# Divide and Conquer: A Deep CASA Approach to Talker-Independent Monaural Speaker Separation

Yuzhou Liu, *Student Member, IEEE*, and DeLiang Wang, *Fellow, IEEE*

***Abstract*—We address talker-independent monaural speaker separation from the perspectives of deep learning and computational auditory scene analysis (CASA). Specifically, we decompose the multi-speaker separation task into the stages of simultaneous grouping and sequential grouping. Simultaneous grouping is first performed in each time frame by separating the spectra of different speakers with a permutation-invariantly trained neural network. In the second stage, the frame-level separated spectra are sequentially grouped to different speakers by a clustering network. The proposed deep CASA approach optimizes frame-level separation and speaker tracking in turn, and produces excellent results for both objectives. Experimental results on the benchmark WSJ0-2mix database show that the new approach achieves the state-of-the-art results with a modest model size.**

***Index Terms*—Monaural speech separation, speaker separation, computational auditory scene analysis, deep CASA.**

## I. Introduction

SPEECH usually occurs simultaneously with interference in real acoustic environments. Interference suppression is needed in a wide variety of speech applications, including automatic speech recognition, speaker identification, and hearing aids. One particular kind of interference is the speech signal from competing speakers. Although human listeners excel at attending to a target speaker even without any spatial cues [4], speech separation remains a challenge for machines despite decades of research. In this study, we address monaural (one microphone) speaker separation, mainly in the case of two concurrent speakers, which is also known as co-channel speech separation.

A traditional approach to monaural speech separation is computational auditory scene analysis (CASA) [37], which is inspired by human auditory scene analysis (ASA) mechanisms [3]. CASA addresses speech separation in two main stages: simultaneous grouping and sequential grouping. With an acoustic mixture decomposed into a matrix of time-frequency (T-F) units, simultaneous grouping aggregates T-F units overlapping in time to short segments, each originating from the same source. In sequential grouping, segments are grouped across time into auditory streams, each corresponding to a distinct source. For example, a tandem algorithm [14] utilizes pitch for simultaneous grouping, and temporal continuity of pitch contours for sequential grouping in order to segregate a voiced speech utterance. In [15], an unsupervised speaker separation method first generates T-F segments based on pitch and onset/offset analysis, and then uses constrained clustering to sequentially group T-F segments into speakers. In [26], segments are generated similarly, but sequential grouping is based on noise tracking and a Bayesian technique (Markov Chain Monte Carlo).

Recently deep learning has been employed to address speaker separation. The general idea is to train a deep neural network (DNN) to predict T-F masks or spectra of two speakers in a mixture [7], [17], [44]. There are usually two output layers in such a DNN, one for an individual speaker. These studies assume that the two speakers do not change between training and testing. It has been shown that such talker-dependent training leads to a significant intelligibility improvement for hearing impaired listeners [11]. However, talker-dependent training does not generalize to untrained speakers. Talker-independent speaker separation has to address the permutation problem [12], [22], i.e., how the output layers are tied to the underlying speakers. The details of the permutation problem are introduced in Section II-A.

Frame-level permutation invariant training (denoted by tPIT) [22] tackles this problem by examining all possible label permutations within each frame during training, and uses the one with the lowest frame-level loss to train the separation network. A locally optimized output-speaker pairing can thus be reached, which leads to excellent frame-level separation performance. However, the correct speaker assignment in tPIT's output may swap frequently across frames. In other words, the frame-level optimized outputs cannot be readily streamed into underlying speakers without reorganization. To address this issue, an utterance-level PIT (uPIT) algorithm [22] is proposed to align each speaker to a fixed output layer throughout a whole training utterance. Recent uPIT improvements include new network structure [24], [43] and new training objectives [24]. Conv-TasNet [29] extends uPIT to the waveform domain using a convolutional encoder-decoder structure. FurcaNeXt [34] integrates gated activations and ensemble learning into Conv-TasNet, and reports very high performance.

Deep clustering (DC) [12] looks at the permutation problem from a different perspective. In DC, a recurrent neural network (RNN) with bi-directional long short-term memory (BLSTM) is trained to assign one embedding vector to each T-F unit

Manuscript received April 24, 2019; revised August 2, 2019; accepted September 1, 2019. Date of publication September 12, 2019; date of current version September 30, 2019. This work was supported in part by two National Institute on Deafness and Other Communication Disorders Grants (R01 DC012048 and R01 DC015521) and in part by the Ohio Supercomputer Center. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Tan Lee. *(Corresponding author: Yuzhou Liu.)*

Y. Liu is with the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210-1277 USA (e-mail: liuyuz@cse.ohio-state.edu).

D. Wang is with the Department of Computer Science and Engineering and the Center for Cognitive and Brain Sciences, The Ohio State University, Columbus, OH 43210-1277 USA (e-mail: dwang@cse.ohio-state.edu).

Digital Object Identifier 10.1109/TASLP.2019.2941148

of the mixture spectrogram. The Frobenius norm between the affinity matrix of embedding vectors and the affinity matrix of the ideal speaker assignment (or the ideal binary mask) is used as the training objective. DC avoids the permutation problem due to the permutation-invariant property of affinity matrices. As training unfolds, embedding vectors of T-F units dominated by the same source are drawn closer together, and embeddings of those units dominated by different sources become farther apart. Clustering these embedding vectors using the K-means algorithm assigns each T-F unit to one of the speakers in the mixture, which can be viewed as binary masking for speech separation. In [28], a concept of attractors is introduced to DC to enable ratio masking. Alternative training objectives, together with a chimera network which simultaneously estimates DC embeddings and uPIT outputs, are proposed in [39]. In [40], iterative phase reconstruction is integrated into the chimera network to alleviate phase distortions. In [41], a phase prediction network is further added to [40] to estimate the clean phase of each speaker source.

PIT and DC represent major approaches to talker-independent speaker separation. There are, however, limitations. As indicated in [22], [27], uPIT sacrifices frame-level performance for the sake of utterance-level assignments. The speaker tracking mechanism in uPIT works poorly for same-gender mixtures. On the other hand, DC is better at speaker tracking, but its frame-level separation is not as good as ratio masking used in tPIT.

Inspired by CASA, PIT and DC, we proposed a deep learning based two-stage method in our preliminary study [27] to perform talker-independent speaker separation. The method consists of two stages, a simultaneous grouping stage and a sequential grouping stage. In the first stage, a tPIT-BLSTM is trained to predict the spectra of the two speakers at each frame without speaker assignment. This stage separates spectral components of the two speakers at the same frame, corresponding to simultaneous grouping in CASA. In the sequential grouping stage, frame-level separated spectra and the mixture spectrogram are fed to another BLSTM to predict embedding vectors for the estimated spectra, such that the embedding vectors corresponding to the same speaker are close together, and those corresponding to different speakers are far apart. A constrained K-means algorithm is then employed to group the two spectral estimates at the same frame across time to different speakers. This stage corresponds to sequential grouping in CASA.

In this study, we adopt the same divide-and-conquer strategy but improve its realization in major ways, resulting in what we call a deep CASA approach. In the simultaneous grouping stage, we utilize a UNet [32] convolutional neural network (CNN) with densely-connected layers [16] to improve the performance of frame-level separation. A frequency mapping layer is added to deal with inconsistencies between different frequency bands. To overcome the effects of noisy phase in inverse short-time Fourier transform (STFT), we explore complex STFT objectives and time-domain objectives as the training targets. In the sequential grouping stage, we introduce a new embedding representation and weighted objective function. In addition, we leverage the latest development in temporal convolutional networks (TCNs) [2], [23], [29], [31], and use a TCN for sequential grouping, which greatly improves speaker tracking. A new dropout scheme is proposed for TCNs to overcome the overfitting problem. The evaluation results and comparisons demonstrate the resulting system achieves better frame-level separation and speaker tracking at the same time compared to uPIT and [27].

The rest of the paper is organized as follows. Section II presents details on monaural speaker separation and permutation invariant training. The proposed algorithm, including the simultaneous and sequential grouping stages, is introduced in Section III. Section IV presents experimental results, comparisons and analysis. Conclusion and related issues are discussed in Section V.

## II. Monaural Speaker Separation and Permutation Invariant Training

### A. *Monaural Speaker Separation*

The goal of monaural speaker separation is to estimate $C$ independent speech signals $x_c(n)$, $c = 1, ..., C$, from a single-channel recording of speech mixture $y(n)$, where $y(n) = \sum_{c=1}^{C} x_c(n)$ and $n$ indexes time. In this work, we focus on the co-channel situation where $C = 2$.

Many deep learning based speaker separation systems [7], [17], [44] address this problem in the T-F domain, where STFT is calculated using an analysis window $w(n)$ with FFT length $N$ and frame shift $R$:

$$Y(t,f) = \sum_{n=-\infty}^{\infty} w(n - tR) y(n) e^{-j2\pi fn/N} \tag{1}$$

$$X_c(t,f) = \sum_{n=-\infty}^{\infty} w(n - tR) x_c(n) e^{-j2\pi fn/N} \tag{2}$$

where $t$ and $f$ denote the frame and frequency, respectively. The magnitude STFT of the mixture signal $|Y(t,f)|$, together with other spectral features, are fed into a neural network to predict a T-F mask $M_c(t,f)$ for each speaker $c$. The masks are multiplied by the mixture to estimate the original sources:

$$|\tilde{X}_c(t,f)| = M_c(t,f) \odot |Y(t,f)| \tag{3}$$

Here $\odot$ denotes element-wise multiplication, and $|\tilde{X}_c(t,f)|$ denotes the estimated magnitude STFT of speaker $c$. An estimate of complex STFT $\hat{X}_c(t,f)$ can be obtained by coupling $|\tilde{X}_c(t,f)|$ with noisy phase. In the end, separated waveforms are resynthesized using inverse STFT (iSTFT):

$$\hat{x}_c(n) = \frac{\sum_{t=-\infty}^{\infty} w(n - tR) \frac{1}{N} \sum_{f=0}^{N-1} \hat{X}_c(t,f) e^{j2\pi fn/N}}{\sum_{t=-\infty}^{\infty} w^2(n - tR)} \tag{4}$$

Various training targets of $|\tilde{X}_c(t,f)|$ have been explored for masking based speech separation [38]. Phase-sensitive approximation (PSA) is found to be effective as it accounts for errors introduced by noisy phase [8], [22]. In PSA, the desired reconstructed signal is defined as: $|X_c(t,f)| \odot \cos(\phi_c(t,f))$, where $\phi_c(t,f)$ is the phase difference between $Y(t,f)$ and $X_c(t,f)$. Overall, the training loss at each frame is computed as:

$$\begin{aligned} J_t^{PSA} = \sum_{f=1}^{F} \sum_{c=1}^{2} &||M_c(t,f) \\ &\odot |Y(t,f)| - |X_c(t,f)| \odot \cos(\phi_c(t,f))|| \end{aligned} \tag{5}$$

where $||\cdot||$ denotes the $l_1$ norm.

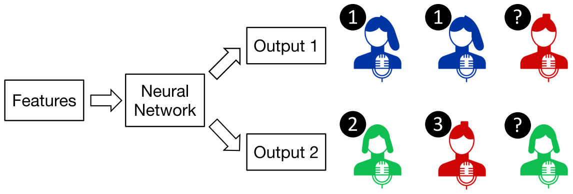

Fig. 1. Illustration of the permutation problem for talker-independent speaker separation.

The above formulation works well only when each output layer is tied to a training target signal with similar characteristics. For instance, we may tie each output to a specific speaker, leading to talker-dependent training. We may also tie two outputs with male and female speakers respectively, leading to gender-dependent training. However, for talker-independent training data, how to select output-speaker pairing becomes a nontrivial problem. Think of a training set consisting of three female speakers, as illustrated in Fig. 1. For the mixture of speakers 1 and 2, we can tie output 1 to speaker 1, and output 2 to speaker 2. For the mixture of speakers 1 and 3, again output 1 can be tied to speaker 1, and output 2 tied to speaker 3. However, it is hard to decide the pairing for the mixture of speakers 2 and 3. If output-speaker pairing is not arranged properly, conflicting gradients may be generated during training, preventing the neural network from converging. This is referred to as the permutation problem [12], [22].

### B. Permutation Invariant Training

Frame-level PIT [22] overcomes the permutation problem by providing target speakers as a set instead of an ordered list, and output-speaker pairing for a given frame $t$, is defined as the pairing that minimizes the loss function over all possible speaker permutations $P$. For tPIT, the frame-level training loss in Eq. (5) is rewritten as:

$$J_t^{tPIT-PSA} = \min_{\theta(t)\in P} \sum_{f,c} ||M_c \odot |Y| - |X_{\theta_c(t)}| \odot \cos(\phi_{\theta_c(t)})|| \tag{6}$$

We omit $(t,f)$ in $M, Y, X$, and $\phi$ for brevity. In (6), $\theta_c(t)$ indexes the speaker paired with output $c$ at frame $t$. $\theta(t)$ includes all $C$ output-speaker pairings at frame $t$, and corresponds to one speaker permutation. The tPIT objective scans all $P$ permutations, and utilizes the permutation with the minimum frame-level loss.

tPIT does a good job in separating two speakers at the frame level [22], [27]. However, due to its locally optimized training objective, an output layer may be tied to different speakers at different frames, and the correct speaker assignment may swap frequently. If we reassign the outputs with respect to the minimum loss for each speaker, tPIT can almost perfectly reconstruct both speakers [27].

Optimal speaker assignments are not obtainable in practice as the targets are not given beforehand. To address this issue, uPIT fixes output-speaker pairing $c \leftrightarrow \theta_c(t)$ for a whole utterance, which corresponds to the pairing that provides the minimum utterance-level loss over all possible permutations.

As reported in [22], [27], uPIT considerably improves the separation performance with a default output assignment. But it has the following shortcomings. First, uPIT's output-speaker pairing is fixed throughout a whole utterance, which prevents frame-level loss to be optimized as in tPIT. As a result, uPIT always underperforms tPIT if their outputs are optimally reassigned. Second, uPIT addresses separation and speaker tracking simultaneously, and due to limited modeling capacity of a neural network, uPIT does not work well for speaker tracking, especially for same-gender mixtures.

## III. Deep CASA Approach to Monaural Speaker Separation

We employ a divide and conquer idea to break down monaural speaker separation into two stages. In the simultaneous grouping stage, a tPIT based neural network separates spectral components of different speakers at the frame-level. The sequential grouping stage then streams frame-level estimates belonging to the same speaker. Unlike uPIT, separation and tracking are optimized in turn in the deep CASA framework. The two stages are detailed in the following subsections.

### A. Simultaneous Grouping Stage

*1) Baseline System:* We adopt the tPIT framework described in [27] as the baseline simultaneous grouping system. The magnitude STFT of the mixture is used as the input. BLSTM is employed as the learning machine. The system is trained using the loss function in Eq. (6). In the end, frame-level spectral estimates are passed to the second stage for sequential grouping.

*2) Alternative Training Targets for tPIT:* As mentioned, the PSA training objective partially accounts for STFT phase, unlike the ideal binary mask (IBM) and ideal ratio mask (IRM). However, PSA cannot completely restore the phase information in clean sources, because it uses noisy phase during iSTFT. Recently, complex ratio masking [42] (cRM) attempts to restore clean phase. The complex ideal ratio mask (cIRM) is defined in the complex STFT domain, with real and imaginary parts. When applied to the complex STFT of the mixture, it perfectly reconstructs clean sources:

$$X_c(t,f) = \mathrm{cIRM}_c(t,f) \otimes Y(t,f) \tag{7}$$

where $\otimes$ denotes point-wise complex multiplication.

We propose complex ratio masking to perform monaural speaker separation. Instead of directly using the cIRM as the training target, we first multiply the complex mixture by the estimated complex mask $\mathrm{cRM}_c$ to perform complex domain reconstruction:

$$\hat{X}_c(t,f) = \mathrm{cRM}_c(t,f) \otimes Y(t,f) \tag{8}$$

The reconstructed sources are then compared with clean sources to form the training objective:

$$\begin{aligned} J_t^{tPIT-CA} = \min_{\theta(t)\in P} \sum_{f,c} \\ \times [\, |\mathrm{Re}(\hat{X}_c - X_{\theta_c(t)})| + |\mathrm{Im}(\hat{X}_c - X_{\theta_c(t)})| \,] \end{aligned} \tag{9}$$

where the $l_1$ norm is applied to both the real and imaginary parts of the loss. We call this training objective complex approximation (CA).

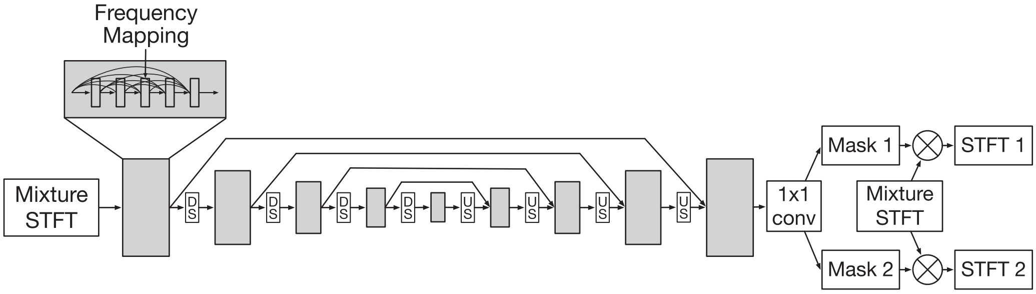

Fig. 2. Diagram of the Dense-UNet used in simultaneous grouping. Gray blocks denote dense CNN layers. DS blocks denote downsampling layers and US blocks denote upsampling layers. Skip connections are added to connect layers at the same level. The inputs, masks and outputs can be defined in either magnitude or complex STFT domain.

We also consider a training objective based on time-domain signal-to-noise ratio (SNR). The proposed framework consists of two steps: First, we organize all frame-level complex estimates $\hat{X}_c$ with respect to the minimum frame-level loss, so that each organized output $\hat{X}_{\theta_c(t)}$ corresponds to a single speaker. The frame-level loss for organization can be defined in three domains: the complex STFT, magnitude STFT and time domain. In each domain, we compare the estimates and ground-truth targets, and calculate the $l_1$ norm of the difference as the loss. We find that the complex STFT loss leads to slightly better separation performance. Second, we apply iSTFT (Eq. 4) to $\hat{X}_{\theta_c(t)}(t,f)$, and compute utterance-level SNR for the final time-domain estimates $\hat{x}_{\theta_c(t)}(n)$:

$$J^{tPIT-SNR} = \sum_{c=1}^{2} 10 \quad \log \frac{\sum_n x_c(n)^2}{\sum_n (x_c(n) - \hat{x}_{\theta_c(t)}(n))^2} \qquad (10)$$

*3) Convolutional Neural Networks for Simultaneous Grouping:* Partly motivated by the recent success of DenseNet [16] and UNet [32] in music source separation [19], [35], we propose a Dense-UNet structure for simultaneous grouping. UNet is a natural choice for spectral-domain source separation. With the "hourglass" architecture and skip connections, UNet models global patterns and preserves fine-grained details in the spectrogram. The use of DenseNet is validated by our preliminary experiments where the proposed Dense-UNet significantly outperforms a standard UNet for speaker separation.

The proposed Dense-UNet is shown in Fig. 2, and it is based on a UNet architecture [32]. It consists of a series of convolutional layers, downsampling layers and upsampling layers. The first half of the network encodes utterance-level STFT feature maps into a higher level of abstraction. Convolutional layers and downsampling layers are alternated in this half, allowing the network to model large T-F contexts. Convolutional layers and upsampling layers are alternated in the second half to project the encoded features back to its original resolution. In this study, we use strided 2 × 2 depthwise convolutional layers [5] as downsampling layers. Strided transpose convolutional layers are used as upsampling layers. Skip connections are added between the layers at the same hierarchical level in the encoder and decoder, and they are important for UNet. As the model goes deeper, the feature maps are projected to more and more abstract representations of the mixture at different resolutions. If skip connections are removed, the network can still produce coarse masks, lacking fine-grain details. Such a phenomenon is discussed in [19].

Next, we replace convolutional layers in the original UNet with densely-connected CNN blocks (DenseNet) [16]. The basic idea of DenseNet is to decompose one convolutional layer with many channels into a sequence of densely connected convolutional layers with fewer channels, where each layer is connected to every other layer in a feedforward fashion:

$$z_l = H_l([z_{l-1}, z_{l-2}, ..., z_0]) \qquad (11)$$

where $z_0$ denotes the input feature map, $z_l$ the output of the $l^{\text{th}}$ layer, [...] concatenation, and $H_l$ the $l^{\text{th}}$ convolutional layer followed by ELU (exponential linear unit) activation [6] and layer normalization [1]. The DenseNet structure has shown excellent performance in image classification [16] and music source separation [35]. In this study, all output layers $z_l$ in a dense block have the same number of channels, denoted by $K$. The total number of layers in each dense block is denoted by $L$. As shown in Fig. 2, we alternate 9 dense blocks with 4 downsampling layers and 4 upsampling layers. After the last dense block, we use a $1 \times 1$ CNN layer to reorganize the feature map, and then output two masks.

In CNNs, convolutional kernels are usually applied across the entire input field. This is reasonable in the case of visual processing, where similar patterns can appear anywhere in the visual field with translation and rotation. However, in the auditory representation of speech, patterns that occur in different frequency bands are usually different. A generic CNN kernel may result in inconsistent outputs at different frequencies. To address this problem, Takahashi and Mitsufuji [35] split the spectral input into several subbands, and train band-dependent CNNs, leading to a substantial rise in model size.

We propose a frequency mapping layer which effectively alleviates this problem with a significant reduction of parameters. The basic idea is to project inconsistent frequency outputs to an organized space using a fully-connected layer. We replace one CNN layer in each dense block with a frequency mapping layer. The input to a frequency mapping layer is a concatenation of CNN layers $z_l^0 = [z_{l-1}, z_{l-2}, ..., z_0] \in \mathbb{R}^{T \times F \times K'}$, where $T$ and $F$ denote time and frequency respectively, $K'$ the number of channels in the input. $z_l^0$ is passed to a $1 \times 1$ convolutional layer, followed by ELU activation and layer normalization, to reduce the number of channels to $K$. The resulting output is denoted by $z_l^1 \in \mathbb{R}^{T \times F \times K}$. We then transpose the $F$ and $K$ dimension of $z_l^1$ to get $z_l^2 \in \mathbb{R}^{T \times K \times F}$. Next, $z_l^2$ is fed to a

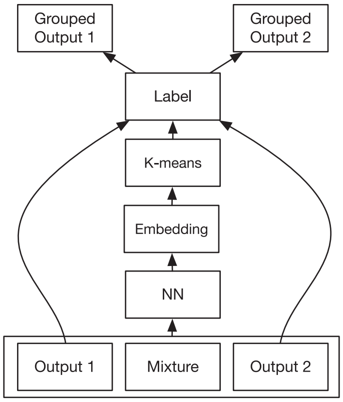

Fig. 3. Diagram of the sequential grouping stage. We use BLSTM or TCN as the neural network in this stage.

$1 \times 1$ convolutional layer, followed by ELU activation and layer normalization, to output $z_l^3 \in \mathbb{R}^{T\times K\times F}$. This layer can also be viewed as a frequency-wise fully connected layer, which takes all frequency estimates as the input and reorganize them in a different space. Finally, $z_l^3$ is transposed back, and the output of the frequency mapping layer $z_l \in \mathbb{R}^{T\times F\times K}$ is generated.

### B. Sequential Grouping Stage

*1) Baseline System:* In this stage, we group frame-level spectral estimates across time using a clustering network, which corresponds to sequential grouping in CASA. In deep clustering based speaker separation, T-F level embedding vectors estimated by BLSTM are clustered into different speakers. We extend this framework to frame-level speaker tracking.

Fig. 3 illustrates our sequential grouping. We first stack the mixture spectrogram and two spectral estimates (including real, imaginary and magnitude STFT) as the input to the system. A neural network then projects frame-level inputs to a $D$-dimensional embedding vector $\mathbf{V}(t) \in \mathbb{R}^D$. The target label is a two-dimensional indicator vector, denoted by $\mathbf{A}(t)$. During the training of tPIT, if the minimum loss is achieved when $\hat{\mathbf{X}}_1(t)$ is paired with speaker 1, and $\hat{\mathbf{X}}_2(t)$ is paired with speaker 2, we set $\mathbf{A}(t)$ to [1 0]. Otherwise, $\mathbf{A}(t)$ is set to [0 1]. In other words, $\mathbf{A}(t)$ indicates the optimal output assignment of each frame. $\mathbf{V}(t)$ and $\mathbf{A}(t)$ can be reshaped into a $T \times D$ matrix $\mathbf{V}$, and a $T \times 2$ matrix $\mathbf{A}$, respectively. A permutation independent objective function [12] is:

$$J^{DC} = ||\mathbf{V}\mathbf{V}^T - \mathbf{A}\mathbf{A}^T||_F^2 \qquad (12)$$

where $||\cdot||_F$ is the Frobenius norm. Optimizing $J^{DC}$ forces $\mathbf{V}(t)$ corresponding to the same optimal assignment to get closer during training, and otherwise to become farther apart.

Because we care more about the speaker assignments of frames where the two outputs are substantially different, a weight $w(t) = \frac{|LD(t)|}{\sum_t |LD(t)|}$ is used during training where $LD(t)$ represents the frame-level loss difference (LD) between the two possible speaker assignments. $LD(t)$ is large if two conditions are both satisfied: 1) the frame-level energy of the mixture is high; 2) the two frame-level outputs, $\hat{X}_1(t,f)$ and $\hat{X}_2(t,f)$, are quite different, so that the losses with respect to different speaker assignments are significantly different. $w(t)$ can be used to construct a diagonal matrix $\mathbf{W} = diag(w(t))$. The final weighted objective function is:

$$J^{DC-W} = ||\mathbf{W}(\mathbf{V}\mathbf{V}^T - \mathbf{A}\mathbf{A}^T)\mathbf{W}||_F^2 \qquad (13)$$

This objective function emphasizes frames where the speaker assignment plays an important role.

During inference, the K-means algorithm is first applied to cluster $\mathbf{V}(t)$ into two groups. We then organize frame-level outputs according to their K-means labels. Finally, iSTFT is employed to convert complex outputs to the time domain.

*2) Temporal Convolutional Networks for Sequential Grouping:* Temporal convolutional networks (TCNs) have been used as a replacement for RNNs, and have shown comparable or better performance in various tasks [2], [23], [29], [31]. In TCNs, a series of dilated convolutional layers are stacked to form a deep network, which enables very long memory. In this study, we adopt a TCN similar to Conv-TasNet [29] for sequential grouping, as illustrated in Fig. 4.

In the proposed TCN, input features are first passed to a 2-D dense CNN block, a $1 \times 1$ convolutional layer and a layer normalization module, to perform frame-level feature preprocessing. The $1 \times 1$ convolutional layer here refers to a 1-D CNN layer with a kernel size of 1. The preprocessed features are then passed to a series of dilated convolutional blocks, with an exponentially increasing dilation factor ($2^0$, $2^1$, ..., $2^{M-1}$) to exploit large temporal contexts. Next, the $M$ stacked dilated convolutional blocks are repeated 3 times to further increase the receptive field. Lastly, the outputs are fed into a $1 \times 1$ convolutional layer for embedding estimation.

In each dilated convolutional block, a bottleneck input with $B$ channels $I_0 \in \mathbb{R}^{T\times B}$ is first passed to a $1 \times 1$ convolutional layer, followed by PReLU (parametric rectified linear unit) activation [10] and layer normalization, to extend the number of channels to $H$, with output denoted by $I_1 \in \mathbb{R}^{T\times H}$. A depthwise dilated convolutional layer [5] with kernel $S \in \mathbb{R}^{3\times H}$, followed by PReLU activation and layer normalization, is then employed to capture the temporal context. The number 3 here indicates the size of the temporal filter in each channel, and there are $H$ depthwise separable filters in the kernel. We adopt non-causal filters to exploit both past and future information, with a dilation factor from $2^0$,... $2^{M-1}$, as in [29]. The output of this part is denoted by $I_2 \in \mathbb{R}^{T\times H}$, which is then passed to a $1 \times 1$ convolutional layer to project the number of channels back to $B$, denoted by $I_3 \in \mathbb{R}^{T\times B}$. In the end, an identity residual connection combines $I_3$ and $I_0$ and forms the final output.

Overfitting is a major concern in sequence models. If not regularized properly, sequence models tend to memorize the patterns in the training data, and get trapped in local minima. To address this issue, various dropout techniques [9], [30], [33] have been proposed for RNNs. Consistent improvements have been achieved if dropout is applied to recurrent connections [30]. Meanwhile, a simple dropout scheme for TCNs is used in [2], i.e., dropping $I_3$ in each dilated convolutional block, but it does not yield satisfactory performance in our experience. Based on these findings, we design a new dropout scheme for the TCN model, denoted by dropDilation. In dropDilation, the dilated connections in depthwise dilated convolutional layers are dropped with a probability of $(1-p)$, where $p$ denotes the keep rate. To be more specific, a binary mask,

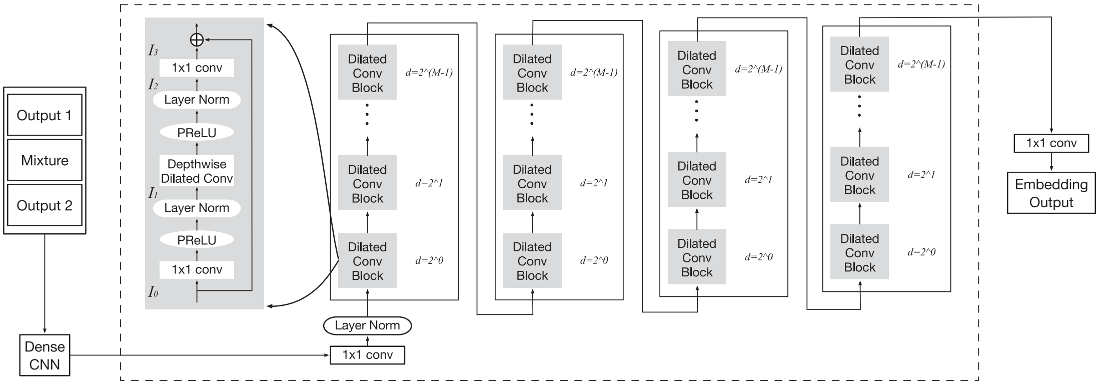

Fig. 4. Diagram of the TCN used in sequential grouping. Outputs from the previous stage are fed into a series of dilated convolutional blocks to predict frame-level embedding vectors. The dilation factor of each block is marked on the right. The detailed structure of a dilated convolutional block is illustrated in the large gray box. The network within the dashed box can be also used for uPIT based speaker separation.

$\mathbf{m} = [m_{-d} \;\; 1 \;\; m_d]^T \in \mathbb{R}^{3\times 1}$, is multiplied with each depthwise dilated convolutional kernel $S \in \mathbb{R}^{3\times H}$ during training, with $m_{-d}$ and $m_d$ drawn independently from a Bernoulli distribution $Bernoulli(p)$. In dropDilation, we only drop the dilated connections while keeping the direct connections to preserve local information.

## IV. Evaluation and Comparison

### A. Experimental Setup

We use the WSJ0-2mix dataset, a monaural two-talker speaker separation dataset introduced in [12], for evaluations. WSJ0-2mix has a 30-hour training set and a 10-hour validation set generated by selecting random speaker pairs in the Wall Street Journal (WSJ0) training set si_tr_s, and mixing them at various SNRs between 0 dB and 5 dB. Evaluation is conducted on the 5-hour open-condition (OC) test set, which is similarly generated using 16 untrained speakers from the WSJ0 development set si_dt_05 and si_et_05. All mixtures are sampled at 8 kHz. STFT with a frame length of 32 ms, a frame shift of 8 ms, and a square root Hanning window is taken for the whole system.

We report results in terms of signal-to-distortion ratio improvement ($\Delta$SDR) [36], perceptual evaluation of speech quality (PESQ) [18], and extended short-time objective intelligibility (ESTOI) [20], to measure source separation performance, speech quality and speech intelligibility, respectively. We also report the final result in terms of scale-invariant signal-to-noise ratio improvement ($\Delta$SI-SNR) [29] for a systematical comparison with other competitive systems.

### B. Models

*1) Simultaneous Grouping Models:* Two models are evaluated for simultaneous grouping: BLSTM and Dense-UNet.

The baseline BLSTM contains 3 BLSTM layers, with $896 \times 2$ units in each layer. In each dense block of Dense-UNet, the number of channels $K$ is set to 64, the total number of dense layers $L$ is set to 5, and all CNN layers have a kernel size of $3 \times 3$ and a stride of $1 \times 1$. The middle layer in each dense block is replaced with a frequency mapping layer. We use valid padding (a term in CNN literature referring to no input padding) for the last CNN layer in each dense block, and same padding (padding the input with zeros so that the output has the same dimension as the original input) for all other layers. The input STFT is zero-padded accordingly.

For both models, when trained with $J_t^{tPIT-PSA}$, the magnitude STFT of the mixture is adopted as the input, and ELU activation is applied to output layers for phase-sensitive mask estimation. If $J_t^{tPIT-CA}$ or $J^{tPIT-SNR}$ is used for training, a stack of real and imaginary STFT is used as the input, and linear output layers are used to predict the real and imaginary parts of complex ratio masks separately.

Both networks are trained with the Adam optimization algorithm [21] and dropout regularization [13]. The initial learning rate is set to 0.0002 for BLSTM, and 0.0001 for Dense-UNet. Learning rate adjustment and early stopping are employed based on the loss on the validation set.

*2) Sequential Grouping Models:* Two models are evaluated for sequential grouping: BLSTM and TCN. Both models are trained on top of a well-tuned simultaneous grouping model.

The baseline BLSTM contains 4 BLSTM layers, with 300×2 units in each layer. In TCN, the maximum dilation factor is set to $2^6 = 64$, to reach a theoretical receptive field of 8.128s. The number of bottleneck units $B$ is selected as 256. The number of units in depthwise dilated convolutional layers $H$ is set to 512. Same padding is employed in all CNN layers. DropDilation with $p = 0.7$ is applied during training.

A 2-D dense CNN block is used in both models for frame-level feature preprocessing, with $K = 16$, $L = 4$, a kernel size of $1 \times 3$ $(T \times F)$ and a stride of $1 \times 1$. The dimensionality of embedding vectors $D$ is set to 40. Both networks are trained with the Adam optimization algorithm, with an initial learning rate of 0.001 for BLSTM, and 0.00025 for TCN. Learning rate adjustment and early stopping are again adopted.

*3) One Stage uPIT Models:* To systematically evaluate the proposed methods, we train a Dense-UNet and a TCN with SNR objectives and uPIT training criterion, i.e., $J^{uPIT-SNR}$. Other details of uPIT Dense-UNet follow those in Section IV-B1, including the number of channels $K$, the number of dense layers $L$, the kernel size, the stride size, frequency mapping layers, padding strategies, the optimization algorithm and the learning rate. The details of uPIT TCN are the same as those in Section IV-B2, including the maximum dilation factor, the number of bottleneck units, the number of units in

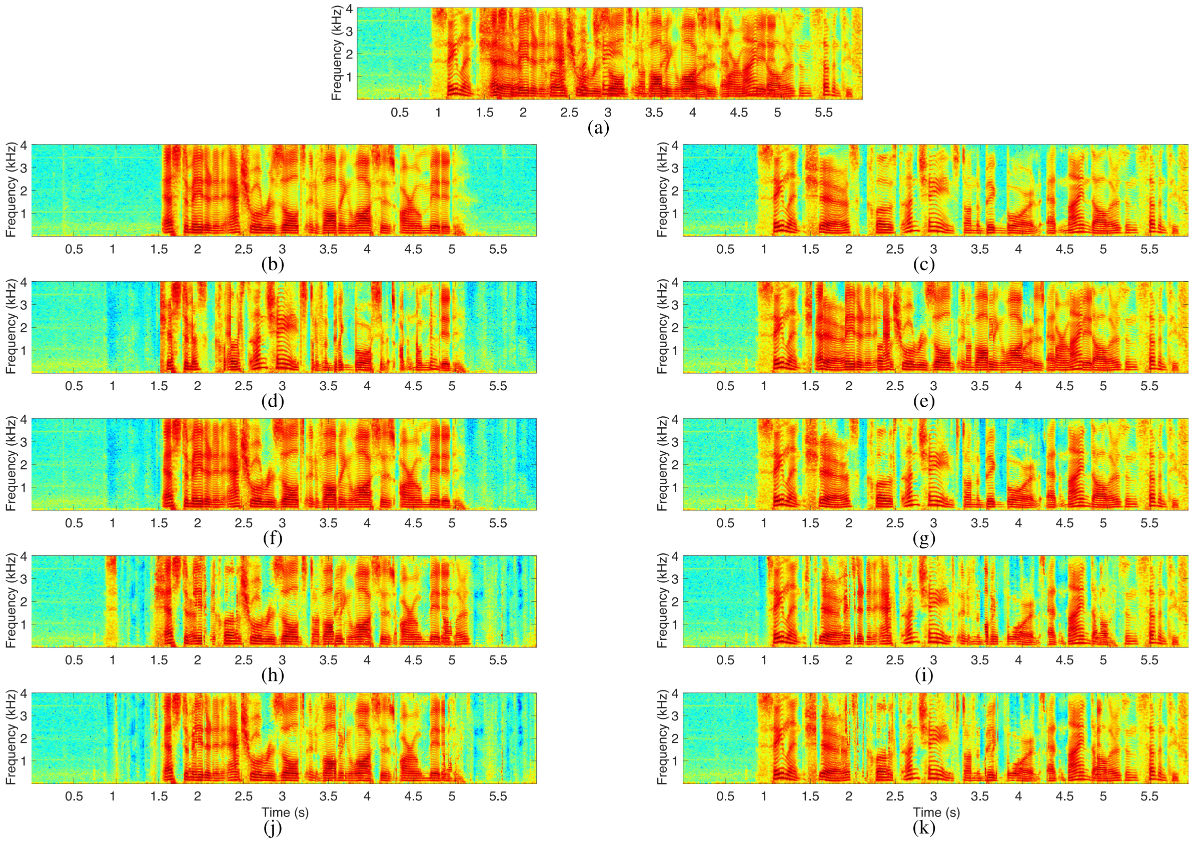

Fig. 5. Speaker separation results of PIT based models in log-scale magnitude STFT. Two models, tPIT Dense-UNet and uPIT Dense-UNet, are trained with CA objectives. The complex outputs from the models are converted to log magnitude STFT for visualization. (a) A male-male test mixture. (b) Speaker 1 in the mixture. (c) Speaker 2 in the mixture. (d) tPIT's output 1 with default assignment. (e) tPIT's output 2 with default assignment. (f) tPIT's output 1 with optimal assignment. (g) tPIT's output 2 with optimal assignment. (h) uPIT's output 1 with default assignment. (i) uPIT's output 2 with default assignment. (j) uPIT's output 1 with optimal assignment. (k) uPIT's output 2 with optimal assignment.

depthwise dilated convolutional layers, the DropDilation rate and the optimization algorithm. An initial learning rate of 0.0001 is used for uPIT TCN.

### C. Results and Comparisons

We first evaluate the simultaneous grouping stage. Table I summarizes the performance of tPIT models with respect to different network structures and training objectives. For all models, outputs are organized with the optimal speaker assignment before evaluation. Scores of mixtures are presented in the first row. Compared to BLSTM, Dense-UNet drastically reduces the number of trainable parameters to 4.7 million, and introduces significant performance gain. The frequency mapping layers in our Dense-UNet introduce a 0.3 dB increment in ΔSDR, 0.1 increment in PESQ, 0.8% increment in ESTOI and a parameter reduction of 0.9 million. Next, we switch from magnitude STFT to complex STFT, and change the training objective to $J_t^{tPIT-CA}$. This change leads to a large improvement, revealing the importance of phase information for source separation. The SNR objective further outperforms the CA objective. We thus adopt tPIT Dense-UNet trained with $J^{tPIT-SNR}$ for simultaneous grouping in the following evaluations.

Table II compares tPIT and uPIT based Dense-UNet in terms of both optimal and default output assignments. Both models are trained with SNR objectives. Thanks to the utterance-level output-speaker pairing, uPIT's default assignment is improved by a large margin over tPIT. However, since frame-level loss is not optimized in uPIT, there is a significant gap between uPIT and tPIT with optimal assignment.

TABLE I
AVERAGE ΔSDR, PESQ AND ESTOI FOR SIMULTANEOUS GROUPING MODELS WITH OPTIMAL OUTPUT ASSIGNMENT ON WSJ0-2MIX OC

| | Objective | # of param. | ΔSDR (dB) | PESQ | ESTOI (%) |
|---|---|---|---|---|---|
| Mixture | - | - | 0.0 | 2.02 | 56.1 |
| tPIT BLSTM | PSA | 46.3M | 13.0 | 3.13 | 86.7 |
| tPIT Dense-UNet | PSA | 4.7M | 14.7 | 3.41 | 90.5 |
| tPIT Dense-UNet | CA | 4.7M | 18.6 | 3.57 | 93.8 |
| tPIT Dense-UNet | SNR | 4.7M | 19.1 | 3.63 | 94.3 |

TABLE II
AVERAGE ΔSDR, PESQ AND ESTOI FOR tPIT AND uPIT BASED DENSE-UNET TRAINED WITH SNR OBJECTIVES

| | Output Assign. | ΔSDR (dB) | PESQ | ESTOI (%) |
|---|---|---|---|---|
| tPIT Dense-UNet | Optimal | 19.1 | 3.63 | 94.3 |
| | Default | 0.0 | 1.99 | 55.8 |
| uPIT Dense-UNet | Optimal | 17.0 | 3.40 | 91.6 |
| | Default | 15.2 | 3.24 | 88.9 |

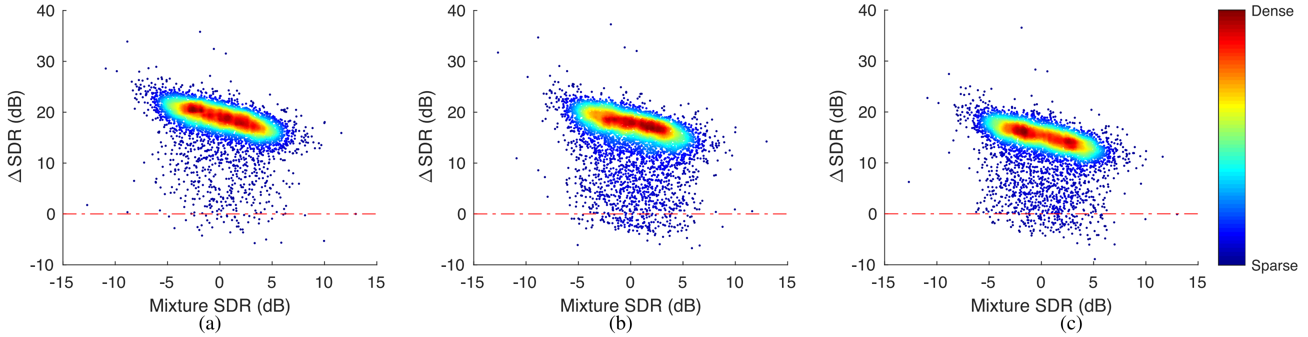

Fig. 6. Scatter-plots of ΔSDR for different methods on WSJ0-2mix OC. (a) Deep CASA (tPIT Dense-UNet + TCN). (b) uPIT Dense-UNet. (c) uPIT TCN.

TABLE III
COMPARISON OF DIFFERENT SEQUENTIAL GROUPING METHODS ON WSJ0-2MIX OC

| Simul. Group. | Seq. Group. | ΔSDR (dB) | PESQ | ESTOI (%) |
|---|---|---|---|---|
| tPIT Dense-UNet | BLSTM | 16.4 | 3.31 | 90.8 |
| tPIT Dense-UNet | TCN | 17.9 | 3.49 | 92.9 |
| uPIT Dense-UNet | - | 15.2 | 3.25 | 89.0 |
| uPIT Dense-UNet | Optimal | 17.0 | 3.40 | 91.6 |
| uPIT TCN | - | 13.5 | 3.06 | 85.9 |
| uPIT TCN | Optimal | 14.9 | 3.19 | 88.1 |

Fig. 5 illustrates the differences between tPIT and uPIT based Dense-UNet in more details. Because SNR objectives lead to less structured outputs in the T-F domain, the models illustrated in the figure are trained with CA objectives. Speaker assignment swaps frequently in the default outputs of tPIT. However, if we organize the outputs with the optimal assignment, the outputs almost perfectly match the clean sources, as shown in the fourth row. On the other hand, the default outputs of uPIT are much closer to the clean sources compared to tPIT. However, for this same-gender mixture, uPIT makes several assignment mistakes in the default outputs, e.g., from 2s to 2.5s, and from 5s to 5.2s. If we optimally organize uPIT's outputs, as in the last row, we can see uPIT exhibits much worse frame-level performance than tPIT. In some frames, e.g., around 4.9s, the predicted frequency patterns are totally mixed up. These observations reveal uPIT's limitations in both frame-level separation and speaker tracking for challenging speaker pairs.

Next, we evaluate different sequential grouping models in Table III. The first two models are trained on top of the tPIT Dense-UNet with the SNR objective. As shown in the table, TCN substantially outperforms BLSTM, both having around 8 million parameters. In BLSTM only neighboring frames are recurrently connected. On the other hand, in TCN, each frame is linked to neighboring and distant frames, facilitating utterance-level speaker tracking. The dropDilation technique in our TCN introduces 0.5 dB ΔSDR gain compared to conventional dropout [2].

In the last four rows of Table III, we report the results of uPIT models. The first uPIT model is trained using Dense-UNet, and it significantly underperforms both deep CASA systems. Even if the outputs are optimally reassigned, uPIT Dense-UNet still systematically underperforms deep CASA (tPIT Dense-UNet + TCN), due to its frame-level separation errors. We also train a TCN model with uPIT objectives, and it yields much worse results than uPIT Dense-UNet.

TABLE IV
FRAME ASSIGNMENT ERRORS FOR DIFFERENT METHODS FOR FRAMES WITH SIGNIFICANT ENERGY (AT LEAST −20 DB RELATIVE TO MAXIMUM FRAME-LEVEL ENERGY)

| Simul. Group. | Seq. Group. | Frame Assign. Errors (%) |
|---|---|---|
| tPIT Dense-UNet | TCN | 1.38 |
| uPIT Dense-UNet | - | 3.43 |
| uPIT TCN | - | 3.07 |

To further analyze the differences between deep CASA and uPIT, we present frame assignment error (FAE) for the best performing deep CASA system and the two uPIT based models in Table IV. FAE is defined as the percentage of incorrectly assigned frames in terms of the minimum frame-level loss. As shown in the table, uPIT Dense-UNet generates the highest FAE, because the network is not specifically designed for sequence modeling. uPIT TCN slightly outperforms uPIT Dense-UNet due to its long receptive field. However, because uPIT TCN does not handle frequency patterns as well, its overall separation performance is worse than uPIT Dense-UNet. Deep CASA cuts FAE by half compared to uPIT models. Such results demonstrate the benefits of the proposed divide-and-conquer strategy, which optimizes frame-level separation and speaker tracking in turn, and achieves better performance in both objectives.

Fig. 6 displays the scatter-plots of ΔSDR for deep CASA, uPIT Dense-UNet and uPIT TCN, where color indicates density. Generally speaking, ΔSDR is higher when mixture SDR is lower. Compared to the two uPIT based models, deep CASA not only improves the average results, but also reduces outlier cases, i.e., test samples with ΔSDR far from the dense central region. Such an observation is also reflected by standard deviations, which are 4.2 dB ΔSDR, 0.35 PESQ, and 5.9% ESTOI for deep CASA, 5.5 dB ΔSDR, 0.49 PESQ, and 9.8% ESTOI for uPIT Dense-UNet, and 4.7 dB ΔSDR, 0.45 PESQ, and 9.3% ESTOI for uPIT TCN.

Table V compares deep CASA and uPIT systems with respect to different gender combinations. Both systems achieve better results on male-female combinations than same gender conditions. The performance gap is larger for female-female mixtures, consistent with the observation in [25]. This might be due to the unbalanced gender distribution in WSJ0-2mix OC, which contains 1086 male-male mixtures, but only 394 female-female mixtures. On the other hand, the performance

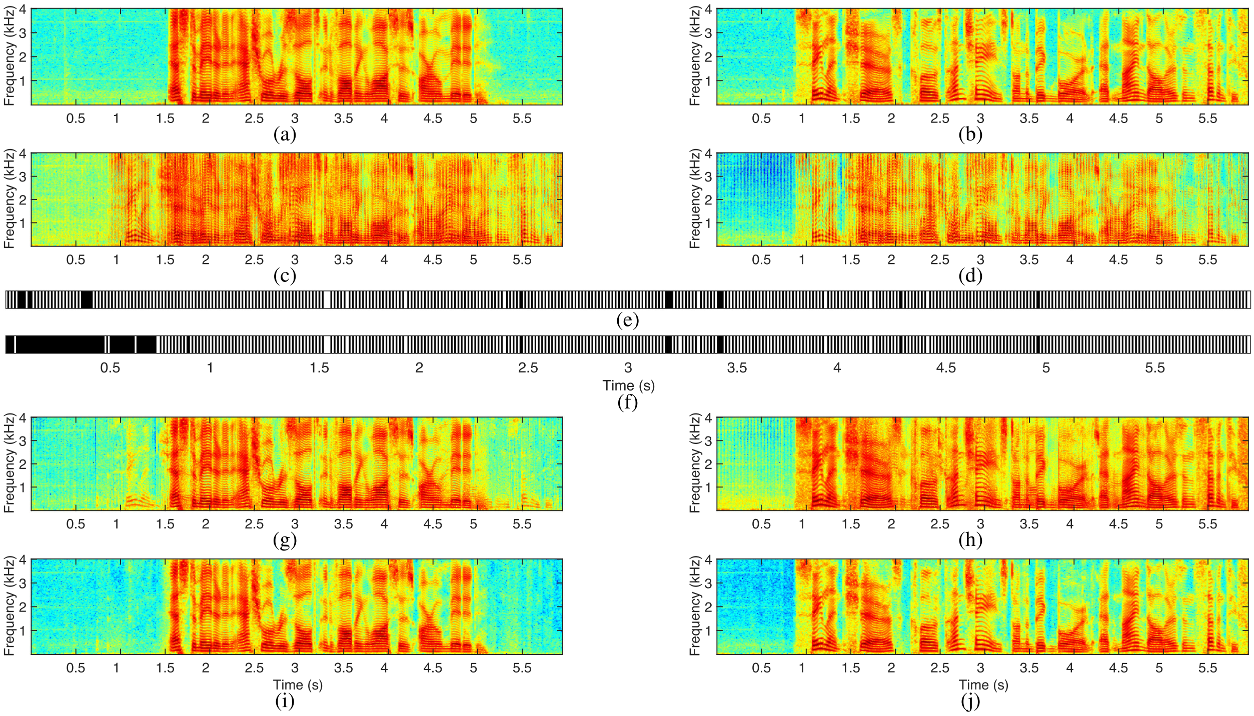

Fig. 7. Speaker separation results of the deep CASA system, with tPIT Dense-UNet trained with SNR objectives for simultaneous grouping and TCN for sequential grouping. The same test mixture is used as in Fig. 5. The complex outputs from the models are converted to log magnitude STFT for visualization. (a). Speaker 1 in the mixture. (b) Speaker 2 in the mixture. (c) tPIT's output 1 with default assignment. (d) tPIT's output 2 with default assignment. (e) Optimal assignment (black and white bars represent two different assignments). (f) K-means assignment. (g) tPIT's output 1 with K-means assignment. (h) tPIT's output 2 with K-means assignment. (i) tPIT's output 1 with K-means assignment after iSTFT and STFT. (j) tPIT's output 2 with K-means assignment after iSTFT and STFT.

TABLE V
AVERAGE ΔSDR, PESQ AND ESTOI FOR DEEP CASA AND uPIT WITH RESPECT TO DIFFERENT GENDER COMBINATIONS

| Model | Gender Comb. | ΔSDR (dB) | PESQ | ESTOI (%) |
|---|---|---|---|---|
| tPIT Dense-UNet + TCN Assign. | Female-Male | 18.9 | 3.57 | 93.9 |
| | Female-Female | 15.7 | 3.32 | 90.5 |
| | Male-Male | 17.2 | 3.45 | 92.5 |
| uPIT Dense-UNet | Female-Male | 17.4 | 3.43 | 92.0 |
| | Female-Female | 11.0 | 2.90 | 83.6 |
| | Male-Male | 13.7 | 3.12 | 86.8 |
| tPIT Dense-UNet + Opt. Assign. | Female-Male | 19.4 | 3.64 | 94.4 |
| | Female-Female | 18.8 | 3.61 | 93.9 |
| | Male-Male | 18.7 | 3.62 | 94.3 |

gap between different gender combinations is much smaller in deep CASA than in uPIT, likely because deep CASA is better at speaker tracking.

Fig. 7 illustrates the results of deep CASA. As shown in the second row, tPIT Dense-UNet trained with SNR objectives generates entirely different default outputs compared to the same model trained with CA (cf. Fig. 5). The optimal assignments alternate almost every frame, leading to striped patterns. To study the phenomenon, we analyze the overall training process of tPIT Dense-UNet trained with $J^{tPIT-SNR}$. At the beginning, the SNR objective leads to similar outputs as the CA objective. However, because there is 75% overlap between neighboring frames in the proposed STFT, models trained with SNR only need to make accurate predictions every other frame, with frames in between left blank. Such patterns start to occur after a few hundred training steps. The competing speaker then gradually fills in the blanks, and the striped patterns are thus formed. As shown in Fig. 7 f, the K-means labels predicted by the sequential grouping system almost perfectly match the optimal labels in speech-dominant frames. However, organizing the default outputs with respect to the K-means labels leads to magnitude STFT that is quite different from the clean sources. Residual patterns from the interfering speaker still exist in some frames. If we convert the complex outputs in Fig. 7 g and 7 h to the time-domain, these residual patterns will be cancelled by the overlap-and-add operation in iSTFT due to their opposite phases. In the last row, we apply iSTFT and STFT in turn to the organized complex outputs, and the new results can almost perfectly match the clean sources.

Simultaneous and sequential grouping are optimized in turn in the above deep CASA systems. We now consider joint optimization, where the two stages are trained together with small learning rates (1/8 of the initial learning rates) for 40 epochs. For the simultaneous grouping module, we organize the outputs using estimated K-means labels, and compare them with the clean sources to form an SNR objective. Meanwhile, the sequential grouping module is trained using the weighted objective in Eq. (13). As joint training unfolds, we observe smoother outputs. Joint optimization introduces slight but consistent improvements in all three metrics (on average by 0.1 dB ΔSDR, 0.02 PESQ, and 0.3% ESTOI, and a reduction of standard deviation by 0.2 dB ΔSDR, 0.02 PESQ, and 0.4% ESTOI).

TABLE VI
NUMBER OF PARAMETERS, AVERAGE ΔSDR, ΔSI-SNR, PESQ AND ESTOI FOR VARIOUS STATE-OF-THE-ART SYSTEMS EVALUATED ON WSJ0-2MIX OC

| | # of param. | ΔSDR (dB) | ΔSI-SNR (dB) | PESQ | ESTOI (%) |
|---|---|---|---|---|---|
| Mixture | - | 0.0 | 0.0 | 2.02 | 56.1 |
| uPIT [22] | 92.7M | 10.0 | - | - | - |
| Conv-TasNet [29] | 5.1M | 15.6 | 15.3 | 3.24 | - |
| Wang et al. [41] | 56.6M | 15.4 | 15.2 | 3.45 | - |
| FurcaNeXt [34] | 51.4M | **18.4** | - | - | - |
| Deep CASA | 12.8M | 18.0 | **17.7** | **3.51** | **93.2** |
| IBM | - | 13.8 | 13.4 | 3.28 | 89.1 |
| IRM | - | 13.0 | 12.7 | 3.68 | 92.9 |
| PSM | - | 16.7 | 16.4 | 3.98 | 96.0 |

TABLE VII
NUMBER OF PARAMETERS, AVERAGE ΔSDR, ΔSI-SNR, PESQ AND ESTOI FOR VARIOUS STATE-OF-THE-ART SYSTEMS EVALUATED ON WSJ0-3MIX OC

| | # of param. | ΔSDR (dB) | ΔSI-SNR (dB) | PESQ | ESTOI (%) |
|---|---|---|---|---|---|
| Mixture | - | 0.0 | 0.0 | 1.66 | 38.5 |
| uPIT [22] | 92.7M | 7.7 | - | - | - |
| Conv-TasNet [29] | 5.1M | 13.1 | 12.7 | 2.61 | - |
| Wang et al. [41] | 56.6M | 12.5 | 12.1 | **2.77** | - |
| Deep CASA | 12.8M | **14.6** | **14.3** | **2.77** | **80.8** |
| IBM | - | 13.6 | 13.3 | 2.86 | 82.1 |
| IRM | - | 13.0 | 12.6 | 3.44 | 88.6 |
| PSM | - | 16.8 | 16.4 | 3.80 | 93.7 |

Table VI compares the deep CASA system with joint optimization and other state-of-the-art talker-independent methods on WSJ0-2mix OC. For all methods, we list the best reported results, and leave unreported fields blank. The numbers of parameters in different methods are estimated according to the papers. The uPIT system [22] is the basis of this study. Conv-TasNet [29] extends uPIT to the waveform domain, where a TCN is utilized for separation. We have also trained a similar uPIT TCN in this work. However, due to the different domains of signal representation, our uPIT TCN yields slightly worse results than Conv-TasNet, which suggests that better performance may be achieved by extending the deep CASA framework to the time domain. In [41], a phase prediction network is trained on top of a DC network. It yields high PESQ. FurcaNeXt [34] produces very high ΔSDR. The deep CASA system generates slightly lower ΔSDR results, but has much fewer parameters. In addition, deep CASA yields the best results in terms of ΔSI-SNR, PESQ and ESTOI. The last three rows present the results of the IBM, IRM and ideal phase-sensitive mask (PSM) with the STFT configuration in Section IV-A. Deep CASA systematically outperforms the ideal masks in terms of ΔSDR and ΔSI-SNR. However, there is still room for improvement in terms of PESQ.

Although the deep CASA algorithm is formulated for two speakers, it can be extended to three or more speakers. First, additional output layers need to be added in the simultaneous grouping stage. In sequential grouping, we employ the setup in [27] to predict one embedding vector for each frame-level spectral estimate. A constrained K-means algorithm is then used to assign each frame-level embedding to a different speaker. Table VII compares such a three-speaker deep CASA system with other approaches on the WSJ0-3mix dataset [12]. Deep CASA significantly outperforms other systems in terms of ΔSDR and ΔSI-SNR, and matches the PESQ score of [41].

## V. CONCLUDING REMARKS

We have proposed a deep CASA approach to talker-independent monaural speaker separation. Simultaneous grouping is first conducted to separate two speakers at the frame level. Sequential grouping is then employed to stream separated frame-level spectra into two sources. The deep CASA algorithm optimizes frame-level separation and speaker tracking in turn in the two-stage framework, leading to much better performance than DC and PIT. Our contributions also include novel techniques such as complex ratio masking, SNR objectives, Dense-UNet with frequency mapping layers and TCN with dropDilation. Experimental results on the benchmark WSJ0-2mix dataset show that the proposed algorithm produces the state-of-the-art results, with a modest model size.

A major difference between our sequential grouping stage and deep clustering is that embedding operates at the T-F unit level in DC, and at the frame level in deep CASA. There are several advantages to our approach. First, DC excels at speaker tracking due to clustering, but it is not better than ratio masking for frame-level separation. Therefore, divide and conquer is a natural choice. Second, deep CASA is more flexible. Almost all DC based algorithms are built on time-frequency processing. Our sequential grouping works on frame-level outputs, which can be produced by estimating magnitude STFT, complex masks, or even time-domain signals. In addition, we reduce the computational complexity of clustering from $O(FT)$ in DC to $O(T)$ in deep CASA.

## References

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

[2] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” 2018, *arXiv:1803.01271*.

[3] A. S. Bregman, *Auditory Scene Analysis*. Cambridge, MA, USA: MIT Press, 1990.

[4] D. S. Brungart, “Informational and energetic masking effects in the perception of two simultaneous talkers,” *J. Acoust. Soc. Amer.*, vol. 109, pp. 1101–1109, 2001.

[5] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 1251–1258.

[6] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (ELUs),” 2015, arXiv:1511.07289.

[7] J. Du, Y. Tu, Y. Xu, L. R. Dai, and C. H. Lee, “Speech separation of a target speaker based on deep neural networks,” in *Proc. Int. Conf. Signal Process.*, 2014, pp. 473–477.

[8] H. Erdogan, J. R. Hershey, and S. Watanabe, “Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 708–712.

[9] Y. Gal and Z. Ghahramani, “A theoretically grounded application of dropout in recurrent neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1019–1027.

[10] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification,” in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 1026–1034.

[11] E. W. Healy, M. Delfarah, J. L. Vasko, B. L. Carter, and D. L. Wang, “An algorithm to increase intelligibility for hearing impaired listeners in the presence of a competing talker,” *J. Acoust. Soc. Amer.*, vol. 141, pp. 4230–4239, 2017.

[12] 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, Signal Process.*, 2016, pp. 31–35.

[13] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” 2012, *arXiv:1207.0580*.

[14] G. Hu and D. L. Wang, “A tandem algorithm for pitch estimation and voiced speech segregation,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 8, pp. 2067–2079, Nov. 2010.

[15] K. Hu and D. L. Wang, “An unsupervised approach to cochannel speech separation,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 1, pp. 122–131, Jan. 2013.

[16] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 4700–4708.

[17] P. S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, “Deep learning for monaural speech separation,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 1562–1566.

[18] International Telecommunication Union Radiocommunication, “Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs,” Recommendation P.862, 2001.

[19] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde, “Singing voice separation with deep U-Net convolutional networks,” in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2017, pp. 745–751.

[20] J. Jensen and C. H. Taal, “An algorithm for predicting the intelligibility of speech masked by modulated noise maskers,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 11, pp. 2009–2022, Nov. 2016.

[21] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980.

[22] 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, Oct. 2017.

[23] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, “Temporal convolutional networks: A unified approach to action segmentation,” in *Proc. Eur. Conf. Comput. Vision*, 2016, pp. 47–54.

[24] C. Li, L. Zhu, S. Xu, P. Gao, and B. Xu, “CBLDNN-based speaker-independent speech separation via generative adversarial training,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 711–715.

[25] Z.-X. Li, Y. Song, L.-R. Dai, and I. McLoughlin, “Listening and grouping: An online autoregressive approach for monaural speech separation,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 4, pp. 692–703, Apr. 2019.

[26] S. Liang, W. Liu, and W. Jiang, “A new Bayesian method incorporating with local correlation for IBM estimation,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 3, pp. 476–487, Mar. 2013.

[27] Y. Liu and D. L. Wang, “A CASA approach to deep learning based speaker-independent co-channel speech separation,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 5399–5403.

[28] 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, Apr. 2018.

[29] Y. Luo and N. Mesgarani, “Conv-TasNet: Surpassing ideal time-frequency magnitude masking for speech separation,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 8, pp. 1256–1266, Aug. 2019.

[30] S. Merity, N. S. Keskar, and R. Socher, “Regularizing and optimizing LSTM language models,” in *Proc. Int. Conf. Learn. Represent.*, 2018.

[31] A. Pandey and D. L. Wang, “A new framework for CNN-based speech enhancement in the time domain,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 7, pp. 1179–1188, Jul. 2019.

[32] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Proc. Med. Image Comput. Comput.-Assisted Intervention*, 2015, pp. 234–241.

[33] S. Semeniuta, A. Severyn, and E. Barth, “Recurrent dropout without memory loss,” in *Proc. Int. Conf. Comp. Ling.*, 2016, pp. 1757–1766.

[34] Z. Shi, H. Lin, L. Liu, R. Liu, and J. Han, “FurcaNeXt: End-to-end monaural speech separation with dynamic gated dilated temporal convolutional networks,” 2019, *arXiv:1902.04891*.

[35] N. Takahashi and Y. Mitsufuji, “Multi-scale multi-band DenseNets for audio source separation,” in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2017, pp. 21–25.

[36] E. Vincent, R. Gribonval, and C. Févotte, “Performance measurement in blind audio source separation,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 4, pp. 1462–1469, Jul. 2006.

[37] D. L. Wang and G. Brown, Eds., *Computational Auditory Scene Analysis: Principles, Algorithms and Applications*. Hoboken, NJ, USA: Wiley-IEEE Press, 2006.

[38] Y. Wang, A. Narayanan, and D. L. Wang, “On training targets for supervised speech separation,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 12, pp. 1849–1858, Dec. 2014.

[39] 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.

[40] Z.-Q. Wang, J. Le Roux, D. L. Wang, and J. R. Hershey, “End-to-end speech separation with unfolded iterative phase reconstruction,” in *Proc. Interspeech*, 2018, pp. 2708–2712.

[41] Z.-Q. Wang, K. Tan, and D. L. Wang, “Deep learning based phase reconstruction for speaker separation: A trigonometric perspective,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 71–75.

[42] D. S. Williamson, Y. Wang, and D. L. Wang, “Complex ratio masking for monaural speech separation,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 3, pp. 483–492, Mar. 2016.

[43] C. Xu, W. Rao, X. Xiao, E. S. Chng, and H. Li, “Single channel speech separation with constrained utterance level permutation invariant training using grid LSTM,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 6–10.

[44] X. L. Zhang and D. L. Wang, “A deep ensemble learning method for monaural speech separation,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 5, pp. 967–977, May 2016.

Authors’ photographs and biographies not available at the time of publication.