Interspeech 2022
18-22 September 2022, Incheon, Korea

Attentive Training: A New Training Framework for Talker-independent Speaker Extraction

Ashutosh Pandey and DeLiang Wang

[1]Department of Computer Science and Engineering, The Ohio State University, USA
{pandey.99, wang.77}@osu.edu
email@address

Abstract

Listening in a multitalker scenario, we typically attend to a single talker through auditory selective attention. Inspired by human selective attention, we propose attentive training: a new training framework for talker-independent speaker extraction with an intrinsic selection mechanism. In the real world, multiple talkers very unlikely start speaking at the same time. Based on this observation, we train a deep neural network to create a representation for the first speaker and utilize it to extract or track that speaker from a multitalker noisy mixture. Experimental results demonstrate the superiority of attentive training over widely used permutation invariant training for talker-independent speaker extraction, especially in mismatched conditions in terms of the number of speakers, speaker interaction patterns, and the amount of speaker overlaps.

Index Terms: speaker extraction, speaker separation, talker-independent, attentive training

1. Introduction

The cocktail party effect refers to the amazing ability of auditory perception attending to (hence extracting) a single speaker in a multitalker noisy scenario [1]. This effect has influenced the development of the perceptual theory of selective attention [2]. Separating all speakers or extracting a single one from a multitalker mixture is considered very challenging for machines, however, the introduction of deep learning to such tasks has led to dramatic advances in recent years [3].

Currently, there are two approaches to speaker extraction: speaker separation and target speaker extraction. Speaker separation aims at separating all speakers from a mixture. Early speaker separation work is extended from deep neural network (DNN) based speech enhancement, and such separation is talker-dependent. When applied to talker-independent speaker separation, these models suffer from the well-known permutation ambiguity problem, i.e., an underlying model cannot consistently assign DNN output streams to different speakers during training. Deep clustering [4] and permutation invariant training (PIT) [5] are two representative approaches to resolving the permutation ambiguity problem. In particular, the simplicity of PIT has led to many subsequent models for speaker separation [6, 7, 8, 9, 10].

Target speaker extraction aims at extracting a single speaker from a multitalker mixture, where the target speaker is cued with the help of some additional information in the form of audio [11, 12, 13, 14, 15, 16] or images [17, 18, 19]. Other kinds of cues include spatial [20, 21], speech activity [22], or onsets [23]. Target speaker extraction is closer to auditory selective attention, but it is not intrinsic to model training.

In this work, we propose a new training framework, which we name *attentive training*, for talker-independent speaker extraction (or tracking). In the real-world environments, it is very unlikely that multiple talkers start speaking at the same time; such a case would lead to their grouping into the same auditory stream on the basis of common onsets [24]. Therefore, we can assume that a given multitalker mixture has nonoverlapping speech in the beginning. We provide this important cue to a DNN to create a representation for the starting speaker in the early part of processing and utilize this representation for attending to this speaker throughout training. In other words, network selects the first speaker and then attends to it for the rest of the multitalker mixture. This, in a way, resembles speech enhancement in the sense that we treat the speech signals of the first speaker as *speech*, and the utterances of other speakers plus environmental sounds as *background noise*.

The attentive training framework is consistent with the dominant feature integration theory of attention [25]. According to this theory, attention serves to integrate perceptual features extracted in separate analyses into an object. Learning and attention are integral parts of perception.

A similar idea of using speaker onsets as a cue was proposed in serialized output training (SOT) [26]. The SOT uses onset order of speakers to determine the transcription order at the output of an automatic speech recognition (ASR) system. The proposed attentive training, on the other hand, aims at dealing with speaker interference in a speech enhancement system, and hence, is fundamentally different from SOT. The SOT outputs transcriptions of all the speakers in a mixture, whereas the attentive training outputs the enhanced speech corresponding to the first speaker.

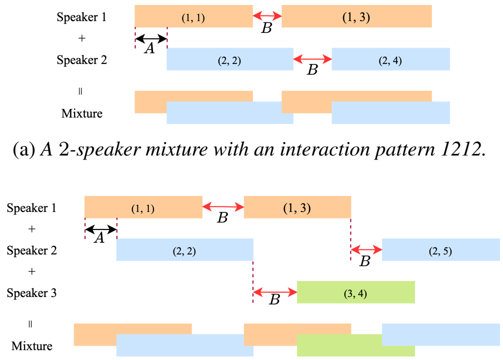
To evaluate the efficacy of attentive training, we create a multitalker dataset by setting the first speaker to start slightly ahead of the rest of speakers in an utterance. Next, we train an end-to-end time-domain model to predict the first speaker in a given mixture. We demonstrate that a DNN model trained with attentive training generalizes well to different test conditions, such as an untrained number of speakers, utterances with larger gaps between consecutive segments of the target speaker, and smaller speaker overlaps.

Further, we compare attentive training with PIT, and observe significant improvements in many conditions, especially in mismatched conditions. We demonstrate that attentive training can overcome the shortcomings of PIT, such as sensitivity to the number of speakers, interaction patterns and amount of speaker overlaps.

Additionally, we introduce a novel data generation technique for mixing an arbitrary number of speakers in a controlled way. Given a set of speakers, their corresponding utterances,

This research was supported in part by two NIDCD grants (R01DC012048 and R02DC015521) and the Ohio Supercomputer Center.

10.21437/Interspeech.2022-10491



(a) *A* 2*-speaker mixture with an interaction pattern 1212.*

(b) *A* 3*-speaker mixture with an interaction pattern 12132.*

Figure 1: *Examples of interaction patterns with* 2 *and* 3 *speakers, and a minimum onset gap of* A *between the first and the second speaker. In pair* (a, b) *inside a box,* a *and* b *respectively represent the speaker order and the segment order.*

and a set of noises, our technique can mix any number of speakers with specified overlaps and speaker orders. Also, mixtures are generated dynamically during training which provides an additional advantage of data augmentation [10]. Our data generation technique should be a useful tool for speaker separation and diarization research, as it can utilize speakers from any corpora and generate mixtures in a flexible way. We provide our data generation script online.

Although this study focuses on extracting the first speaker from a mixture, we believe that the simple and effective mechanism of attentive training has the potential to be applicable to a variety of selection, tracking, and related tasks, such as multitalker speaker separation and speaker diarization. For speaker separation and diarization, a straightforward extension would be to use an iterative strategy where the first speaker is extracted first, then second, and so on, as in [27].

2. Attentive Training Mechanism

2.1. Problem Definition

A multitalker mixture $\boldsymbol{y}$ with N samples is modeled as

$$\boldsymbol{y} = \sum_{i=1}^{C} \boldsymbol{s}_i + \boldsymbol{n} \tag{1}$$

where $\{\boldsymbol{y}, \boldsymbol{s}_i, \boldsymbol{n}\} \in \mathbb{R}^{1 \times N}$, $\boldsymbol{s}_i$ is the i^{th} speaker, $\boldsymbol{n}$ is the background noise, and C is the total number of speakers. Let m_i represent the sample when i^{th} speaker starts to speak. Without the loss of generality, we can assume that indices $i = 1, 2, \ldots, C$ are sorted in the order of $m_i < m_{i+1}$. The goal of attentive training is to get a close estimate $\hat{\boldsymbol{s}}_1$ of the first speaker $\boldsymbol{s}_1$ from $\boldsymbol{y}$.

2.2. Data Generation

In this section, we describe our technique to generate multitalker mixture. We assume that we are given a set $\boldsymbol{S} = \{S_1, \ldots, S_M\}$ of speakers, their corresponding utterances $\boldsymbol{U}^{S_k} = \{\boldsymbol{s}_1^k, \ldots, \boldsymbol{s}_{kN}^k\}$, and a set of noise segments $\boldsymbol{N} = \{\boldsymbol{n}_1, \ldots, \boldsymbol{n}_{nN}\}$. We create a multitaker mixture by adding together speech segments of multiple speakers and a noise segment.

First, we sort a given set of speech segments in an increasing order of their onset times, and based on this, define a concept called *interaction pattern* representing the order of speakers in a mixture. For example, an interaction pattern of 1212 represents a mixture created by adding 4 segments sorted in increasing order of their onset times, where the first and the third segments are from the first speaker and the second and the fourth segments are from the second speaker. We also define two parameters A and B, where A is the minimum allowed gap between the onset of the first and the second speaker, and B is the gap between two adjacent nonoverlapping segments (regardless of speakers). We illustrate two interaction patterns in Fig. 1. For data generation, we use interaction patterns from a predefined set $\boldsymbol{P} = \{p_1, \ldots, p_P\}$.

Algorithm 1 A pseudo code for generating a random multitalker mixture.

```
Input: $\boldsymbol{S}, \boldsymbol{U}, \boldsymbol{N}, \boldsymbol{P}$
Output: $\boldsymbol{y}, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_C, \boldsymbol{n}$
Sample a speaker pattern $p$ from $\boldsymbol{P}$
Set $C$ = Len(Unique($p$))
Sample $C$ speakers $S_1, \ldots, S_C$ from $\boldsymbol{S}$
Initialize List $\boldsymbol{V}$ = [ ], $\boldsymbol{E}$ = [ ], Set $\boldsymbol{E}_1$ ={ }, Bool Overlap = False
for $k$ in $p$ and $i$ in $\{1, 2, \ldots,$ Len(p)$\}$ do
    Sample an utterance $\boldsymbol{s}$ from $\boldsymbol{U}^{S_k}$;
    Remove silences in the beginning and the end of $\boldsymbol{s}$
    Sample a value $T$ for segment length
    Extract a random segment $\boldsymbol{x}$ of length $T$ from $\boldsymbol{s}$
    Set Overlap = True with a probability $p_{overlap}$
    if i = 1 then
        PadLeft = 0
    else if i = 2 then
        Sample a value for $B$
        if $k$ in $\boldsymbol{E}_1$ then
            Overlap = False
        end if
        if no Overlap then
            Set PadLeft = $\boldsymbol{E}$[ -1 ] + $B$
        else
            Sample PadLeft from [A, $\boldsymbol{E}$[ -1 ]]
        end if
    else
        Sample a value for $B$
        if $k$ in $\boldsymbol{E}_1$ and $\boldsymbol{E}_1$[k] = $\boldsymbol{E}[-1]$ then
            Overlap = False
        end if
        if no Overlap then
            Set PadLeft = $\boldsymbol{E}$[ -1 ] + $B$
        else
            Sample PadLeft from [$\boldsymbol{E}$[ -2 ] + B, $\boldsymbol{E}$[ -1 ] ]
        end if
    end if
    Apply a left padding of PadLeft to $\boldsymbol{x}$
    Set $\boldsymbol{V}$[ $i$ ] = $\boldsymbol{x}$
    Set $\boldsymbol{E}$[ -1 ] = Len($\boldsymbol{x}$) + PadLeft
    Set $\boldsymbol{E}_1$[ $k$ ] = Len($\boldsymbol{x}$) + PadLeft
    Sort $\boldsymbol{E}$ in increasing order
    Set $\boldsymbol{E}$ = [$\boldsymbol{E}$[ -2 ], $\boldsymbol{E}$[ -1 ]]
end for
Apply right padding to segments in $\boldsymbol{V}$ to match lengths
Sample a separate value of loudness for all segments
Scale all segments for loudness
Create a multitalker mixture by adding all segments together
Sample a noise segment from $\boldsymbol{N}$
Sample a value for noise loudness
Scale the noise segment for loudness and add it to the multitalker mixture
```

Similar to the LibriCSS dataset [28], we generate mixtures in a way that a given mixture can have an arbitrary number of speakers, but at a given time instant, only a maximum of two speakers can overlap. Algo. 1 describes the steps used in generating a sample mixture from $\boldsymbol{S}, \boldsymbol{U}, \boldsymbol{N}$, and $\boldsymbol{P}$. In Algo. 1, Len($\boldsymbol{x}$) represents the length of $\boldsymbol{x}$, Unique(p) denotes the set of unique elements in p, and $\boldsymbol{E}$[-k] denotes the k^{th} element in $\boldsymbol{E}$ from the end.

In Algo. 1, the list $\boldsymbol{E}$ is used to keep track of allowed overlap regions and the set $\boldsymbol{E}_1$ is used to make sure that two different

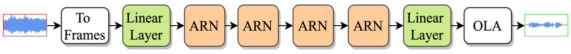


Figure 2: *The model architecture used in this study.*

segments from the same speaker do not overlap. We remove silences from all utterances and then pad zeroes in the beginning to shift a given segment. We use no padding for the first speaker, the second speaker has a minimum padding of A, and the rest of the speakers use padding in a way that a maximum of two speakers overlap at a time.

2.3. DNN Model

We employ a recently proposed attentive recurrent network (ARN) for time-domain speech enhancement [29]. The model architecture is shown in Fig. 2. It comprises an input linear layer followed by four ARN layers and one output linear layer. An input mixture $\boldsymbol{y}$ is first converted to frames $\boldsymbol{Y} \in \mathbb{R}^{T \times L}$, where T is the number of frames and L is the frame size. Next, frames in $\boldsymbol{Y}$ are projected to size D, processed by a stack of four ARN layers, and projected back to size L using the output linear layer. Finally, an overlap-and-add (OLA) is used to get the enhanced waveform. A more detailed description of ARN can be found in [29].

2.4. Loss Function

For attentive training, we use an utterance level SNR loss between the first speaker $\boldsymbol{s}_1$ and its estimate $\hat{\boldsymbol{s}}_1$, defined as

$$\mathrm{L}(\boldsymbol{s}_1, \hat{\boldsymbol{s}}_1) = -10 \cdot \log_{10} \frac{||\boldsymbol{s}_1||^2}{||\boldsymbol{s}_1 - \hat{\boldsymbol{s}}_1||^2} \tag{2}$$

3. Experiments

3.1. Datasets

We create two datasets: one from the WSJ0 corpus [30] and the other from Librispeech [31]. For WSJ0, we use a random split of 80% and 20% from si_tr_s speakers for training and validation. All speakers from si_dt_05 and si_et_05 are used for evaluation. For Librispeech, we use speakers from train-clean-100, dev-clean, and test-clean for training, validation, and evaluation respectively. The Librispeech corpus is more challenging than WSJ0 as it has a larger number of speakers, and a wide variety of acoustic conditions [32].

We use noises from the WHAM! corpus [33]. First, we split training noises into 10-s chunks, and validation and test noises into 15-s chunks. All chunks shorter than 3 seconds are omitted. We use an LKFS [34] based loudness for controlling the SNR. We sample loudness from $[-25, -30]$ dB for speaker segments and from $[-35, -40]$ dB for noise segments.

Table 1: *Attentive training and PIT comparisons for single speaker test set with speaker pattern 1111.*

Dataset	WSJ0			LibriSpeech		
Metric	SI-SNR	PESQ	eSTOI	SI-SNR	PESQ	eSTOI
Mix.	9.6	2.54	76.3	9.5	2.33	72.2
PIT-2	**18.3**	**3.52**	**92.9**	**15.0**	3.06	85.2
AT-2	18.0	**3.52**	92.5	14.9	**3.11**	**85.4**
PIT-3	**17.7**	3.44	91.7	13.1	2.92	82.7
AT-3	17.4	**3.47**	**92.3**	**14.4**	**3.05**	**84.7**

3.2. Experimental settings

All the utterances are resampled to 16 kHz. A frame size of 16 ms, frame shift of 4 ms, and $D = 512$ is used for ARN. ARN uses BLSTMs with 256 hidden units in both directions. We train 2-speaker and 3-speaker models. All the models are trained on interaction patterns with 4 segments. A 2-speaker model is trained on 1 and 2 speakers. A 3-speaker model is trained on 1, 2, and 3 speakers. For a p-speaker PIT model, we use p linear layers at the output, and for an input with k ($k <= p$) speakers, we select the minimum loss assignment from all possible C_k^p assignments.

All training samples are randomly and dynamically generated during training, and an episode of 100k samples is considered as one epoch. We use $p_{overlap} = 0.75, A = 1$ second. B is sampled from [0.25, 0.50] seconds. Segment length, T, is sampled from [2, 3] seconds for training and from [2, 4] seconds for validation and test. We use interaction patterns 1221 and 1231 for validation of 2-speaker and 3-speaker models respectively. We provide our dataset generation script at `https://github.com/ashutosh620/AttentiveTraining`.

All the training utterances longer than 10 seconds are trimmed to 10 seconds. All the models are trained for 100 epochs using the loss in (2) with a batch size of 26 utterances. The learning rate is initialized with 0.0004 and scaled by 0.98 after every two epochs.

All the models are evaluated on interaction patterns from {1111, 1212, 1221, 122221, 1231, 123231, 12341, 123451} and three overlap types: {*Max, Half, None*}. Following Algo. 1, *Max* uses maximum allowed overlap, *Half* uses half of the allowed regions for overlap, and *None* uses no overlap. We generate 3000 evaluation utterances for each combination of the interaction pattern and overlap type. The pattern 1212 is used to assess performance for an alternating pattern of the target and interfering speaker, 1221 is used to assess performance with a larger gap between two consecutive segments of the target speaker. The pattern 122221 is used to assess performance with an even larger gap not used during training. Similarly, patterns 1231 and 123231 are used to assess performance for 3 speakers with different gaps, where 123231 is not used during training. Patterns 12341 and 123451 are used to assess performance for untrained numbers of 4 and 5 speakers.

We use scale-invariant SNR (SI-SNR), extended short-time objective intelligibility (eSTOI) [35], and perceptual evaluation of speech quality (PESQ) [36] as evaluation metrics. Objective scores are computed for the first speaker and eSTOI is reported in percentage.

3.3. Experimental results

Attentive training and PIT are compared in Tables 1 and 2, where PIT-2 and AT-2 are 2-speaker models, and PIT-3 and AT-3 are 3-speaker models. Table 1 shows results for interaction pattern 1111, i.e., test utterances with one speaker. We observe that 2-speaker models are better than 3-speaker models and attentive training and PIT obtain similar scores except for the 3-speaker model on Librispeech where attentive training is better than PIT.

Next, we compare attentive training and PIT for 2 to 5 speakers in Table 2. We notice that a 2-speaker model is better than a 3-speaker model for interaction patterns with 2 speakers, and vice versa. Additionally, for interaction patterns with 4 and 5 speakers, a 3-speaker model is better than a 2-speaker model.

Further, we can see that attentive training performs better than PIT for all the cases. Also, performance improvements are

Table 2: *Attentive training and PIT comparisons for* 2 *to* 5 *speakers. (a) number of speakers, (b) interaction pattern.*

		Dataset			WSJ0									LibriSpeech								
		trained?		Type	*Max*			*Half*			*None*			*Max*			*Half*			*None*		
(a)	(b)	(a)	(b)	Metric	SI-SNR	PESQ	eSTOI	SI-SNR	PESQ	ESTOI	SI-SNR	PESQ	eSTOI	SI-SNR	PESQ	eSTOI	SI-SNR	PESQ	ESTOI	SI-SNR	PESQ	eSTOI
2	1212	-	-	Mix.	-0.6	1.85	51.9	-0.7	2.07	61.6	-1.0	2.48	75.9	-0.6	1.67	51.4	-0.7	1.86	59.5	-1.0	2.28	71.9
		✓	✓	PIT-2	13.8	2.97	84.7	15.3	3.13	88.5	17.2	3.31	93.1	11.7	2.62	76.4	13.1	2.80	81.0	15.4	3.04	87.0
				AT-2	**14.0**	**3.04**	**85.7**	**15.5**	**3.24**	**89.3**	**17.5**	**3.58**	**93.6**	**12.6**	**2.77**	**79.2**	**13.9**	**2.97**	**83.0**	**16.0**	**3.34**	**88.0**
		✓	✓	PIT-3	12.5	2.82	81.7	13.9	2.98	85.7	16.0	3.22	91.4	9.9	2.41	71.7	10.9	2.57	76.0	13.5	2.93	83.3
				AT-3	**12.8**	**2.93**	**83.6**	**14.4**	**3.13**	**87.6**	**16.3**	**3.44**	**92.5**	**11.1**	**2.60**	**75.8**	**12.3**	**2.78**	**79.7**	**14.7**	**3.18**	**85.4**
	1221	-	-	Mix.	-0.6	1.93	51.6	-0.8	2.21	62.4	-1.0	2.45	75.8	-0.6	1.77	51.2	-0.8	2.06	60.1	-1.0	2.31	71.6
		✓	✓	PIT-2	13.7	3.02	84.7	15.5	3.20	88.7	17.4	3.30	93.1	11.6	2.69	76.2	13.3	2.91	81.2	15.4	3.05	86.6
				AT-2	**13.9**	**3.14**	**85.8**	**15.7**	**3.43**	**89.5**	**17.9**	**3.68**	**93.7**	**12.6**	**2.90**	**79.0**	**14.2**	**3.21**	**83.3**	**16.2**	**3.47**	**87.9**
		✓	✓	PIT-3	12.4	2.88	81.7	14.2	3.08	86.1	16.2	3.20	91.1	9.8	2.51	71.5	11.1	2.76	76.4	13.3	2.97	82.6
				AT-3	**12.6**	**3.01**	**83.4**	**14.6**	**3.31**	**87.8**	**16.8**	**3.52**	**92.7**	**11.1**	**2.70**	**75.7**	**12.6**	**3.03**	**80.0**	**14.6**	**3.30**	**85.0**
	122221	-	-	Mix.	-3.6	2.11	51.6	-3.7	2.27	62.8	-3.8	2.42	76.0	-3.7	2.00	51.1	-3.8	2.16	60.2	-3.9	2.32	71.6
		✓	×	PIT-2	**13.6**	**3.06**	84.8	**15.5**	3.16	88.8	17.3	3.20	93.1	11.4	2.80	76.4	12.9	2.93	81.4	14.9	3.02	87.0
				AT-2	**13.6**	**3.36**	**85.7**	15.4	**3.55**	**89.5**	**17.4**	**3.69**	**93.7**	**11.9**	**3.14**	**78.7**	**13.5**	**3.34**	**83.2**	**14.8**	**3.46**	**88.4**
		✓	×	PIT-3	12.4	2.95	81.9	14.3	3.06	86.4	16.4	3.13	91.3	9.9	2.72	71.8	11.3	2.86	77.0	13.6	3.01	83.3
				AT-3	**12.6**	**3.21**	**83.7**	**14.7**	**3.40**	**88.2**	**16.8**	**3.51**	**92.8**	**11.1**	3.05	**76.0**	**12.6**	**3.24**	**80.3**	**14.7**	**3.40**	**85.4**
3	1231	-	-	Mix.	-0.6	2.01	56.3	-0.7	2.24	65.5	-1.0	2.45	75.9	-0.6	1.86	55.3	-0.8	2.08	62.7	-1.0	2.32	71.7
		×	×	PIT-2	6.0	2.66	79.2	5.8	2.85	85.5	8.9	3.10	92.9	5.8	2.48	73.6	5.3	2.65	78.7	8.9	2.91	86.6
				AT-2	**8.3**	**2.88**	**84.4**	**8.4**	**3.09**	**88.8**	**13.6**	**3.50**	**93.6**	**8.3**	**2.73**	**79.4**	**8.4**	**2.94**	**83.4**	**12.9**	**3.32**	**88.3**
		✓	✓	PIT-3	**12.0**	2.85	82.2	14.3	3.07	86.9	15.5	3.20	91.4	9.9	2.55	73.4	11.7	2.79	78.8	13.5	2.98	83.8
				AT-3	11.8	**2.94**	**83.3**	**14.6**	**3.26**	**88.5**	**16.3**	**3.50**	**92.6**	**10.9**	**2.71**	**77.0**	**12.6**	**2.99**	**80.9**	**14.5**	**3.29**	**84.9**
	123231	-	-	Mix.	-3.5	2.10	56.8	-3.6	2.26	65.9	-3.8	2.42	76.1	-3.6	1.98	55.6	-3.7	2.14	62.6	-3.9	2.32	71.6
		×	×	PIT-2	0.8	2.52	79.1	1.0	2.69	85.1	5.9	2.95	92.9	0.4	2.40	72.5	0.3	2.53	77.7	5.0	2.81	86.7
				AT-2	**2.6**	**2.67**	**84.2**	**3.2**	**2.89**	**88.4**	**9.6**	**3.37**	**93.7**	**1.9**	**2.56**	**78.1**	**2.1**	**2.73**	**82.3**	**7.2**	**3.12**	**88.7**
		✓	×	PIT-3	11.6	2.84	82.8	13.5	3.00	87.2	14.8	3.11	91.5	9.5	2.62	74.0	11.1	2.80	78.9	12.9	2.98	84.3
				AT-3	**11.9**	**3.04**	**84.4**	**13.9**	**3.28**	**88.7**	**15.6**	**3.48**	**92.8**	**10.7**	**2.87**	**77.4**	**12.4**	**3.11**	**81.1**	**14.5**	**3.39**	**85.4**
4	12341	-	-	Mix.	-2.3	2.13	59.8	-2.4	2.26	65.9	-2.6	2.43	76.1	-2.4	1.98	57.5	-2.5	2.12	62.6	-2.8	2.31	71.8
		×	×	PIT-2	2.2	2.60	80.6	2.3	2.72	84.7	4.5	2.95	93.1	1.8	2.44	73.7	1.8	2.57	77.7	4.5	2.81	86.4
				AT-2	**4.1**	**2.77**	**85.6**	**4.8**	**2.96**	**88.5**	**9.2**	**3.37**	**93.9**	**3.5**	**2.63**	**79.2**	**4.1**	**2.80**	**82.6**	**8.0**	**3.16**	**88.5**
		×	×	PIT-3	7.9	2.79	82.7	8.4	2.91	85.8	10.4	3.08	91.6	7.2	2.55	73.5	7.3	2.67	76.7	8.8	2.88	83.3
				AT-3	**11.7**	**3.05**	**85.3**	**13.8**	**3.27**	**88.6**	**15.5**	**3.49**	**92.9**	**10.7**	**2.83**	**77.8**	**12.4**	**3.06**	**80.9**	**14.3**	**3.35**	**85.2**
5	123451	-	-	Mix.	-3.5	2.10	56.9	-3.6	2.26	65.7	-3.8	2.42	76.0	-3.6	1.98	55.7	-3.7	2.14	62.8	-4.0	2.32	71.8
		×	×	PIT-2	0.4	2.47	77.5	0.5	2.64	84.2	1.9	2.85	93.2	-0.1	2.37	71.7	-0.1	2.51	77.1	1.7	2.74	86.4
				AT-2	**2.2**	**2.64**	**83.4**	**2.7**	**2.87**	**88.1**	**5.9**	**3.28**	**93.8**	**1.4**	**2.52**	**77.3**	**1.7**	**2.70**	**82.0**	**3.5**	**2.97**	**88.7**
		×	×	PIT-3	4.7	2.64	79.4	5.0	2.80	84.5	6.9	2.98	91.7	4.7	2.47	71.3	4.6	2.61	75.8	5.5	2.80	82.8
				AT-3	**10.8**	**3.00**	**84.3**	**13.2**	**3.26**	**88.6**	**14.6**	**3.46**	**92.9**	**10.1**	**2.84**	**77.1**	**12.2**	**3.10**	**81.1**	**14.1**	**3.38**	**85.4**

higher for the difficult Librispeech dataset, and for less overlap cases, such as *Half* and *None*. Most impressive improvements are observed in PESQ scores. Worth mentioning is that for the pattern 122221, which is not used during training, we observe larger improvements compared to other 2-speaker patterns. Similarly, with a 3-speaker model, larger improvements are observed for the pattern 123231 compared to 1231. This indicates that attentive training is helpful in improving generalization for inputs with a trained number of speakers but an untrained interaction pattern.

We notice that AT-3 is the only model that obtains strong results for all the cases. PIT-3 is much worse than AT-3 for 4 and 5 speakers. For example, on WSJ0 with *Max* overlap, AT-3 obtains an average SI-SNR of 10.8 dB and PESQ of 3.00 for the 5-speaker pattern 123451, but PIT-3 obtains a much worse lower SI-SNR of 4.7 dB and PESQ of 2.64. We also notice that AT-2 and PIT-2 do not generalize well to 3 or more speakers. Even though better than PIT-2, AT-2 is far worse that AT-3 for 3 and more speakers. We expect AT-2 to generalize to an untrained number of speakers as it is trained to treat the first speaker as the target and the rest of the signal as the interference. We believe that the unexpected behavior from AT-2 to not generalize may be caused by limited interaction patterns used in AT-2 training. It uses patterns with only 1 or 2 speakers during training, and as a result, learns to preserve the first speaker and suppress the second one, not the expected behavior of preserving the first speaker and suppressing the rest. AT-3 on the other hand learns to extract the first speaker and suppress the remaining speakers.

In summary, proposed attentive training leads to a robust DNN model that generalizes to untrained test conditions in terms of number of speakers, speaker interaction patterns, and amount of speaker overlaps. Also, improvements over PIT in mismatched conditions are impressive.

Table 3: *Effect of reducing the onset difference between the first and the second speaker.*

Onset diff.	1 s		0.75 s		0.5 s		0.25 s	
Model	PIT-2	AT-2	PIT-2	AT-2	PIT-2	AT-2	PIT-2	AT-2
SI-SNR	13.8	13.9	13.8	13.7	13.7	13.2	13.6	11.6
PESQ	2.97	3.04	2.97	3.02	2.97	2.99	2.96	2.88
ESTOI	84.7	85.6	84.7	85.2	84.6	84.2	84.7	81.7

Finally, we analyze the effect of reducing the onset difference between the first and the second speaker. We modify the onset difference of the test set with the interaction pattern of 1212. Table 3 compares AT-2 and PIT-2 on WSJ. We observe that even though AT-2 is trained with a minimum onset difference of 1 second, it obtains a similar performance for lower onset differences of 0.75 and 0.5 seconds. A considerable drop is observed only when the onset difference is reduced to a small value of 0.25 seconds. Moreover, AT-2 and PIT-2 are comparable up to an onset difference of 0.5 seconds.

4. Conclusion

We have proposed a novel attentive training framework for talker-independent speaker extraction. The proposed framework has an intrinsic mechanism for speaker selection. Experimental results have demonstrated the superiority of attentive training over permutation invariant training, especially in mismatched test conditions. Future work includes evaluating the proposed framework for multitalker speaker separation and speaker diarization.

5. References

[1] E. C. Cherry, "Some experiments on the recognition of speech, with one and with two ears," *The Journal of the Acoustical Society of America*, vol. 25, no. 5, pp. 975–979, 1953.

[2] D. Broadbend, "Perception and communication," *Pergamon Press, New York*, 1958.

[3] D. L. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 1702–1726, 2018.

[4] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *ICASSP*, 2016, pp. 31–35.

[5] 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 Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1901–1913, 2017.

[6] Y. Liu and D. Wang, "Divide and conquer: A deep CASA approach to talker-independent monaural speaker separation," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 27, no. 12, pp. 2092–2102, 2019.

[7] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing ideal time-frequency magnitude masking for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 1256–1266, 2019.

[8] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path RNN: Efficient long sequence modeling for time-domain single-channel speech separation," in *ICASSP*, 2020, pp. 46–50.

[9] J. Chen, Q. Mao, and D. Liu, "Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation," *arXiv:2007.13975*, 2020.

[10] N. Zeghidour and D. Grangier, "Wavesplit: End-to-end speech separation by speaker clustering," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2840–2849, 2021.

[11] M. Delcroix, K. Zmolikova, K. Kinoshita, A. Ogawa, and T. Nakatani, "Single channel target speaker extraction and recognition with speaker beam," in *ICASSP*, 2018, pp. 5554–5558.

[12] Q. Wang, H. Muckenhirn, K. Wilson, P. Sridhar, Z. Wu, J. Hershey, R. A. Saurous, R. J. Weiss, Y. Jia, and I. L. Moreno, "Voicefilter: Targeted voice separation by speaker-conditioned spectrogram masking," *arXiv:1810.04826*, 2018.

[13] C. Xu, W. Rao, E. S. Chng, and H. Li, "Spex: Multi-scale time domain speaker extraction network," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 28, pp. 1370–1384, 2020.

[14] T. Li, Q. Lin, Y. Bao, and M. Li, "Atss-net: Target speaker separation via attention-based neural network," *arXiv:2005.09200*, 2020.

[15] Z. Zhang, B. He, and Z. Zhang, "X-tasnet: Robust and accurate time-domain speaker extraction network," *arXiv:2010.12766*, 2020.

[16] W. Wang, C. Xu, M. Ge, and H. Li, "Neural speaker extraction with speaker-speech cross-attention network," in *INTERSPEECH*, 2021, pp. 3535–3539.

[17] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein, "Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation," *arXiv:1804.03619*, 2018.

[18] T. Afouras, J. S. Chung, and A. Zisserman, "The conversation: Deep audio-visual speech enhancement," *arXiv:1804.04121*, 2018.

[19] C. Li and Y. Qian, "Listen, watch and understand at the cocktail party: Audio-visual-contextual speech separation." in *INTERSPEECH*, 2020, pp. 1426–1430.

[20] R. Gu, L. Chen, S.-X. Zhang, J. Zheng, Y. Xu, M. Yu, D. Su, Y. Zou, and D. Yu, "Neural spatial filter: Target speaker speech separation assisted with directional information." in *INTERSPEECH*, 2019, pp. 4290–4294.

[21] A. Brendel, T. Haubner, and W. Kellermann, "A unified probabilistic view on spatially informed source separation and extraction based on independent vector analysis," *IEEE Transactions on Signal Processing*, vol. 68, pp. 3545–3558, 2020.

[22] M. Delcroix, K. Zmolikova, T. Ochiai, K. Kinoshita, and T. Nakatani, "Speaker activity driven neural speech extraction," in *ICASSP*, 2021, pp. 6099–6103.

[23] Y. Hao, J. Xu, P. Zhang, and B. Xu, "WASE: Learning when to attend for speaker extraction in cocktail party environments," in *ICASSP*. IEEE, 2021, pp. 6104–6108.

[24] A. S. Bregman, *Auditory scene analysis: The perceptual organization of sound*. MIT press, 1994.

[25] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive psychology*, vol. 12, no. 1, pp. 97–136, 1980.

[26] N. Kanda, Y. Gaur, X. Wang, Z. Meng, and T. Yoshioka, "Serialized output training for end-to-end overlapped speech recognition," *arXiv:2003.12687*, 2020.

[27] T. von Neumann, K. Kinoshita, M. Delcroix, S. Araki, T. Nakatani, and R. Haeb-Umbach, "All-neural online source separation, counting, and diarization for meeting analysis," in *ICASSP*, 2019, pp. 91–95.

[28] Z. Chen, T. Yoshioka, L. Lu, T. Zhou, Z. Meng, Y. Luo, J. Wu, X. Xiao, and J. Li, "Continuous speech separation: Dataset and analysis," in *ICASSP*, 2020, pp. 7284–7288.

[29] A. Pandey and D. Wang, "Self-attending RNN for speech enhancement to improve cross-corpus generalization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1374–1385, 2022.

[30] D. B. Paul and J. M. Baker, "The design for the wall street journal-based CSR corpus," in *Workshop on Speech and Natural Language*, 1992, pp. 357–362.

[31] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*, 2015, pp. 5206–5210.

[32] A. Pandey and D. Wang, "On cross-corpus generalization of deep learning based speech enhancement," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 28, pp. 2489–2499, 2020.

[33] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. L. Roux, "WHAM!: Extending speech separation to noisy environments," *arXiv:1907.01160*, 2019.

[34] E. Grimm, R. Van Everdingen, and M. Schöpping, "Toward a recommendation for a european standard of peak and LKFS loudness levels," *SMPTE motion imaging journal*, vol. 119, no. 3, pp. 28–34, 2010.

[35] J. Jensen and C. H. Taal, "An algorithm for predicting the intelligibility of speech masked by modulated noise maskers," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 2009–2022, 2016.

[36] 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 *ICASSP*, 2001, pp. 749–752.