation is determined by choosing the minimum cosine distance (maximum cosine similarity) which is given by

$$\pi^{(k+1)} = \arg\min_{\pi \in \mathrm{P}} \sum_{c=1}^{2} \left(1 - < \mathbf{r}_{\pi(c)}^{(k+1)}, \mathbf{r}_c^{(k)} > \right) \quad (5)$$

where *P* represents the set of permutations on $\{u_1, u_2\}$, which is*2*! in this study. The training loss L_{CL-SEG} introduced in Section II.D directly optimizes the cosine similarity between a CL-SE pair ($\mathbf{r}_i$, $\mathbf{r}_j$), which matches the objective in (5). However, (5) suggests to make short-term decision that only compares the similarity between two consecutive chunks *k* and *k+1*. Apart from this decision policy, in experiments, we propose to apply a simple refreshing strategy to accumulate speaker embeddings $\mathbf{r}_c^{(1 \dots k)}$ as long-term memory, which is promising to obtain improved results for longer utterances. The speaker embedding $\mathbf{r}_c^{(1 \dots k)}$ is updated by averaging the new CL-SE and the previously accumulated speaker embedding, i.e., $\mathbf{r}_c^{(1 \dots k)} = \left(\mathbf{r}_c^{(k)} + \mathbf{r}_c^{(1 \dots k-1)} \right) / 2$.

III. Experiments and Analysis

A. Setup

We conducted experiments on well-studied corpus WSJ0 2-mix introduced in [5]. The 30-hour training set and 10-hour development set is generated by randomly mixing two utterances of 101 speakers from folder si_tr_s at signal-to-noise ratios (SNR) between 0-10dB, respectively. The development set is used for hyper-parameter tuning and performance evaluation under closed condition (CC). 5-hour evaluation data is created similarly from 16 unseen speakers in folder si_dt_05 and si_et_05 and used for open condition (OC) evaluation. Following the common practice, all signals are down-sampled to 8kHz considering computation power constraints. The log magnitude of the spectrogram with zero mean normalization is taken as the input feature. STFT is computed with 32ms window length, 8ms hop size and Hann window, leading to feature dimension of 129 per frame.

B. Training Procedure

To fairly compare with other frameworks that incorporate speaker information in training for enhancing speech separation and tracing speakers, such as [9, 10], we adopt the same TFE design and the same training procedure of the baseline methods. The TFE contains two BiLSTM layers with 300 cells each direction and a feedforward layer with tanh activation of 129×*D* nodes, where *D* is the dimension of T-F embedding and is set to 40 to match the best setup for DC [5]. In this study, the TFE is trained from scratch using 100-frame chunks, and then fine-tuned with 400-frame chunks for curriculum learning following the method suggested in [15].

Besides, considering there are 101 speakers in our training set, we set the dimension of the CL-SEG as 128. In the joint training stage, 400 frames per chunk are taken for training the TFE as fine-tuned setup. As a result, a batch of T-F embeddings produced by TFE and fed as input to CL-SEG is with the batch size of 400×129×40. This batch size is too large for mini-batch training. To handling the computation issue, we divide each 400-frame chunk into four 100-frame sub-chunks. Then, the speaker embeddings extracted from four sub-chunks are averaged to form the final speaker embeddings.

All parameters are optimized by Adam [16] with learning rate initialized as 1e-3. The learning rate is halved by every 3 epochs if the validation loss has not decreased. The batch size is taken at 64. Hard weight threshold β is 40dB as that used in [5, 7, 8]. Joint training factor α is found to achieve most balanced results at 0.001.

C. Results and Analysis

The evaluation results are reported by signal-to-distortion (SDR) improvement between estimated separated speech and raw mixture employing BSS_EVAL toolbox [17]. All experiments are implemented in TensorFlow v1.0.

First, to show the negative influence of CCP problem and then demonstrate that our proposed method effectively alleviates it, for experiment in Table II, we compare two types of chunk assignment: optimal assignment and default assignment. Optimal assignment uses oracle information to avoid CCP problem and default assignment represents the practical inference scenario, when there is no clean source to decide the chunk assignment and CCP problem raises.

For comparison, DC baseline [5] is listed along with our proposed methods. In order to make the expression clearer, we make the following definition. TFE-W*hard*, TFE-W*soft*, TFE-W*semi-soft*, and TFE‡-W*semi-soft* respectively indicate our proposed TFE using hard weighting, soft weighting, semi-soft weighting, and semi-soft weighting with curriculum learning without considering CL-SEG model. TFE‡-CL-SEG refers to our final model where semi-soft weighting, curriculum learning and joint TFE and CL-SEG training are applied. All these methods are trained and inferenced in chunks. In the inference, for optimal assignment, K-means method is adopted to cluster the T-F embeddings of each chunk. Then, separated chunks that belong to each source are concatenated into utterances by choosing the minimum error between chunk-level clean source (oracle) and separated source. While for default assignment, all methods, except our TFE‡-CL-SEG, perform K-means on T-F embeddings of the utterance, which is gathered from consecutive chunks in this utterance. In the contrary, our TFE‡-CL-SEG operates separation and clustering on chunks and the chunk assignment is determined with assist of the extracted CL-SEs as discussed in Section II. E.

From Table II, all methods suffer from a relative performance degradation due to CCP problem. For default assignment, we can observe that TFE‡-CL-SEG achieves the best performance under CC and OC among other methods. The performance gain benefits from additional supervision from speaker label information and joint training. Besides, it obtains the least performance degradation when CCP problem raises, which demonstrates the effectivity of our proposed method to alleviate the CCP problem.

We also compare with other approaches by default chunk assignment, including several classic approaches (Oracle NMF, CASA) and speaker-aware approaches (ASAM-spk [9] and Blind Speaker Adaptation (BSA) [10]), The experiments are reported in Table III. Note that ASAM-spk and BSA share the same TFE structure as ours. Moreover, they make use of additional recordings from the speakers to enhance separation performance. Also, both methods are trained and inferenced on utterance-level mixture features, and K-means is performed on the T-F embeddings of the utterance as DC baseline. In Table III, we can see that TFE‡-CL-SEG achieves the best performance among comparison methods. ASAM-spk and BSA are superior to DC baseline. This is easy to understand since additional speaker information is used in training and inference. It is encouraged to see that, compared to ASAM-spk and BSA, where utterance-level input is used, our proposed TFE‡-CL-SEG operates on chunks, which is more promising to support real-time streaming speech separation and speaker tracing.

According to our knowledge, there are several SI-SS methods have achieved competitive performance on WSJ0 2-mix under OC. For example, uPIT-BLSTM-ST [3] gains 10.0 dB with 92.7M parameters, cuPIT-Grid-RD [4] gains 10.2dB with 47.2M parameters, DC++ [8] obtains 10.8dB with 13.6M parameters, chimera++ [7] reaches 11.5 dB with 32.9M parameters. Compared to the above methods, the SDR performance of our proposed TFE-CL-SEG is slightly inferior but our model is of 6.6M parameters which is much less than that of the above methods. The SDR improvement performance gap possibly comes from the network model design and loss function used for TFE. In our work, we only adopt the first DC method proposed in [5]. We argue that our proposed CL-SEG can be applied to more powerful TFE design to achieve better performance with trivial addition of parameters. In our future study, we will work on replace TFE with chimera++, DANet, DC++ to evaluate our proposed CL-SEG.

TABLE II
SDR IMPROVEMENT (DB) OF DC AND TFE TRAINED WITH DIFFERENT SETUPS AND JOINT TRAINING ON WSJ 2-MIX. INPUT FRAME DENOTES THE NUMBER OF FRAMES IN ONE CHUNK RESPECTIVELY FOR TFE AND CL-SEG. ‡DENOTES CURRICULUM LEARNING.

Method	# of	Input	Opt. assign.		Def. assign	
	param.	frames	CC	OC	CC	OC
DC [5]	5.5M	100	6.5	6.5	5.9	5.8
TFE-W$_{\text{hard}}$	5.5M	100/-	8.8	7.1	-	-
TFE-W$_{\text{soft}}$	5.5M	100/-	7.9	6.2	-	-
TFE-W$_{\text{semi-soft}}$	5.5M	100/-	9.0	7.1	-	-
TFE‡-W$_{\text{semi-soft}}$	5.5M	400/-	10.1	7.9	9.6	7.4
TFE‡-CL-SEG	6.6M	400/100	**10.2**	**8.2**	**10.0**	**8.0**

TABLE III
SDR IMPROVEMENT (DB) ON WSJ0 2-MIX.

Method	# of	Input	Def. assign	
	param.	frames	CC	OC
Oracle NMF [5]	-	9	5.1	-
CASA [5]	-	-	2.9	3.1
DC [5]	5.5M	100	5.9	5.8
ASAM-spk [9]	-	All	8.16	6.16
BSA [10]	-	All	6.37	-
TFE‡-CL-SEG	6.6M	400/100	**10.0**	**8.0**

IV. CONCLUSIONS

In this paper, we investigated the CCP problem under the deep-clustering-based SI-SS framework. With the assistance of the speaker label information in training, the mixture feature is casted into two embedding spaces in our proposed method: T-F embedding space for clustering, and speaker embedding space for cluster assignment. Different from other speaker-aware techniques, our approach generates chunk-level speaker embedding that can be easily exploited in the prediction phase to group clusters and trace speakers. Compared to the utterance-level training methods, our approach, operating at the chunk level, can handle the real time streaming speech separation. Experiments on WSJ0 2-mix demonstrate that our proposed approach outperforms the baseline DC methods and other speaker-aware methods under both CC and OC conditions.

ACKNOWLEDGMENT

This paper was partially supported by Shenzhen Science & Technology Fundamental Research Programs (No: JCYJ20170817160058246 & JCYJ20180507182908274). Special acknowledgements are given to AOTO-PKUSZ Joint Research Center for Artificial Intelligence on Scene Cognition & Technology Innovation for its support.

REFERENCES

[1] C. E. Colin, “Some Experiments on the Recognition of Speech, with One and with Two Ears,” *The Journal of The Acoustical Society of America*, vol. 25, no. 6, pp. 975, 1953.

[2] 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 Conf. Acoust., Speech and Signal Process. (ICASSP)*, Mar. 2017, pp. 241-245.

[3] M. Kolbæk, D. Yu, Z.-H. Tan, et al. "Multi-talker Speech Separation with Utterance-level Permutation Invariant Training of Deep Recurrent Neural Networks," *IEEE/ACM Transactions on Audio Speech & Language Processing (TASLP)*, 2017, vol. 99, pp. 1-10.

[4] C. Xu, W. Rao, X. Xiao, et al. "Single Channel Speech Separation with Constrained utterance-level Permutation Invariant Training using Grid LSTM," in *Proc. IEEE Conf. Acoust., Speech and Signal Process. (ICASSP)*, Apr. 2018, pp. 6-10.

[5] J. R. Hershey, Z. Chen, J. Le Roux, et al, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. IEEE Conf. Acoust., Speech and Signal Process. (ICASSP)*, Mar. 2016, pp.31-35.

[6] Z. Chen, Y. Luo and Mesgarani N, "Deep attractor network for single-microphone speaker separation," in *Proc. IEEE Conf. Acoust., Speech and Signal Process. (ICASSP)*, Mar. 2017, pp. 246-250.

[7] Z. Q. Wang, J. L. Roux and J. R. Hershey, "Alternative Objective Functions for Deep Clustering," in *Proc. IEEE Conf. Acoust., Speech and Signal Process. (ICASSP)*, Apr. 2018, pp. 686-690.

[8] Y. Isik, J. R. Hershey, Z. Chen, et al, "Single-channel multi-speaker separation using deep clustering," in *Proc. IEEE Conf. Acoust., Speech and Signal Process. (ICASSP)*, Mar. 2016, pp.31-35.

[9] J. M. Xu, J. Shi, G. C. Liu, et al. "Modeling Attention and Memory for Auditory Selection in a Cocktail Party Environment," in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2018.

[10] J. Zegers, V. H. Hugo, "Improving Source Separation via Multi-Speaker Representations," in *Proc. Interspeech*, 2017, pp. 1919-1923.

[11] L. Drude, T. V. Neumann and R. Haeb-Umbach, "Deep Attractor Networks for Speaker Re-Identification and Blind Source Separation", in *Proc. IEEE Conf. Acoust., Speech and Signal Process. (ICASSP)*, Apr. 2018, pp. 11-15.

[12] K. Zmolikova, M. Delcroix, K. Kinoshita, et al., "Speaker-aware neural network based beamformer for speaker extraction in speech mixtures," in *Proc. Interspeech*, 2017, pp. 2655-2659.

[13] J. Wang, J Chen, D. Su, et al. "Deep Extractor Network for Target Speaker Recovery From Single Channel Speech Mixtures," in *Proc. Interspeech*, Sep. 2018.

[14] S. A. Gelfand, Hearing – An Introduction to Psychological and Physiological Acoustics, Marcel Dekker, New York, 1981.

[15] Y. Bengio, J. Louradour, R. Collobert, et al., "Curriculum learning," in *Proc. of the 26th annual international conference on machine learning (ICML)*, 2009, pp. 41-48.

[16] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv: 1412.6980*, 2014.

[17] E. Vincent, R. Gribonval and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Trans. Audio, Speech, Language Process*, 2006, vol. 14, no. 4, pp. 1462-1469.

[18] Z. Q. Wang, J. L. Roux and J. R. Hershey, "Multi-channel Deep Clustering: Discriminative Spectral and Spatial Embeddings for Speaker-independent Speech Separation," in *Proc. IEEE Conf. Acoust., Speech and Signal Process. (ICASSP)*, 2018, pp. 1-5.

[19] Y. Z. Liu and D. L. Wang, "Divide and Conquer: A Deep CASA Approach to Talker-independent Monaural Speaker Separation", *arxiv: 1904.11148*.

[20] D. L. Wang, J. D. Chen, "Supervised Speech Separation based on Deep Learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (*TASLP*), 2018, 26(10): 1702-1726.

[21] Z. P. Zhang, et al. "Facial landmark detection by deep multi-task learning." *European conference on computer vision* (ECCV), 2014, pp. 94-108.