Causal Deep CASA for Monaural Talker-Independent Speaker Separation

Yuzhou Liu and DeLiang Wang, *Fellow, IEEE*

***Abstract*—Talker-independent monaural speaker separation aims to separate concurrent speakers from a single-microphone recording. Inspired by human auditory scene analysis (ASA) mechanisms, a two-stage deep CASA approach has been proposed recently to address this problem, which achieves state-of-the-art results in separating mixtures of two or three speakers. A main limitation of deep CASA is that it is a non-causal system, while many speech processing applications, e.g., telecommunication and hearing prosthesis, require causal processing. In this study, we propose a causal version of deep CASA to address this limitation. First, we modify temporal connections, normalization and clustering algorithms in deep CASA so that no future information is used throughout the deep network. We then train a C-speaker ($C \geq 2$) deep CASA system in a speaker-number-independent fashion, generalizable to speech mixtures with up to C speakers without the prior knowledge about the speaker number. Experimental results show that causal deep CASA achieves excellent speaker separation performance with known or unknown speaker numbers.**

***Index Terms*—Monaural speaker separation, talker-independent speaker separation, deep CASA, causal processing.**

I. Introduction

INTERFERENCE from competing speakers is considered a major challenge in speech communication, automatic speech processing systems, and hearing prosthesis. Based on deep learning, many talker-independent monaural speaker separation algorithms have been proposed in recent years to address this problem. Two main approaches are deep clustering (DC) [6] and permutation invariant training (PIT) [13]. Deep clustering learns an embedding vector for each time-frequency (T-F) unit of the mixture. Clustering the embedding vectors results in binary T-F masks, which can be used to separate the speakers from the mixture. In PIT, each output layer in a deep neural network (DNN) is associated with one speaker in the mixture. During training, PIT examines the losses with respect to all possible output-speaker permutations, and optimizes the DNN using the minimum loss. Based on the types of output-speaker pairing, PIT can be categorized into frame-level PIT (tPIT), where the pairings can change frame by frame, and utterance-level PIT (uPIT), where the pairing is fixed throughout each training utterance. Many extensions have been proposed recently for DC and PIT, including [15], [19], [20], [22], [27], [28]. Inspired by research in computational auditory scene analysis (CASA) [25], we recently proposed deep CASA [18] which breaks down the speaker separation task into two stages, i.e., tPIT based simultaneous grouping and clustering based sequential grouping. Deep CASA achieves frame-level separation and speaker tracking in turn. Compared to one-stage PIT or DC which optimizes the two objectives at the same time, deep CASA substantially mitigates the mistakes in speaker tracking, and leads to improvements in speaker separation performance.

Although deep CASA produces the state-of-the-art speaker separation results, it has a major limitation from the viewpoint of real-world deployment: it is non-causal. Causal processing is a major requirement in many real-time speech applications, including telecommunication and hearing aids. For example, mobile communication involves real-time interaction and is sensitive to processing delay. For hearing prosthesis, a processing delay longer than 10 ms would create a misalignment between real and processed signals, hampering speech perception [5]. The deep CASA system [18] utilizes future information as long as 9 seconds for separation and speaker tracking, making it unsuitable for these applications. It is therefore important to develop a causal version of deep CASA.

Models based on uPIT can be easily extended to the causal version if causal DNNs are utilized [13], [20]. However, lacking future information for speaker tracking [1], causal uPIT significantly underperforms non-causal uPIT in a variety of settings, as demonstrated in [1], [13], [20]. On the other hand, even with causal DNNs, clustering based methods like DC [6] and deep attractor networks (DAN) [19] struggle to operate causally, as the centroids of clusters are hard to estimate in an online fashion. Recently, researchers start to incorporate uPIT as a parallel training target for DC based systems, and use the spectral outputs from uPIT during inference [1], [26]–[28]. In this way, speaker separation can be achieved causally without a clustering step [1].

Another challenge for real-world deployment is that the number of concurrent speakers is usually unknown beforehand. Again, DC and DAN fail to operate properly in such a scenario, as the speaker number is needed for clustering. To tackle this problem, Higuchi *et al.* [7] perform offline source counting by computing the rank of the covariance matrix of the embedding

Manuscript received October 26, 2019; revised April 19, 2020 and June 14, 2020; accepted June 18, 2020. Date of publication July 8, 2020; date of current version July 28, 2020. This work was supported in part by two NIDCD under 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 Dr. Huseyin Hacihabiboglu. *(Corresponding author: Yuzhou Liu.)*

Yuzhou Liu is with the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210-1277 USA (e-mail: liu.2376@osu.edu).

DeLiang 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.2020.3007779

vectors. An accuracy of 67.3% is achieved for counting two- and three-speaker mixtures. The non-causal setup and mediocre performance in [7] make the study far from practical utility. On the other hand, a C-output uPIT model can be directly applied to speech mixtures with up to C speakers, without the prior knowledge about the speaker number, as some of the outputs can be trained to generate silence as a placeholder [13]. Another direction for speaker-number-independent separation is to recursively remove one speaker at a time from the mixture [12], [23]. In [23], a one-and-rest permutation invariant training (OR-PIT) algorithm is proposed to train such a network. A binary classifier is trained to produce the stopping signal for the system. Satisfactory results have been achieved on two- and three-speaker mixtures. However, it should be noted that the stopping signal generator needs the entire utterance as input, and can not be easily extended to causal processing.

This study aims to make deep CASA causal. First, all non-causal connections and normalization are replaced with their causal versions throughout the deep CASA network. We then propose two causal clustering algorithms for the sequential grouping stage, both matching the performance of non-causal clustering. Finally, we fine-tune a three-speaker deep CASA system with two-speaker mixtures. The proposed causal deep CASA algorithm achieves excellent results on both two- and three-speaker mixtures, with no knowledge about the speaker number.

The rest of the paper is organized as follows. Section II reviews the non-causal deep CASA system. Causal processing is introduced in Section III. Section IV presents experimental results and comparisons. Concluding remarks are given in Section V.

II. A Deep CASA Approach to Monaural Speaker Separation

Monaural speaker separation aims to separate C speakers $x_c(n),\ c = 1, \ldots, C$, from a single-microphone recording of speech mixture $y(n)$, where $y(n) = \sum_{c=1}^{C} x_c(n)$ and n indexes time. In this section, we review two versions of deep CASA in [18], namely a two-speaker version and a multi-speaker version. The systems are presented in two parts: simultaneous grouping and sequential grouping.

A. Simultaneous Grouping

Given the complex short-time Fourier transform (STFT) of the mixture $Y(t, f)$, where t and f index frame and frequency, simultaneous grouping is performed to separate the C speakers at the frame level. C outputs, $\hat{X}_c(t, f)$ $(c = 1, \ldots, C)$, are generated to estimate the complex STFT of the C speakers. The training of simultaneous grouping follows the tPIT criterion, where the frame-level output-speaker pairing is chosen as the pairing that minimizes the l_1 loss function over all possible speaker permutations. The outputs are then organized to C streams using the resulting tPIT pairings:

$$\hat{X}_c(t, f) \rightarrow \hat{X}_{o_c}(t, f), \quad c = 1, \ldots, C \tag{1}$$

Here o_c denotes the mapping from speaker outputs to speaker streams, which can change across frames. Next, C time-domain signals, $\hat{x}_{o_c}(n)$ $(c = 1, \ldots, C)$, are generated by applying inverse STFT to the organized streams. Finally, a signal-to-noise ratio (SNR) objective $J^{\text{tPIT}-\text{SNR}}$ is used to tune the network:

$$J^{\text{tPIT}-\text{SNR}} = \sum_{c=1}^{C} 10 \log \frac{\Sigma_n x_c(n)^2}{\Sigma_n [x_c(n) - \hat{x}_{o_c}(n)]^2} \tag{2}$$

A Dense-UNet architecture is used for simultaneous grouping, as shown in Fig. 1. It consists a sequence of upsampling layers, downsampling layers, and dense convolutional blocks, and can be divided into two halves. In the first half, an alternation of dense convolutional blocks and downsampling layers projects the input feature map into a high level of abstraction. Dense blocks and upsampling layers are then alternated in the second half to restore the encoded features back to the original resolution. The dense blocks at the same hierarchical level in the two halves are linked with skip connections. The output layers in Dense-UNet estimate complex T-F masks for the C speakers, which are then multiplied with $Y(t, f)$ to generate source estimates $\hat{X}_c(t, f)$. Other details, including the number of layers, downsampling, upsampling, dense convolutional blocks, and frequency mapping layers, follow those in [18].

B. Sequential Grouping

The sequential grouping stage tracks all frame-level spectral estimates $\hat{X}_c(t, f)$, and assigns them to the C speakers. Mixture spectrogram and C spectral estimates (including real, imaginary and magnitude STFT) are stacked to form the input to this stage. Based on the number of concurrent speakers, two versions of sequential grouping are presented as follows.

1) Two-Speaker Sequential Grouping: When there are only two concurrent speakers, a DNN can be trained to project each frame-level input to a D-dimensional embedding vector $\mathbf{V}(t) \in \mathbb{R}^{1 \times D}$. The target label is a two-dimensional indicator vector which gives a one-hot representation of the tPIT output assignment, 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]. A weighted objective function between $\mathbf{V}$ ($T \times D$, where T denotes the total number of frames) and $\mathbf{A}$ ($T \times 2$) is defined:

$$J^{DC-W} = ||\mathbf{W}(\mathbf{V}\mathbf{V}^T - \mathbf{A}\mathbf{A}^T)\mathbf{W}||_F^2 \tag{3}$$

In the above equation, $\mathbf{W}$ denotes a $T \times T$ diagonal matrix whose main diagonal corresponds to a frame-level weight vector $w(t) = \frac{|LD(t)|}{\sum_t |LD(t)|}$, where $LD(t)$ represents the frame-level loss difference (LD) between the two possible speaker assignments. $||\cdot||_F$ denotes the Frobenius norm.

Minimizing J^{DC-W} forces $\mathbf{V}(t)$ corresponding to the same optimal assignment to get closer during training, and otherwise to become farther apart. Clustering $\mathbf{V}(t)$ with the K-means algorithm yields a binary label for each frame, which can be used to organize the frame-level outputs from simultaneous grouping. Deep CASA with such a sequential grouping stage is denoted by two-speaker deep CASA in this study.

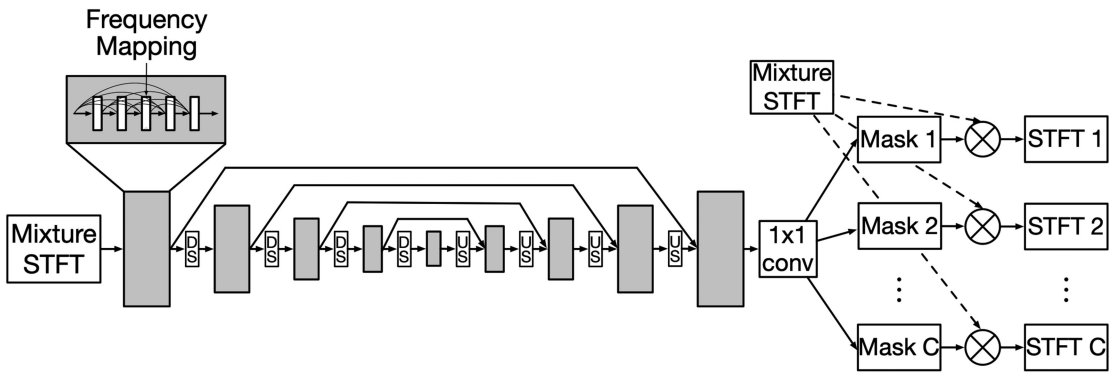

Fig. 1. Diagram of the Dense-UNet in simultaneous grouping. Gray, 'DS' and 'US' blocks denote dense convolutional blocks, downsampling layers and upsampling layers, respectively. Dense convolutional blocks at the same level are linked with skip connections. The inputs, masks and outputs are defined in the complex STFT domain.

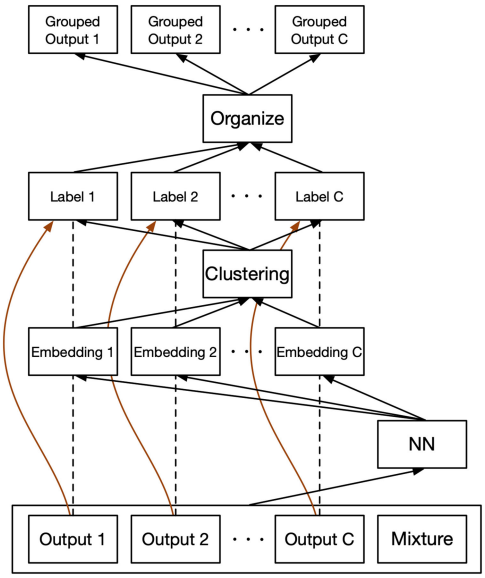

Fig. 2. Diagram of multi-speaker sequential grouping.

Two-speaker sequential grouping works excellently in the case of two concurrent speakers, as there are only two possible output assignments, i.e., swap or not swap. The trained $\mathbf{V}(t)$ exhibits two unique patterns accordingly. However, when the number of speakers C increases, the number of possible assignments is $C! = 1 \times 2 \times \cdots \times C$, and it becomes intractable to use one vector $\mathbf{V}(t)$ to represent all the assignments. Even if $\mathbf{V}(t)$ can be trained to convey $C!$ patterns, it is difficult to figure out the pattern-assignment pairing during inference.

2) Multi-Speaker Sequential Grouping: To avoid the intractable embedding patterns, for a C-speaker ($C \geq 2$) mixture, we use a DNN to predict C embedding vectors at each frame $\mathbf{V}_c(t) \in \mathbb{R}^{1\times D}$, each corresponding to one output $\hat{\mathbf{X}}_c(t)$ of the Dense-UNet, as shown in Fig. 2. The target label for $\mathbf{V}_c(t)$ is a C-dimensional indicator vector, denoted by $\mathbf{A}_c(t)$. During the training of tPIT, if the minimum loss is achieved when $\hat{\mathbf{X}}_c(t)$ is paired with speaker c', the c'th element of $\mathbf{A}_c(t)$ is set to 1, and all other elements are set to 0. In other words, $\mathbf{A}_c(t)$ indicates the optimal speaker assignment of $\hat{\mathbf{X}}_c(t)$. Similar to two-speaker sequential grouping, a weight $w_c(t) = \frac{|LD(t)|}{\sum_t |LD(t)|}$ is used during training to emphasize frames where the speaker assignment plays an important role. Here $LD(t)$ denotes the frame-level loss difference between the minimum and maximum loss. $w_c(t)$ can be used to construct a $CT \times CT$ diagonal weight matrix $\mathbf{W} = \mathrm{diag}(w_c(t))$. $\mathbf{V}_c(t)$ and $\mathbf{A}_c(t)$ can be reshaped into a $CT \times D$ matrix $\mathbf{V}$ and a $CT \times C$ matrix $\mathbf{A}$, respectively. The final weighted objective function between $\mathbf{V}$ and $\mathbf{A}$ is:

$$J^{DC-W} = ||\mathbf{W}(\mathbf{V}\mathbf{V}^T - \mathbf{A}\mathbf{A}^T)\mathbf{W}||_F^2 \quad (4)$$

Optimizing J^{DC-W} forces $\mathbf{V}_c(t)$ corresponding to the same speaker to get closer during training, and $\mathbf{V}_c(t)$ corresponding to different speakers to become farther apart. The trained $\mathbf{V}_c(t)$ exhibits C unique patterns, each corresponding to one speaker.

During inference, the K-means algorithm is first applied to cluster $\mathbf{V}_c(t)$ into C groups. However, if no post-processing is conducted, several embeddings at one frame may be assigned to the same speaker. We thus design a constrained clustering algorithm to force the frame-level embeddings to different labels, as given in Algorithm 1. The input to the algorithm includes C centroids calculated using the K-means algorithm. In each frame, the resulting permutation $\Theta(t)$ corresponds to the assignment that maximizes the sum of similarities between embeddings and centroids. Here P denotes the union of all permutations. After the constrained clustering algorithm, frame-level outputs are organized according to their labels, and resynthesized to the time domain. Deep CASA with multi-speaker sequential grouping is denoted by multi-speaker deep CASA in this study.

Algorithm 1: Constrained Clustering.

Input: Embedding vectors $\mathbf{V}_c(t)$, K-means centroids $\boldsymbol{\mu}_c$

Output: Frame-level labels of all outputs $\Theta(t)$ (resulting permutation)

1: **for** `t in {1,...,T}` **do**

2: $\quad \Theta(t) \leftarrow \mathrm{argmax}_{\theta(t)\in P} \sum_{c=1}^{C} \mathbf{V}_{\theta_c(t)}(t)\boldsymbol{\mu}_c^T$

3: **end for**

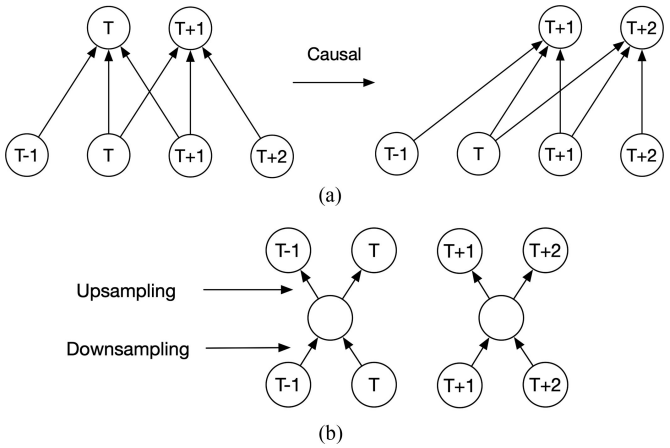

Fig. 3. Temporal convolution in deep CASA. (a) A temporal convolutional layer with matched temporal resolution in the input and output. Non-causal and causal versions are illustrated on the left and right, respectively. (b). Temporal downsampling and upsampling layers in non-causal Dense-UNet.

A temporal convolutional network (TCN) [3], [14] is used as the sequence model for both two-speaker and multi-speaker sequential grouping. In the TCN, input features are fed to 8 consecutive dilated convolutional blocks, with an exponentially increasing dilation factor. The 8 blocks are repeated 3 more times before embedding estimation. A dropDilation technique is utilized to overcome the overfitting problem during training. Other details of the TCN follow those in [18].

To build deep CASA from scratch, the simultaneous grouping and sequential grouping modules need to be trained in turn separately. We have shown in [18] that the two modules can be further fine-tuned jointly with a smaller learning rate to produce smoother source estimates. In joint optimization, the outputs of Dense-UNet are organized using the estimated clustering labels, and compared with the clean sources to form an SNR objective. In the meantime, the sequential grouping module is tuned using the same weighted objective in Eq. (4). Joint optimization is applied in this study.

III. A Causal Extension to Deep CASA

In this section, we present causal deep CASA. To turn deep CASA into a causal version, four aspects need to be examined: temporal convolution, normalization, clustering and speaker-number-independent training.

A. Temporal Convolution

Dense-UNet and TCN consist of a series of temporal convolutional layers, which are non-causal in the original deep CASA system. The left part of Fig. 3(a) illustrates a non-causal temporal convolutional layer in TCN. To generate the output of frame T, future information from frame $T+1$ is used, making the layer non-causal. In the causal extension, we change non-causal convolution to their causal versions when the temporal resolution stays the same in the input and output, as shown in the right part of Fig. 3(a).

There are two special types of temporal convolutional layers in Dense-UNet, downsampling and upsampling layers. Temporal downsampling is achieved using strided convolutional layers of size 2. Upsampling layers are transpose convolutional layers of size 2. Fig. 3(b) illustrates one pass of temporal downsampling and upsampling. During the downsampling process, inputs from every two frames are encoded into one single unit, which halves the temporal resolution. The upsampling layer then projects the encodings to the original resolution. As a result of encoding, the output at frame $T-1$ requires inputs at both frame $T-1$ and T, making the layers non-causal. Since there is no solution to fix the non-causality of such layers, we remove all frame-wise downsampling and upsampling in Dense-UNet, but keep the frequency-wise downsampling and upsampling.

B. Normalization

Normalization is utilized extensively in deep CASA to accelerate training and stabilize neuron activations. Empirical results indicate that the choice of normalization significantly impacts the performance of speaker separation [20]. In non-causal deep CASA, standard layer normalization (LN) [2] is adopted, where the features are normalized over all but the batch dimension. Take Dense-UNet as an example. Feature maps in Dense-UNet have 4 dimensions: $\mathbf{z} \in \mathbb{R}^{B\times T\times F\times K}$, where B, T, F, K denote the batch size, the number of frames, frequency bins, and channels, respectively. A global mean and variance are calculated for each training sample in a batch, and are then utilized to normalize the feature map:

$$E[\mathbf{z}] = \frac{1}{TFK}\sum_{t,f,k}\mathbf{z}(b,t,f,k) \tag{5}$$

$$Var[\mathbf{z}] = \frac{1}{TFK}\sum_{t,f,k}(\mathbf{z}(b,t,f,k) - E[\mathbf{z}])^2 \tag{6}$$

$$LN(\mathbf{z}) = \frac{\mathbf{z} - E[\mathbf{z}]}{\sqrt{Var[\mathbf{z}]+\epsilon}} \odot \boldsymbol{\gamma} + \boldsymbol{\beta} \tag{7}$$

where $\boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^{1\times 1\times 1\times K}$ are trainable gain and bias, ϵ is a small constant added to variance to avoid dividing by zero, and $\odot$ denotes point-wise multiplication. The means and variances are calculated on a whole utterance in both training and inference, which makes layer normalization not applicable in a causal setup.

In this study, we explore three causal normalization techniques as substitutes for layer normalization. In standard batch normalization (BN) [8], features are normalized over all but the channel dimension during training:

$$E[\mathbf{z}] = \frac{1}{BTF}\sum_{b,t,f}\mathbf{z}(b,t,f,k) \tag{8}$$

$$Var[\mathbf{z}] = \frac{1}{BTF}\sum_{b,t,f}(\mathbf{z}(b,t,f,k) - E[\mathbf{z}])^2 \tag{9}$$

$$BN(\mathbf{z}) = \frac{\mathbf{z} - E[\mathbf{z}]}{\sqrt{Var[\mathbf{z}]+\epsilon}} \odot \boldsymbol{\gamma} + \boldsymbol{\beta} \tag{10}$$

where $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ again denote trainable gain and bias. Mean and variance gathered in the training phase are utilized for all test utterances. Since recalculation of statistics is not needed, batch normalization is causal during inference.

Because of the complexity of Dense-UNet/TCN, a small batch size is used (4 or 8) during training. Channel-dependent mean and variance in BN may fluctuate severely across mini-batches. We propose a channel-independent version of batch normalization (ciBN) to overcome this issue. In ciBN, features are normalized over all dimensions during training:

$$E[\mathbf{z}] = \frac{1}{BTFK} \sum_{b,t,f,k} \mathbf{z}(b,t,f,k) \tag{11}$$

$$Var[\mathbf{z}] = \frac{1}{BTFK} \sum_{b,t,f,k} (\mathbf{z}(b,t,f,k) - E[\mathbf{z}])^2 \tag{12}$$

$$ciBN(\mathbf{z}) = \frac{\mathbf{z} - E[\mathbf{z}]}{\sqrt{Var[\mathbf{z}] + \epsilon}} \odot \boldsymbol{\gamma} + \boldsymbol{\beta} \tag{13}$$

Mean and variance gathered in training are used for inference.

We also consider a causal version of layer normalization (cLN), where the features are normalized in a causal fashion.

$$E[\mathbf{z}(t=\tau)] = \frac{1}{\tau FK} \sum_{t \le \tau, f, k} \mathbf{z}(b,t,f,k) \tag{14}$$

$$Var[\mathbf{z}(t=\tau)] = \frac{1}{\tau FK} \sum_{t \le \tau, f, k} (\mathbf{z}(b,t,f,k) - E[\mathbf{z}(t=\tau)])^2 \tag{15}$$

$$cLN(\mathbf{z}(t=\tau)) = \frac{\mathbf{z}(t=\tau) - E[\mathbf{z}(t=\tau)]}{\sqrt{Var[\mathbf{z}(t=\tau)] + \epsilon}} \odot \boldsymbol{\gamma} + \boldsymbol{\beta} \tag{16}$$

Here $\mathbf{z}(t=\tau)$ denotes the τth frame of the feature map. In cLN, normalization is conducted frame by frame, with frame-dependent mean and variance calculated using all previous frames. A similar normalization technique was used in the causal version of Conv-TasNet [20].

The three causal normalization techniques can also be applied to the TCN in the sequential grouping stage. All operations stay the same, but the frequency dimension is neglected.

In addition to BN, ciBN and cLN, we plan to explore multi-GPU training with data parallelism and synchronized batch normalization in future research, which can greatly increase the batch size in training.

C. Clustering

Once embedding vectors are generated, a clustering step is needed to assign them to different speakers. Most clustering based speaker separation algorithms, e.g., deep clustering and deep CASA, perform this step in an offline fashion. In deep clustering, the K-means algorithm iteratively generates centroids of clusters using all embedding vectors in the whole utterance. It is difficult to make a causal extension to K-means for deep clustering, as embedding vectors corresponding to some clusters may not be present in the beginning part of an utterance. Therefore, the number of clusters is unclear for causal processing.

On the other hand, in the setting of multi-speaker deep CASA, there are C embedding vectors in each frame, each belonging to a unique cluster. The design of causal clustering becomes much easier. The details are given in Algorithm 2.

Algorithm 2: Causal Clustering for Multi-Speaker Deep CASA.

Input: Embedding vectors $\mathbf{V}_c(t)$, frame-level energy of the mixture $E(t)$, energy threshold α, maximal queue size $S_{\max}$
Output: Frame-level labels of all outputs $\Theta(t)$
for `c in {1,...,C}` **do**
  $Q_c \leftarrow$ `NEW_FIFO_QUEUE()`
  Q_c.`enqueue`$(\mathbf{V}_c(1))$
  $\boldsymbol{\mu}_c \leftarrow Q_c$.`mean()`
  $\Theta_c(1) \leftarrow c$
end for
$E_{\max} \leftarrow E(1)$, $t \leftarrow 2$
while `t <= T` **do**
  $\Theta(t) \leftarrow \text{argmax}_{\theta(t) \in P} \sum_{c=1}^{C} \mathbf{V}_{\theta_c(t)}(t) \boldsymbol{\mu}_c^T$
  if $E(t) > \alpha E_{\max}$ **then**
    for `c in {1,...,C}` **do**
      Q_c.`enqueue`$(\mathbf{V}_{\Theta_c(t)}(t))$
      if Q_c.`size`$() > S_{\max}$ **then**
        Q_c.`dequeue`$()$
      end if
      $\boldsymbol{\mu}_c \leftarrow Q_c$.`mean()`
    end for
  end if
  $E_{\max} \leftarrow$ `max`$(E_{\max}, E(t))$
  $t \leftarrow t + 1$
end while

At the start of the algorithm, C first-in-first-out (FIFO) queues are created to store embedding vectors belonging to the clusters. Each embedding vector in the first frame is pushed to one of the queues to form the initial data. Centroids of the clusters are calculated as mean values of the queues. Starting from frame two, each embedding vector is assigned to a unique cluster using the assignment that maximizes the sum of similarities between embeddings and centroids. If the energy of the current frame is insignificant, we move to the next frame. Otherwise, we push the embedding vectors to their corresponding queues, and update the centroids. In order to keep the centroids relatively near the current frame, we remove the oldest item in the queue when the size of the queue exceeds $S_{\max}$. To decide whether a frame has significant energy, we keep track of the maximum frame energy $E_{\max}$. Frames whose energy is weaker than $\alpha E_{\max}$ are considered uninformative, and would not be used for centroid calculation. The frame-level assignment continues until all frames are processed. The two parameters α and $S_{\max}$ are set to 0.3 and 20 in our study, and the system performance is insensitive to these specific values.

We also design a causal clustering algorithm for two-speaker deep CASA, as shown in Algorithm 3. In two-speaker deep CASA, each frame only has one embedding vector, indicating the frame-level optimal assignment. At the first frame, we create 2 FIFO queues to store embedding vectors. The first embedding vector is pushed to the first queue. Starting from frame two, if the second queue is empty, we check the similarity of embedding

Algorithm 3: Causal Clustering for Two-Speaker Deep CASA.

Input: Embedding vectors $\mathbf{V}(t)$, frame-level energy of the mixture $E(t)$, energy threshold α, similarity threshold ρ, maximal queue size $S_{\max}$
Output: Frame-level label $\Theta(t)$
for `c in {1,2}` **do**
  $Q_c \leftarrow$ `NEW_FIFO_QUEUE()`
end for
Q_1.`enqueue`$(\mathbf{V}(1))$
$\boldsymbol{\mu}_1 \leftarrow \mathbf{V}(1)$, $E_{\max} \leftarrow E(1)$, $\Theta(1) \leftarrow 1$, $t \leftarrow 2$
while `t` $\leq$ `T` **do**
  if Q_2.`empty()` **then**
    if $\mathbf{V}(t-1)\mathbf{V}(t)^T < \rho$ **then**
      $\Theta(t) \leftarrow 2$
    else
      $\Theta(t) \leftarrow 1$
    end if
  else
    $\Theta(t) \leftarrow \text{argmax}_{c\in\{1,2\}} \mathbf{V}(t)\boldsymbol{\mu}_c^T$
  end if
  if $E(t) > \alpha E_{\max}$ `or` (Q_2.`empty()` `and` $\Theta(t) == 2$) **then**
    $Q_{\Theta(t)}$.`enqueue`$(\mathbf{V}(t))$
    if $Q_{\Theta(t)}$.`size()` $> S_{\max}$ **then**
      $Q_{\Theta(t)}$.`dequeue()`
    end if
    $\boldsymbol{\mu}_{\Theta(t)} \leftarrow Q_{\Theta(t)}$.`mean()`
  end if
  $E_{\max} \leftarrow$ `max`$(E_{\max}, E(t))$
  $t \leftarrow t+1$
end while

vectors between the current frame and the previous frame. If the similarity is lower than ρ, we set the current frame to cluster 2, and push the embedding vector to the second queue. Otherwise the current frame is set to cluster 1, and the checking continues. Once the second queue loads the first item, the algorithm starts to follow the same process as in Algorithm 2. The energy threshold α, similarity threshold ρ, and $S_{\max}$, are set to 0.3, 0.5 and 10, respectively. Both Algorithm 2 and 3 are easy to implement and fast during inference.

D. Speaker-Number-Independent Training

The total number of concurrent speakers is usually unknown in real-world causal applications. A system that generalizes well to an unknown speaker number is crucial for these situations. Although multi-speaker deep CASA is designed for C concurrent ($C \geq 2$) speakers, if trained properly, a C-speaker system can generate good results for speech mixtures with less than C speakers, without the prior knowledge about the speaker number. In such cases, some of the outputs produce significantly lower energy than other outputs, corresponding to silence. The details of speaker-number-independent training are presented in Section IV-C.

IV. Evaluation and Comparison

A. Experimental Setup

We evaluate our systems on two-speaker and three-speaker separation datasets, WSJ0-2mix and WSJ0-3mix [6]. Both datasets have a 30-hour training set and a 10-hour validation set generated by selecting random speakers in the Wall Street Journal (WSJ0) training set, and mixing them at various SNRs between 0 dB and 5 dB. Evaluation is conducted on the open-condition (OC) test sets, which are similarly generated using 16 untrained speakers from the WSJ0 development set. All mixtures are sampled at 8 kHz. We calculate STFT with a frame length of 32 ms, a frame shift of 8 ms, and a square root Hanning window.

Performance is evaluated in terms of signal-to-distortion ratio improvement (ΔSDR) [24], perceptual evaluation of speech quality (PESQ) whose values range from -0.5 to 4.5 [9], and extended short-time objective intelligibility (ESTOI) whose values range typically between 0 and 1 [10]. Results are also reported in terms of scale-invariant signal-to-noise ratio improvement (ΔSI-SNR) [21] for a systematical comparison with other systems.

B. Models

All deep CASA systems in this study adopt the basic structure of Dense-UNet and TCN as in [18]. In Dense-UNet, the number of dense layers in a dense block is set to 5, the number of channels in each dense layer is set to 64, and all dense layers have a kernel size of 3×3 and a stride of 1×1. The middle layer in each dense block is replaced with a frequency mapping layer. The network is optimized with respect to $J^{t\text{PIT}-\text{SNR}}$.

In TCN, the maximum dilation factor is set to 64. The number of bottleneck units is selected as 256. The number of units in depthwise dilated convolutional layers is set to 512. DropDilation with a keep rate of 0.7 is applied during training.

Both networks are trained with the Adam optimization algorithm [11]. The initial learning rate is set to 0.0001 for Dense-UNet, and 0.00025 for TCN. Learning rate adjustment and early stopping are employed based on the loss on the validation set.

For causal deep CASA, temporal connections, normalization and clustering algorithms are modified as described in Section III. The model looks back 72 past frames in simultaneous grouping, and 1016 past frames in sequential grouping. Thus the theoretical receptive field of causal deep CASA is 8.704 seconds, all in the past. The latency of causal deep CASA corresponds to one frame of STFT, which is 32 ms.

C. Results and Comparisons

We first evaluate causal deep CASA on two-speaker mixtures. Different simultaneous grouping models are compared in Table I. Outputs are organized with the optimal speaker assignment before evaluation. The first row corresponds to Dense-UNet with non-causal connections and normalization. A modest performance drop is observed when we switch to the causal versions. Two normalization techniques are evaluated for causal processing. BN leads to negligibly better results than ciBN. Due to slow training, we did not use cLN for causal Dense-UNet, and leave it as future work.

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

Simul. Group.	Temporal convolution	Normalization	ΔSDR (dB)	PESQ	ESTOI (%)
Dense-UNet	Non-causal	LN	19.1	3.63	94.3
Dense-UNet	Causal	BN	18.0	3.52	93.2
Dense-UNet	Causal	ciBN	17.8	3.52	93.1

TABLE II
AVERAGE ΔSDR, PESQ AND ESTOI FOR DIFFERENT SEQUENTIAL GROUPING MODELS ON WSJ0-2MIX OC

Seq. Group.	Temporal convolution	Normalization	Clustering	ΔSDR (dB)	PESQ	ESTOI (%)
Two-speaker	Causal	BN	Causal	13.9	3.02	87.0
Two-speaker	Causal	ciBN	Causal	14.6	3.12	88.5
Two-speaker	Causal	cLN	Causal	15.1	3.18	89.4
Multi-speaker	Causal	cLN	Causal	14.8	3.14	88.9
Two-speaker	Causal	cLN	Offline	15.2	3.19	89.5
Multi-speaker	Causal	cLN	Offline	14.9	3.16	89.0

TABLE III
AVERAGE ΔSDR, PESQ, ESTOI AND FRAME ASSIGNMENT ERROR (FAE) FOR DEEP CASA WITH JOINT OPTIMIZATION ON WSJ0-2MIX OC

Deep CASA with joint optimization	Causal	ΔSDR (dB)	PESQ	ESTOI (%)	FAE (%)
Two-speaker	✗	**18.0**	**3.51**	**93.2**	**1.22**
Multi-speaker	✗	17.8	3.50	93.0	1.45
Two-speaker	✓	**15.5**	**3.25**	**90.1**	**3.58**
Multi-speaker	✓	15.2	3.23	89.7	3.86

Table II compares different sequential grouping models for two-speaker mixtures. All sequential grouping TCNs in the table are built with causal connections and normalization, and trained on top of the causal Dense-UNet with BN. The first three rows compare three causal normalization techniques for two-speaker deep CASA. Thanks to the matched calculation of statistics in the training and test, cLN substantially outperforms the other two techniques. We also train a causal TCN with cLN under the multi-speaker setup, as given in the fourth row. It performs slightly worse than the two-speaker version, reflecting the principle of Occam's razor. When the number of concurrent speakers is fixed to 2, one embedding vector per frame is enough to indicate the optimal output assignment. The extra embedding vectors in multi-speaker deep CASA do not convey much information, and lead to worse performance during inference.

The last two rows in Table II report the results of causal TCNs with non-causal clustering. All settings follow the third and fourth row in Table II except for the clustering algorithms. The causal clustering algorithms yield almost the same results as non-causal offline clustering, demonstrating the effectiveness of the proposed clustering.

Next, we jointly optimize the two stages of deep CASA. The results are reported in Table III. Four deep CASA systems are evaluated, either causal or non-causal, and two-speaker or multi-speaker. Joint optimization is performed in a similar fashion as in [18]. Compared to the results in Table II, modest improvements are achieved by causal deep CASA when joint optimization is performed. There is still a small gap between two-speaker and multi-speaker deep CASA in Table III, consistent with Table II. In addition to ΔSDR, PESQ and ESTOI, frame assignment error (FAE) is reported to show the percentage of incorrectly assigned frames in terms of minimum frame-level loss, in other words, errors in speaker tracking. FAE nearly triples when we switch from non-causal deep CASA to the causal ones, which suggests a major cause why causal deep CASA performs worse in terms of all metrics.

To further illustrate the FAE of non-causal and causal deep CASA, we compare their separated results in Fig. 4. The first two rows show a male-male test mixture and the two target speakers. The third row shows the results of non-causal deep CASA, which makes correct assignment decisions in almost every frame, and only misses a few high frequency details. The fourth row corresponds to causal deep CASA. From 0 s to 2.5 s, and 3.3 s to 5.5 s, causal and non-causal deep CASA almost generate identical outputs. However, causal deep CASA makes successive incorrect assignments between 2.5 s and 3.3 s, due to the lack of future information and limited past information.

Table IV compares causal deep CASA (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 their papers. The second and third row present two non-causal methods. Conv-TasNet [20] extends uPIT to the waveform domain using a convolutional neural network. The sign prediction network [28] combines DC and uPIT, and train a separate network for phase reconstruction. All the other systems in the table are causal. The Listen and Group system [16] estimates frame-level spectral

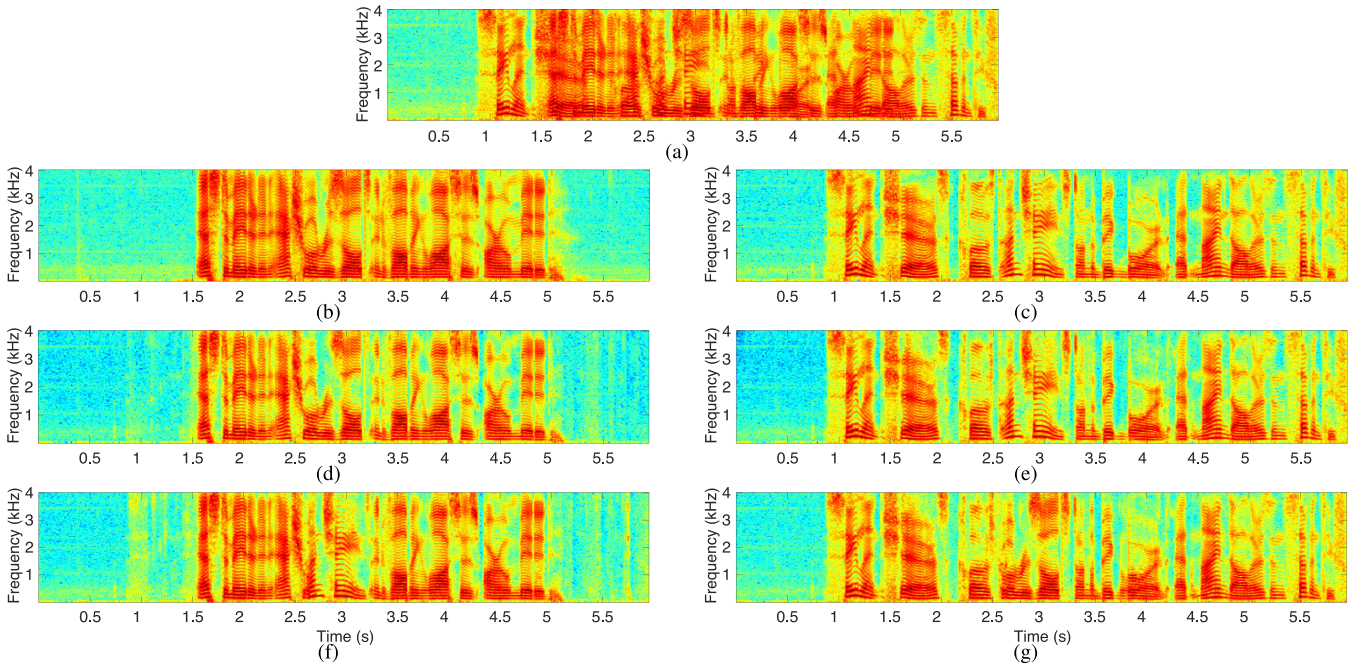

Fig. 4. Speaker separation results of deep CASA in log magnitude STFT. Two jointly-optimized two-speaker models, non-causal and causal deep CASA, are compared. (a) A male-male test mixture. (b) Speaker 1 in the mixture. (c) Speaker 2 in the mixture. (d) Non-causal deep CASA's output 1. (e) Non-causal deep CASA's output 2. (f) Causal deep CASA's output 1. (g) Causal deep CASA's output 2.

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

	# of param.	Causal	ΔSDR (dB)	ΔSI-SNR (dB)	PESQ	ESTOI (%)
Mixture	-	-	0.0	0.0	2.02	56.1
Conv-TasNet [20]	5.1M	✗	15.6	15.3	3.24	-
Sign Prediction Net [28]	56.6M	✗	15.4	15.2	3.45	-
uPIT [13]	46.3M	✓	7.0	-	-	-
Conv-TasNet [20]	5.1M	✓	11.0	10.6	-	-
LSTM-TasNet [20]	32.0M	✓	11.2	10.8	-	-
Listen and Group [16]	8.2M	✓	11.0	-	-	-
Two-speaker deep CASA	12.8M	✓	**15.5**	**15.2**	**3.25**	**90.1**
IBM	-	-	13.8	13.4	3.28	89.1

outputs in an autoregressive fashion. It consists of two stages. In the first stage, the frame-level mixture and source estimates from the previous frame are transformed into mid-level representations. The second stage groups mid-level representations to two sources. We present the fully causal version of Listen and Group, which has no look-aheads for phase reconstruction. Other models include causal versions of uPIT, LSTM-TasNet and Conv-TasNet. As demonstrated in the table, our causal deep CASA system outperforms all causal methods by a large margin. It even surpasses the ideal binary mask (IBM), and matches the performance of non-causal Conv-TasNet, demonstrating the power of the proposed causal extension.

Table V compares multi-speaker deep CASA (with joint optimization) and other state-of-the-art methods on three-speaker mixtures WSJ0-3mix. As shown in the upper half of the table, deep CASA produces systematically better results than other methods under the non-causal setup. When we switch to the causal setting, the performance of deep CASA drops significantly as expected, mostly due to the lack of future information for sequential grouping. Despite the fact that causal processing lacks future information, which is inherently useful for speech processing, the proposed causal extension keeps the assignment errors to a low level and substantially outperforms the best published causal results by Conv-TasNet [20] on WSJ0-3mix OC.

Although multi-speaker deep CASA is designed for C concurrent speakers, in theory, a C-speaker system can be directly applied to speech mixtures with less than C speakers. In Table VI, we evaluate the three-speaker deep CASA systems presented in Table V on two-speaker mixtures (WSJ0-2mix OC). The two outputs with significant energy are selected as active speakers during evaluation. As shown in Table VI, the three-speaker

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

	# of param.	Causal	ΔSDR (dB)	ΔSI-SNR (dB)	PESQ	ESTOI (%)
Mixture	-	-	0.0	0.0	1.66	38.5
uPIT [13]	92.7M	✗	7.7	-	-	-
Conv-TasNet [20]	5.1M	✗	13.1	12.7	2.61	-
Sign Prediction Net [28]	56.6M	✗	12.5	12.1	2.77	-
Multi-speaker deep CASA	12.8M	✗	**14.8**	**14.5**	**2.83**	**81.5**
Conv-TasNet [20]	5.1M	✓	8.2	7.8	-	-
Multi-speaker deep CASA	12.8M	✓	**10.1**	**9.8**	**2.28**	**70.6**
IBM	-	-	13.6	13.3	2.86	82.1

TABLE VI
AVERAGE ΔSDR, PESQ AND ESTOI FOR MULTI-SPEAKER DEEP CASA, TRAINED ON WSJ0-3MIX AND EVALUATED ON WSJ0-2MIX OC

	Training set	Causal	ΔSDR (dB)	ΔSI-SNR (dB)	PESQ	ESTOI (%)
Multi-speaker deep CASA	WSJ0-3mix	✗	14.8	14.4	3.12	87.6
		✓	11.4	10.9	2.76	82.7

TABLE VII
AVERAGE ΔSDR, ΔSI-SNR, PESQ AND ESTOI FOR SPEAKER-NUMBER-INDEPENDENT SYSTEMS EVALUATED ON WSJ0-2MIX OC AND WSJ0-3MIX OC

	Causal	WSJ0-2mix OC				WSJ0-3mix OC			
		ΔSDR (dB)	ΔSI-SNR (dB)	PESQ	ESTOI (%)	ΔSDR (dB)	ΔSI-SNR (dB)	PESQ	ESTOI (%)
uPIT [13]	✗	10.1	-	-	-	7.8	-	-	-
OR-PIT [23]	✗	15.0	14.8	3.12	-	12.9	12.6	2.60	-
Multi-speaker deep CASA	✗	**17.6**	**17.4**	**3.40**	**92.0**	**14.8**	**14.5**	**2.77**	**81.2**
Multi-speaker deep CASA	✓	**14.2**	**13.9**	**3.06**	**87.8**	**10**	**9.6**	**2.12**	**69.7**

systems yield substantially worse results than the two-speaker systems (cf. Table III) on two-speaker test mixtures, possibly due to the mismatch between training and test. Moreover, there is significant residual energy in the discarded output of three-speaker deep CASA, i.e., -16.9 dB relative to the other two outputs.

To make the three-speaker systems generalize to two-speaker mixtures, we fine-tune three-speaker deep CASA with mixtures from both WSJ0-2mix and WSJ0-3mix. The fine-tuning is conducted similarly as joint optimization, where the two stages are updated together with a small learning rate. To enable the training of three-speaker models on WSJ0-2mix, we extend WSJ0-2mix with a third silent channel, which contains zero energy. To avoid infinite SNR objective for the silent channel, a time-domain l_1 loss is used instead to tune the simultaneous grouping module. Table VII shows the results of three-speaker deep CASA fine-tuned on WSJ0-2mix and WSJ0-3mix, and compares it with other speaker-number-independent approaches trained on WSJ0-2mix and WSJ0-3mix. All comparison approaches are uPIT based, as deep clustering based methods do not perform well when the number of speakers is unknown. The results are reported on both WSJ0-2mix OC and WSJ0-3mix OC. The first three rows summarize speaker-number-independent training of non-causal systems, where deep CASA substantially outperforms the other two approaches in terms of all four metrics on both datasets. While there is no prior result on a causal algorithm for speaker-number-independent separation, speaker-number-independent causal deep CASA, as shown in the fourth row, yields satisfactory results, and even outperforms speaker-number-dependent causal methods in Table IV and V. For two-speaker mixtures, speaker-number-independent training reduces the relative energy in the discarded output of three-speaker deep CASA to -43.5 dB, negligible for practical utility.

TABLE VIII
REAL TIME FACTOR OF DEEP CASA

	RTF
Causal deep CASA	0.0110
Non-causal deep CASA	0.0077

Finally Table VIII reports the computational costs of neural networks in terms of real time factor (RTF), which is defined as the ratio of processing time to input signal duration. RTF is evaluated by running the neural networks for three-speaker deep CASA (implemented in Tensorflow) on a single NVIDIA V100 GPU. One hundred seconds of input mixtures are processed for evaluation. As shown in the table, although the removal of temporal downsampling layers slightly slows the inference speed of causal deep CASA, it runs much faster than real time. Non-causal and causal DNNs are on a similar scale in terms of RTF. In addition to the neural networks, all clustering algorithms in this study have the complexity of $O(T)$ and can run fast on CPUs with proper optimization.

V. Conclusions

We have proposed a causal deep CASA algorithm for monaural talker-independent speaker separation. We adapt temporal connections and normalization in deep CASA, and propose two causal clustering algorithms. Experimental results on the benchmark WSJ0-2mix and WSJ0-3mix datasets show that the proposed causal algorithm outperforms all published results for causal speaker separation. In addition, speaker-number-independent training broadens the utility of causal deep CASA to a more realistic scenario when the speaker number is not given beforehand. This study represents a major step towards speaker separation in real-time applications.

Although causal deep CASA shows excellent performance on simulated datasets, it has several limitations. First, its performance degrades substantially with the increase of concurrent speakers. It should be noted that this is a common problem in other studies [20], [23], due to the fact that additional speakers increase the difficulty of both simultaneous and sequential grouping. Second, the current system assumes simultaneous speakers, and does not perform well on real conversations with varying degrees of overlapped speech. Third, in this study, causal deep CASA is only trained and evaluated on clean speaker mixtures without other kinds of interference. Recently, we have extended non-causal deep CASA to deal with room reverberation [4] and background noise [17]. We plan to extend causal deep CASA to overcome these limitations in future research.

References

[1] R. Aihara, T. Hanazawa, Y. Okato, G. Wichern, and J. L. Roux, "Teacher-student deep clustering for low-delay single channel speech separation," in *Proc. Int. Conf. Acoust., Speech*, 2019, pp. 690–694.

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

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

[4] M. Delfarah, Y. Liu, and D. L. Wang, "Talker-independent speaker separation in reverberant conditions," in *Proc. Int. Conf. Acoust., Speech*, 2020, pp. 8723–8727.

[5] R. Herbig and J. Chalupper, "Acceptable processing delay in digital hearing aids," *Hearing Rev.*, vol. 17, pp. 28–31, 2010.

[6] J. R. Hershey, Z. Chen, J. L. Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. Int. Conf. Acoust., Speech*, 2016, pp. 31–35.

[7] T. Higuchi, K. Kinoshita, M. Delcroix, K. Zmolíková, and T. Nakatani, "Deep clustering-based beamforming for separation with unknown number of sources," in *Proc. Interspeech*, 2017, pp. 1183–1187.

[8] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[9] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ) A new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 2, 2001, pp. 749–752,.

[10] 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.

[11] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.

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

[13] 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.

[14] 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.

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

[16] 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.

[17] Y. Liu, M. Delfarah, and D. L. Wang, "Deep CASA for talker-independent monaural speech separation," in *Proc. Int. Conf. Acoust., Speech*, 2020, pp. 6354–6358.

[18] Y. Liu and D. L. Wang, "Divide and conquer: A deep CASA approach to talker-independent monaural speaker separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 12, pp. 2092–2102, Dec. 2019.

[19] 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.

[20] 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.

[21] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR – half-baked or well done?" in *Proc. Int. Conf. Acoust., Speech*, 2019, pp. 626–630.

[22] Z. Shi, H. Lin, L. Liu, R. Liu, J. Han, and A. Shi, "Deep attention gated dilated temporal convolutional networks with intra-parallel convolutional modules for end-to-end monaural speech separation," in *Proc. Interspeech*, 2019, pp. 3183–3187.

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

[24] 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.

[25] D. L. Wang and G. Brown, Eds.,*Computational Auditory Scene Analysis: Principles, Algorithms and Applications*. New York, NY, USA: Wiley, 2006.

[26] Z.-Q. Wang, J. L. Roux, and J. R. Hershey, "Alternative objective functions for deep clustering," in *Proc. Int. Conf. Acoust., Speech*, 2018, pp. 686–690.

[27] Z.-Q. Wang, J. L. 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.

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