Published in final edited form as:
IEEE/ACM Trans Audio Speech Lang Process. 2023 ; 31: 1360–1370. doi:10.1109/taslp.2023.3260711.

Attentive Training: A New Training Framework for Speech Enhancement

Ashutosh Pandey [Student Member, IEEE],
Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210 USA

DeLiang Wang [Fellow, IEEE]
Department of Computer Science and Engineering and the Center for Cognitive and Brain Sciences, The Ohio State University, Columbus, OH 43210 USA

Abstract

Dealing with speech interference in a speech enhancement system requires either speaker separation or target speaker extraction. Speaker separation has multiple output streams with arbitrary assignments while target speaker extraction requires additional cueing for speaker selection. Both of these are not suitable for a standalone speech enhancement system with one output stream. In this study, we propose a novel training framework, called *Attentive Training*, to extend speech enhancement to deal with speech interruptions. Attentive training is based on the observation that, in the real world, multiple talkers very unlikely start speaking at the same time, and therefore, a deep neural network can be trained to create a representation of the first speaker and utilize it to attend to or track that speaker in a multitalker noisy mixture. We present experimental results and comparisons to demonstrate the effectiveness of attentive training for speech enhancement.

Index Terms—

speech enhancement; speaker extraction; speaker separation; talker-independent; attentive training

I. Introduction

Speech signals in the real world are degraded by acoustic interferences, such as background noise, interfering talkers, and room reverberation. Acoustic interferences degrade the intelligibility and quality of speech for both human and machine listeners. For example, the performance of speech based applications, such as automatic speech recognition (ASR), hearing aids, and telecommunications, deteriorates when dealing with degraded speech. Speech enhancement aims at improving the intelligibility and quality of a degraded signal by removing acoustic interference from it. Monaural speech enhancement utilizes recordings from a single microphone to provide a versatile and cost efficient solution to the problem.

pandey.99@osu.edu .

This study is focused on monaural speech enhancement that can deal with both speech and nonspeech interference.

Speech enhancement has been widely studied in the signal processing community for decades. Some of the traditional methods include spectral subtraction, Wiener filtering and statistical-model-based methods [1]. The rise of deep learning and its application to speech enhancement has led to dramatic advances over the last decade, and it is firmly established as the mainstream methodology today [2].

Popular approaches to speech enhancement utilize time-frequency representations, such as short-time Fourier transform (STFT), to represent input features and training targets, and aim at enhancing only the spectral magnitude [3], [4], [5], [6], [7], [8], [9], [10], [11]. A recent trend has been to jointly enhance the spectral magnitude and phase by using either complex spectrogram enhancement [12], [13], [14], [15], [16], [17], [18], [19], [20] or time-domain speech enhancement [21], [22], [23], [24], [25], [26], [27], [28], [29].

Speech enhancement is generally formulated as the problem of removing nonspeech interferences from a speech signal. However, in the real world, interfering signals can also be speech from interfering talkers. How to deal with interfering talkers in a speech enhancement system? Dealing with interfering talkers requires two steps: speaker selection and speaker extraction. Human listeners have the amazing ability of auditory perception attending to (hence extracting) a single speaker in a multitalker scenario. This ability is widely referred to as the *cocktail party effect* [30], and has inspired the perceptual theory of selective attention [31]. For humans, speaker selection is dependent on listener attention as well as intention. For machine separation so far, we either separate all speakers from a mixture or provide a cueing signal for speaker selection followed by speaker extraction. The former is called speaker separation and the latter is commonly known as target speaker extraction.

Speaker separation is the task of reconstructing all the speakers from a multitalker mixture. Early works on speaker separation were extended from speech enhancement and talker-dependent, i.e., systems that extract speech signals from only a given speaker and cannot generalize to untrained speakers. When extending to talker-independent speaker separation, these models suffer from a well known permutation ambiguity problem, where a DNN is not able to consistently assign output streams to different speakers during training. Deep clustering [32] and permutation invariant training (PIT) [33] are two representative approaches to resolving the permutation ambiguity problem. Deep clustering and its variants [34] employ a DNN to map each T-F unit of the input mixture to an embedding space, where embeddings are trained to be closer for the units corresponding to a single speaker and far for different speakers. Finally, embedding vectors are clustered into groups corresponding to the different speakers in the mixture to obtain a T-F mask for each speaker. In contrast, PIT allows for end-to-end optimization to separate speech signals by dynamically assigning the best matching permutation of the ground-truth signals with the output signals. In particular, the simplicity of PIT has led to many subsequent models for speaker separation [35], [36], [37], [38], [39].

Speaker separation can separate all underlying speakers but it assigns output streams arbitrarily, which is not suitable for speech enhancement systems that need to attend to one output stream. For example, if we design a system that always picks a fixed output stream, it will correspond to either silence or sporadic interruptions when the main speech stream goes to one of the other outputs.

Target speaker extraction is the task of extracting a single speaker from a multitalker mixture, where the target speaker is cued using additional information in the form of audio [40], [41], [42], [43], [44], [45] or image [46], [47], [48]. Recent studies have also explored other kinds of cues, such as spatial location [49], [50], speech activity [51], and onset [52]. Target speaker extraction is similar to auditory selective attention, but requires a priori cueing that may not be available in many applications of speech enhancement.

How to extend a speech enhancement system to deal with speech interruptions without requiring speaker cues? This requires designing an intrinsic speaker selection mechanism. Attention is a major part of perception, and this has inspired us to leverage auditory selective attention to address the problem. If a person is listening (attending) to a talker, he would typically continue listening to that talker irrespective of other speech interruptions, particularly when the interruptions are short. Based on this, we propose a new training framework, which we name *attentive training*, for speech enhancement. In 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 [53]. Therefore, we can assume that a given multitalker mixture has nonoverlapping speech intervals at the beginning. With attentive training, a model presented with a multitalker mixture will start attending to (extracting) nonoverlapping speech segments in the beginning and then continue attending to it while ignoring other speakers. In other words, attentive training treats the speech signals of the first speaker as *target speech*, and the utterances of other speakers plus environmental sounds as *background interference*.

The attentive training framework is consistent with the dominant feature integration theory of attention [54]. According to this perceptual account, attention serves to integrate perceptual features extracted in separate analyses into an object. The attended object forms the target (or foreground), and the remaining objects in a scene become the background. Furthermore learning and attending are integral parts of perception.

Note that attentive training uses the onset of the first speaker as a cue for intrinsic speaker selection. In the context of ASR, a similar idea of using speaker onsets as a cue has been proposed in serialized output training (SOT) [55]. The idea of SOT is to output speaker transcriptions from an ASR system in the order of speaker onsets in the input mixture. The proposed attentive training is fundamentally different from SOT as it is designed for a speech enhancement system that aims at extracting only the first speaker from a mixture.

We create a multitalker dataset in a controlled way, where the first speaker is set to start slightly ahead of the rest of the speakers. Next, we train a recently proposed time-domain attentive recurrent network (ARN) [29] with attentive training to estimate the first speaker from a multitalker mixture. We show that ARN is effective in extracting the first speaker and generalizes well to different test conditions, such as an untrained number of speakers, mixtures with larger gaps between the consecutive segments of the target speaker, and smaller speaker overlaps. For instance, a model trained using mixtures with a maximum of 3 speakers obtains strong results for mixtures with 5 speakers.

We compare attentive training with PIT for speaker separation. We find that attentive training obtains substantially better results than PIT when compared on the enhancement metric of the first speaker. We also investigate a decoupled approach to attentive training in which the nonoverlapping speech of the first speaker in the beginning of a mixture is used to create a speaker representation to be used as a cue for target speaker extraction. We observe that end-to-end attentive training obtains better results than decoupled attentive training. We also train a target speaker extraction model using independent enrollment utterances. We find that target speaker extraction with independent enrollment utterances performs slightly better than attentive training. When contrasting target speaker extraction and decoupled attentive training, we conclude that target speaker extraction is slightly better than attentive training only because of additional information in the form of clean enrollment utterances.

We also examine an attentive training model trained with onset differences of more than 1 second between the first and the second speaker, and show that it generalizes well to an onset difference of 0.5 seconds. Also, we train speaker verification systems on top of the hidden layers in ARN to demonstrate that a few of them encode speaker information, which verifies that ARN learns speaker representations implicitly for selection and extraction.

Along the way 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, 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 [39]. 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.

This study focuses on extracting the first speaker from a mixture to illustrate the effectiveness of attentive training. A straightforward and useful extension of attentive training would be to develop a speech enhancement system that aims at removing interfering speech only from the interval of speaker overlaps. The preserved speaker should be the one that enters into the overlapping interval from the past. It will reduce to a speech enhancement system handling nonoverlapping speech signals from multiple speakers in the output stream. Designing such a system will require a careful consideration into fixing hyperparameters, such as gaps between consecutive segments of different speakers, to output a perceptually meaningful signal.

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 [56].

A preliminary study on attentive training has been published in [57] where a smaller ARN is trained on a smaller dataset and compared only with speaker separation using PIT. The remainder of the paper is organized as follows. A definition and different methods of attentive training are discussed in Section II. Section III describes the data generation algorithm. Section IV] details employed DNN models. Experimental settings are given in Section V and results and comparisons are presented in Section VI. Concluding remarks are given in Section VII.

II. Speaker Tracking and Attentive Training

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 $\{\boldsymbol{y}, \boldsymbol{s}_i\} \in \mathbb{R}^{N \times 1}, C$ is the total number of speakers, $\boldsymbol{s}_i$ is the i^{th} speaker, and $\boldsymbol{n}$ is the background noise. Let o_i denote the time sample when i^{th} speaker starts speaking. We assume that speaker indices $i = 1, 2, \ldots, C$ are sorted in the increasing order of onset times. In other words, $i < j$ implies $o_i < o_j$. The goal of attentive training is to separate the first speaker $\boldsymbol{s}_1$ from $\boldsymbol{y}$.

We can extract the first speaker from a mixture using the following methods.

A. Speaker Separation

A speaker separation system has no selection mechanism and reconstructs all the speakers in a mixture. Speaker separation can be utilized to extract the first speaker by first separating all the speakers and then selecting the first speaker using speech onset. Speaker separation is illustrated in Fig. 1a, and modeled as

$$[\hat{\boldsymbol{s}}_1, \cdots, \hat{\boldsymbol{s}}_C] = f_{SS}(\boldsymbol{y}) \tag{2}$$

where f_{SS} represents a DNN for speaker separation. The speaker separation model is trained using an utterance-level PIT loss defined as

$$\mathscr{L} = \sum_{i=1}^{C} \mathscr{D}\left(\boldsymbol{s}_{\phi^*(i)}, \hat{\boldsymbol{s}}_i\right) \tag{3}$$

where $\mathscr{D}(\boldsymbol{a}, \boldsymbol{b})$ is a distance measure between signals $\boldsymbol{a}$ and $\boldsymbol{b}$ and ϕ^* is a permutation of target signals with the minimum cost, i.e.,

$$\phi^* = \arg\min_{\mathscr{P}} \sum_{i=1}^{C} \mathscr{D}\left(\boldsymbol{s}_{\phi^*(i)}, \hat{\boldsymbol{s}}_i\right) \tag{4}$$

where $\mathscr{P}$ represents the set of all possible permutations. We use an utterance-level negative signal-to-noise ratio (SNR) as the distance measure, defined as

$$\mathscr{D}\left(\boldsymbol{s}, \hat{\boldsymbol{s}}\right) = -10 \cdot \log_{10} \frac{\| \boldsymbol{s} \|^2}{\| \boldsymbol{s} - \hat{\boldsymbol{s}} \|^2} \tag{5}$$

B. Target Speaker Extraction

Target speaker extraction extracts a single speaker from a mixture with the help of an additional cue for target selection. The speaker selection mechanism is not intrinsic to model training. We assume that we are given additional information in the form of an enrollment utterance $\boldsymbol{e}_1$ corresponding to the first speaker. Target speaker extraction is illustrated in Fig. 1b. First, a speaker embedding is computed from $\boldsymbol{e}_1$ as

$$\boldsymbol{v}_1 = h(\boldsymbol{e}_1) \tag{6}$$

where $\boldsymbol{v}_1 \in \mathbb{R}^{B \times 1}$, $\boldsymbol{B}$ is the size of the embedding vector, and h is a DNN-based speaker embedding model. Next, $\boldsymbol{v}_1$ and $\boldsymbol{y}$ are used together to estimate $\boldsymbol{s}_1$ as

$$\hat{\boldsymbol{s}}_1 = f_{TSE}(\boldsymbol{y}, \boldsymbol{v}_1) \tag{7}$$

where f_{TSE} represents a DNN for target speaker extraction. It is trained using a distance between the estimated and the ground-truth signal of the first speaker as defined below.

$$\mathscr{L} = \mathscr{D}(\boldsymbol{s}_1, \hat{\boldsymbol{s}}_1) \tag{8}$$

C. Attentive Training

Attentive training aims at estimating $\boldsymbol{s}_1$ directly from $\boldsymbol{y}$ as shown in Fig. 1c. It is defined as

$$\hat{\boldsymbol{s}}_1 = f_{AT}(\boldsymbol{y}) \tag{9}$$

where f_{AT} represents a DNN for attentive training. It is trained using the loss in Eq. 8.

C. Decoupled Attentive Training

Decoupled attentive training decouples end-to-end attentive training in two parts. First, it assumes that we are provided with the nonoverlapping speech segment $\boldsymbol{s}_1^{no}$ in the beginning of $\boldsymbol{y}$, defined as

$$\boldsymbol{s}_1^{no} = \boldsymbol{y}[0 : M - 1] = \boldsymbol{s}_1[0 : M - 1] + \boldsymbol{n}[0 : M - 1] \tag{10}$$

where M is the length of $\boldsymbol{s}_1^{no}$. Next, $\boldsymbol{s}_1^{no}$ is used to generate a speaker embedding of the first speaker

$$\boldsymbol{v}_1^{no} = h(\boldsymbol{s}_1^{no}) \tag{11}$$

Finally, $\boldsymbol{v}_1^{no}$ and $\boldsymbol{y}$ are used together to estimate $\boldsymbol{s}_1$ as

$$\hat{\boldsymbol{s}}_1 = f_{De-AT}(\boldsymbol{y}, \boldsymbol{v}_1^{no}) \tag{12}$$

where f_{De-AT} represents a DNN for decoupled attentive training. Fig. 1d depicts decoupled attentive training. The loss in Eq. 8 is used to train a decoupled attentive training model.

III. Data Generation

This section describes our technique for generating multitalker mixtures. Given a set $\boldsymbol{S} = \{S_1, \ldots, S_J\}$ of speakers, their corresponding utterances $\boldsymbol{U}^{S_j} = \left\{\boldsymbol{s}_1^j, \ldots, \boldsymbol{s}_{Q_j}^j\right\}$, and a set of noise segments $\boldsymbol{N} = \{\boldsymbol{n}_1, \ldots, \boldsymbol{n}_R\}$, where Q_j denotes the number of utterances of speaker S_j and R is the number of noise segments, we create a multitaker noisy 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. Based on this, we define a concept called *interaction pattern* representing the order of speaker segments in a mixture. For example, an interaction pattern of 1212 represents a mixture created by adding 4 segments sorted in the 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 initial 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. 2. For data generation, we use interaction patterns from a predefined set $\boldsymbol{P} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_P\}$.

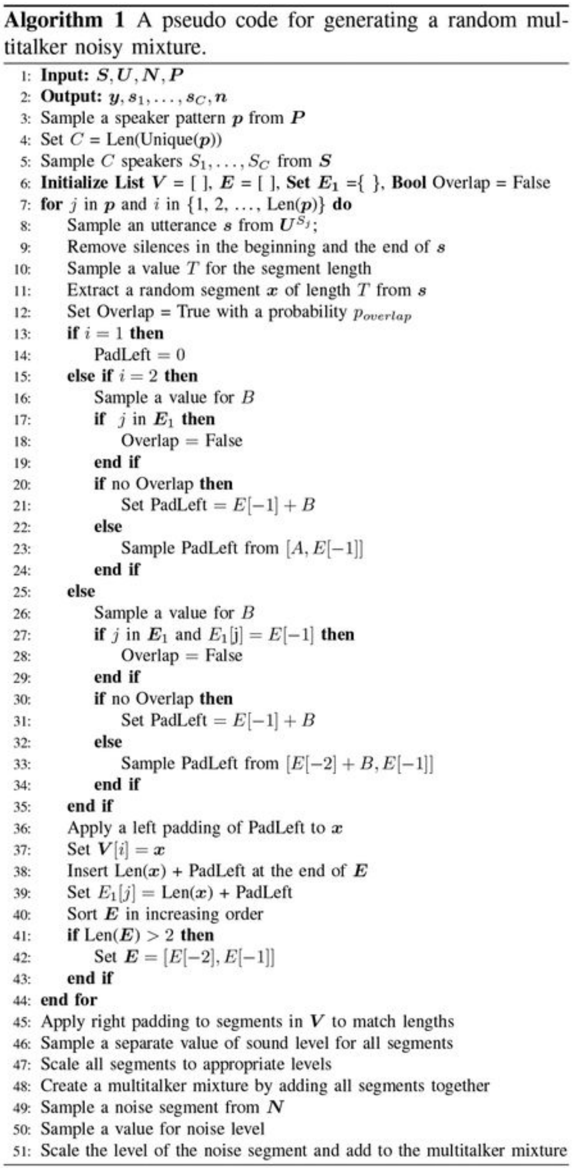
Similar to the LibriCSS dataset [58], we generate mixtures in a way that a given mixture can have an arbitrary number of speakers, but at a given time instant, only a maximum of two speakers can overlap. Algorithm (Algo.) 1 describes the steps used in generating a sample mixture from $\boldsymbol{S}, \boldsymbol{U}, \boldsymbol{N}$, and $\boldsymbol{P}$. In the algorithm, $\mathrm{Len}(\boldsymbol{x})$ represents the length of $\boldsymbol{x}$, and Unique $(\boldsymbol{p})$ denotes the set of unique elements in $\boldsymbol{p}$.

In Algo. 1, the list $\boldsymbol{E}$ is used to keep track of allowed overlap intervals and $E[-k]$ denotes the k^{th} element in $\boldsymbol{E}$ from the end. The allowed interval spans from $E[-2]$ to $E[-1]$, which indicate the ending time samples of the last two segments. The set $\boldsymbol{E}_1$ is used to make sure that two different segments from the same speaker do not overlap (line 27 in Algo. 1). 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 remaining speakers use zero padding in a way that a maximum of two speakers overlaps at a time.

IV. DNN Models

We employ a recently proposed ARN model for time-domain speech enhancement [29]. The model architecture is shown in Fig. 3. It comprises an input linear layer followed by four ARN layers and an 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. An ARN layer comprises an RNN block, a feedforward block, and an attention block. A more detailed description of these blocks can be found in [29]. For speaker separation, we use C linear layers at the output. For decoupled attentive training and target speaker extraction, we utilize a strong speaker embedding model called ECAPA-TDNN [59], which is shown in Fig. 4. The output from ECAPA-TDNN is projected to size D using a linear layer and then multiplied elementwise to the output of the second ARN. We also investigated multiplying to the output of other or all ARNs but observed worse results. We utilize a pretrained ECAPA-TDNN model provided in the SpeechBrain toolkit [60] as it exhibits strong speaker verification performance.



Algorithm 1 A pseudo code for generating a random multitalker noisy 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 $\boldsymbol{p}$ from $\boldsymbol{P}$
Set $C$ = Len(Unique($\boldsymbol{p}$))
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 $j$ in $\boldsymbol{p}$ and $i$ in {1, 2, ..., Len($\boldsymbol{p}$)} do
    Sample an utterance $\boldsymbol{s}$ from $\boldsymbol{U}^{S_j}$;
    Remove silences in the beginning and the end of $\boldsymbol{s}$
    Sample a value $T$ for the 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  $j$ in $\boldsymbol{E}_1$ then
            Overlap $=$ False
        end if
        if no Overlap then
            Set PadLeft $= E[-1] + B$
        else
            Sample PadLeft from $[A, E[-1]]$
        end if
    else
        Sample a value for $B$
        if $j$ in $\boldsymbol{E}_1$ and $E_1$[j] $= E[-1]$ then
            Overlap $=$ False
        end if
        if no Overlap then
            Set PadLeft $= E[-1] + B$
        else
            Sample PadLeft from $[E[-2] + B, E[-1]]$
        end if
    end if
    Apply a left padding of PadLeft to $\boldsymbol{x}$
    Set $\boldsymbol{V}[i] = \boldsymbol{x}$
    Insert Len($\boldsymbol{x}$) + PadLeft at the end of $\boldsymbol{E}$
    Set $E_1[j]$ = Len($\boldsymbol{x}$) + PadLeft
    Sort $\boldsymbol{E}$ in increasing order
    if Len($\boldsymbol{E}$) $> 2$ then
        Set $\boldsymbol{E} = [E[-2], E[-1]]$
    end if
end for
Apply right padding to segments in $\boldsymbol{V}$ to match lengths
Sample a separate value of sound level for all segments
Scale all segments to appropriate levels
Create a multitalker mixture by adding all segments together
Sample a noise segment from $\boldsymbol{N}$
Sample a value for noise level
Scale the level of the noise segment and add to the multitalker mixture
```

V. Experimental Settings

A. Datasets

We generate training and evaluation data from the LibriSpeech corpus [61]. We use all the speakers from train-clean-100, train-clean-360, and train-other-500 for training. All the speakers from test-clean and dev-clean are used respectively for testing and validation. The training set consists of 960 hours of speech data, which is much larger than the set of 100 hours used in the preliminary study [57].

Noises used are from the WHAM! corpus [62]. 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 LKFS based loudness [63] for controlling the SNR. We sample sound levels from $[-25, -30]\mathrm{dB}$ for speaker segments and from $[-35, -40]\mathrm{dB}$ for noise segments. We provide our dataset generation script along with the test and validation metadata files at https://github.com/ashutosh620/AttentiveTraining

For target speaker extraction, each multitalker mixture is paired with a randomly sampled enrollment utterance of the first speaker in the mixture. We trim silences from the beginning and the end of an enrollment utterance and truncate longer utterances to a length of 4 seconds.

We also train and evaluate speaker verification systems to examine the speaker information encoded in the hidden layers of the ARN model trained with attentive training. For this, we create a speaker verification dataset using speech from LibriSpeech and noises from WHAM! as in other experiments. We generate training data dynamically by randomly sampling a speech utterance and mixing it with a randomly sampled noise segment. For test and validation, we randomly sample a list of 10000 pairs of noisy speech utterances from different speakers. We sample positive and negative speaker pairs with equal probability.

B. Training Methodology

All the utterances are resampled to 16 kHz. A frame size of $16\ ms$, frame shift of 4 ms, and $D = 1024$ is used for ARN. A smaller ARN model with $D = 512$ was used in the preliminary study [57]. ARN uses BLSTMs with 512 hidden units in both directions. All the models are trained [60] as it exhibits strong speaker verification performance. on interaction patterns with 4 segments with a maximum of 3 speakers. In other words, a randomly generated multitalker mixture contains either 1, 2, or 3 speakers. For the PIT model, we use 3 linear layers at the output, and for an input with K $(K < \ = 3)$ speakers, we select the minimum loss assignment from all possible C_K^3 assignments.

All the training samples are randomly and dynamically generated during training, and an episode of 281k samples (total number of speech utterances) is considered as one epoch. We use $p_{overlap} = 0.75$ and $A = 1$ 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. The input and the output are scaled by 25.

All the training utterances longer than 10 seconds are trimmed to 10 seconds. All the models are trained for 100 epochs with a batch size of 32 utterances using the Adam optimizer [64]. The learning rate is initialized with 0.0004 and scaled by 0.98 every two epochs.

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 the 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 the interaction pattern 1231 with *Max* overlap for validation.

The ECAPA-TDNN model is trained using a set of 7.2k speakers from the VoxCeleb1 [65] and VoxCeleb2 [66] corpora. Data augmentation techniques, such as additive noise, room reverberation, speed perturbation, and SpecAugment [67] are also utilized. An additive angular margin loss with a margin of 0.2 and scale of 30 is used [68], [69]. A more detailed description can be found in the SpeechBrain toolkit [60].

We also evaluate attentive training for sporadic speech interruptions, which occur often in daily environments. For this, we generate a test dataset with longer interaction patterns from {1211111, 1112111, 1111121}. Each of these patterns comprises 7 segments, 6 of which correspond to the target speaker and 1 corresponds to the interfering speaker.

For training speaker verification systems on top of the hidden layers of the pretrained ARN model, we use a 1-layered ARN model with $D = 256$ followed by a statistical pooling borrowed from ECAPA-TDNN. The pooling layer uses 128 channels for attention [59], [60]. The embedding size is set to 32. A batch of training data comprises 32 pairs of 3 seconds long utterances, where a pair consists of one noisy and one clean utterance from the same speaker. We trim silences from the beginning and the end. All models are trained with a cyclical learning rate varying between 0.00004 and 0.0004 using the triangular policy as described in [70] in conjunction with the Adam optimizer [64].

We develop all the models in PyTorch [71] and exploit automatic mixed precision training to expedite training [72]. Two NVIDIA Volta V100 32GB GPUs are utilized to train all attentive training models.

We use scale-invariant SNR (SI-SNR), extended short-time objective intelligibility (eSTOI) [73], and perceptual evaluation of speech quality (PESQ) [74] as evaluation metrics. Objective scores are computed for the first speaker and eSTOI is reported in percentage.

C. Baseline Models

We also evaluate the effectiveness of attentive training for two widely used models for speaker separation: convolutional time-domain audio separation network (Conv-TasNet) [36] and dual-path recurrent neural network (DPRNN) [37]. We train these models using the four methods shown in Fig. 1. We modify these models to use one output stream for AT, De-AT and TSE, and 3 output streams for speaker separation. For Conv-TasNet, we utilize the best performing model in [36] which uses $R = 3$ repeats of $X = 8$ convolutional blocks. For De-AT and TSE, speaker embeddings are fused after the first repeat using elementwise multiplication. Similarly, we utilize the best performing DPRNN architecture in [37], which uses a stack of 6 dual-path blocks including intra-chunk and intra-chunk RNN. To train DPRNN for De-AT and TSE, we fuse speaker embeddings after the third dual-path block using elementwise multiplication. We also train a time-domain model called SpEx+ proposed specifically for TSE [75].

VI. Results and Comparisons

We denote speech enhancement as SE, attentive training as AT, speaker separation as PIT, decoupled attentive training as De-AT, and target speaker extraction as TSE in the results. A speech enhancement model is trained only on the interaction pattern 1111, i.e., single-talker utterances with background noise. Background noise is present in all of the following evaluations.

A. Comparing Different Methods

We start by comparing different methods for the interaction pattern 1111. Results are given in Table I We observe that SE is the best, PIT is the worst, and AT, De-AT, and TSE obtain similar results. We expect SE to obtain best results for this case as it is trained specifically for the matched interaction pattern of 1111. This result suggests that a model capable of dealing with interfering speech performs worse at removing noise than a model trained specifically for removing noise. In other words, the capability of handling interfering speech comes at the expense of noise removal.

Next, we compare different methods for the multitalker case with 2 and 3 speakers and the trained number of speakers. Results are given in Table II. We can observe that a general order of performance among different methods is PIT < De-AT < AT < TSE. In particular, the performance of PIT is far worse than the other methods for all the cases. This highlights a major issue with PIT when dealing with a varying number of speakers and varying degrees of overlaps [58], [39]. We also observe that AT is similar or better than De-AT. This is encouraging because it implies that end-to-end training can better learn the joint task of speaker selection and tracking than a decoupled approach. As expected, TSE obtains the best results since it is provided with additional cueing in the form of an enrollment utterance. It is worth mentioning that De-AT also uses cueing, but the cueing signal comes from the input mixture itself, and hence, it does not provide additional information on top of the input. Also, TSE uses a clean cueing signal in contrast to a noisy one in De-AT.

Finally, we present evaluation result for the untrained numbers of 4 and 5 speakers in Table III We observe similar performance trends to 2 and 3 speakers except for PIT which is much worse because it is not designed to separate the number of speakers not used during training. It is worth noting that AT obtains an SNR improvement of around 15 dB on higher numbers of untrained speakers. This implies that AT does not require training with more than 3 speakers to obtain good generalization.

We plot spectrograms of a sample multitalker mixture enhanced using different methods in Fig. 5. Notice that not only PIT introduces leakage from interfering talkers in the silence intervals but also removes high-frequency speech components. Plots of AT and TSE look very similar with much reduced leakage and well-retained high-frequency components.

B. Comparison with Baselines

Fig. 6 plots the performance of Conv-TasNet, DPRNN, ARN and SpEx+ on interaction pattern 123231. First, we observe a general trend that TSE is the best and AT is better than PIT and De-AT, except for Conv-TasNet with overlap type None where AT is worse than PIT and De-AT. This may be due to the fact that Conv-TasNet is a fully convolutional model and it does not have a mechanism to store and propagate speaker identity over time. Additionally, ARN is the best performing model for AT, De-AT and TSE. It is encouraging to observe that ARN outperforms SpEx+, the baseline model proposed specifically for TSE. It is interesting to note that the performance differences between Conv-TasNet and DPRNN are not as significant as observed on WSJ0-2mix and WSJ0-3 mix datasets with full overlap. Finally, we notice that even though ARN has the best performance for the cases with a single output stream, it has worst performance for PIT, which uses 3 output streams.

C. Importance of Attentive Training for Speech Enhancement

We have reported in Table I that SE obtains better results than AT when dealing with single-talker input. What happens when a SE model is presented with an input mixture with sporadic speech interruptions? Now, we present results to assess this aspect. We compare AT and SE in Table IV on interaction patterns 1211111, 1112111, and 1111121, which are designed to simulate sporadic interruption scenarios.

We observe that AT obtains much better scores in most of the cases, which suggests that speech enhancement fails when presented with speech interruptions. Attentive training enables speech enhancement to deal with speech interruptions, and this is an important advantage of AT. We notice that AT is better for eSTOI for overlap type *Max* and *Half* but worse for *None*. We believe this is because the computation of eSTOI ignores silence intervals in the target signal, hence favoring SE in nonoverlapping intervals.

Next, we analyze behaviors of AT and SE in different segments of interaction patterns with sporadic interruptions. An interaction pattern of 12111111 contains 7 segments including 6 segments from the target and 1 from an interfering speaker. In Fig 7, we plot objective scores of AT and SE in 6 segments of the target speaker from the beginning to the end. We notice that SE obtains better results than AT in all the segments except for the one before and the one after the interfering talker. Particularly, the performance of SE for the segment before the interfering talker is much worse, implying that it fails in those segments. This establishes that AT is a more robust method than SE and does not fail when presented with speech interruptions.

D. Effects of Speech Onset Differences

The results discussed so far are on test sets in which the onset difference between the first and the second speaker is no smaller than $A = 1$ second. Now, we analyze the behavior of different methods when onset difference is gradually decreased. We plot results for interaction patterns 1221 and 123231 with overlap type *Max* in Fig. 8 The onset difference is gradually decreased from 1 second to 0.25 seconds with a step of 0.25 seconds. We consider two cases of TSE. TSE-1 uses enrollment utterances as specified in the original test set. TSE-2 sets the length of enrollment utterances to the length of onset difference.

We notice that there is a gradual decrease in the performance of all the models as the onset difference is decreased. TSE-2 and De-AT are the most unstable as the performance drops drastically below 0.75 seconds. AT outperforms PIT up to an onset difference of 0.5. The performance of AT drops drastically only for the case of small onset difference of 0.25 s. TSE-1 is the most stable for all the cases. These comparisons indicate that even though AT is sensitive to the onset difference, it generalizes well to smaller onset differences not used during training.

Next, in an attempt to improve the robustness of AT to smaller onset differences, we train ARN with AT using gradually decreasing values of A from {1, 0.75, 0.5, 0.25, 0.0} seconds. Note that $A = 0$ does not imply an onset difference of 0, but the minimum allowed onset difference of 0. We plot the performance of these ARN models in Fig. 9 for interaction pattern 123231 and compare it with PIT and TSE (the better-performing TSE-1) plotted in Fig. 8. We see a gradual improvement in the performance with decreasing value of A. Notable, the performance with $A = 0$ matches that of TSE and considerably outperforms PIT. This implies that the robustness of AT to smaller onset differences is easily improved by setting $A = 0$.

E. Speaker Encoding in ARN

The key idea of attentive training is to generate a speaker representation of the first speaker and use it to track this target speaker over the whole mixture. This implies that the hidden layers of the ARN model should have speaker information encoded in them. To investigate this, we present results on training speaker verification models on top of the hidden layers in the pretrained ARN model with frozen parameters. Speaker verification performance in terms of Equal Error Rate (EER) is given in Table V. We observe that training a speaker verification model from raw waveform obtains an EER of 4.5%. The output from the linear layer at the input improves the performance to 4.2%. Layers 2 and 4 do not provide any improvement. However, layers 1 and 3 respectively improve EER to 3.8% and 3.4%, which represents substantial relative EER improvements of 15.6% and 24.4% respectively. This demonstrates that the ARN model is implicitly creating a speaker representation to track the target speaker. We believe that the speaker recognition performance would be even better if we utilized an ARN trained with $A = 0$ instead of $A = 1$.

VII. Concluding Remarks

We have proposed a novel attentive training framework for speech enhancement. The key idea of attentive training is to attend to a single talker in a given multitalker mixture. Based on the principles of auditory selective attention, attentive training starts attending to (extracting) a speaker based on speech onset and continues attending to it irrespective of other interfering talkers. Attentive training is the first study, to our knowledge, to propose an intrinsic selection mechanism for speaker extraction. We have demonstrated that attentive training has the capability to extend a speech enhancement system to deal with speech interruptions as well as background noises.

We have compared attentive training with different methods of speaker extraction including speaker separation and target speaker extraction. Attentive training is found to be far better than PIT-based speaker separation, which does not have a speaker selection mechanism. Attentive training is competitive with target speaker extraction, which exploits cueing in the form of an enrollment utterance. We have also shown that an approach of decoupling attentive training into speaker selection and tracking obtains similar or worse results than end-to-end training.

Additionally, we have established the importance of attentive training for speech enhancement. We have shown that, when presented with speech interruptions, a speech enhancement system fails during these interruptions. An attentively trained model is found to be far more stable and performs enhancement well during interruptions.

Further, attentive training generalizes to untrained shorter onset differences. For example, a model trained with onset differences of more than 1 second generalizes well to an onset difference of 0.5 seconds. We have also verified that some of the hidden layers of the employed ARN model encode speaker information used for speaker tracking.

We plan to utilize attentive training to train a speech enhancement model to remove interfering speech only from the overlapping intervals instead of tracking the first speaker. Future research also includes investigating attentive training for speaker diarization and separation.

Acknowledgments

This research was supported in part by two NIDCD grants (R01DC012048 and R02DC015521), the Ohio Supercomputer Center, and the Pittsburgh Supercomputing Center (NSF ACI-1928147).

References

[1]. Loizou PC, Speech Enhancement: Theory and Practice, 2nd ed. Boca Raton, FL, USA: CRC Press, 2013.

[2]. Wang DL and Chen J, “Supervised speech separation based on deep learning: An overview,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 26, pp. 1702–1726, 2018. [PubMed: 31223631]

[3]. Lu X, Tsao Y, Matsuda S, and Hori C, “Speech enhancement based on deep denoising autoencoder.” in INTERSPEECH, 2013, pp. 436–440.

[4]. Xu Y, Du J, Dai L-R, and Lee C-H, "A regression approach to speech enhancement based on deep neural networks," IEEE/ACM Transactions on Audio, Speech and Language Processing, vol. 23, pp. 7–19, 2015.

[5]. Weninger F, Erdogan H, Watanabe S, Vincent E, Le Roux J, Hershey JR, and Schuller B, "Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR," in International Conference on Latent Variable Analysis and Signal Separation, 2015, pp. 91–99.

[6]. Chen J, Wang Y, Yoho SE, Wang DL, and Healy EW, "Large-scale training to increase speech intelligibility for hearing-impaired listeners in novel noises," The Journal of the Acoustical Society of America, vol. 139, pp. 2604–2612, 2016. [PubMed: 27250154]

[7]. Fu S-W, Tsao Y, and Lu X, "SNR-aware convolutional neural network modeling for speech enhancement." in INTERSPEECH, 2016, pp. 3768–3772.

[8]. Park SR and Lee J, "A fully convolutional neural network for speech enhancement," in INTERSPEECH, 2017, pp. 1993–1997.

[9]. Chen J and Wang DL, "Long short-term memory for speaker generalization in supervised speech separation," The Journal of the Acoustical Society of America, vol. 141.

[10]. Tan K, Chen J, and Wang DL, "Gated residual networks with dilated convolutions for supervised speech separation," in ICASSP, 2018, pp. 21–25.

[11]. Pandey A and Wang DL, "On adversarial training and loss functions for speech enhancement," in ICASSP, 2018, pp. 5414–5418.

[12]. Williamson DS, Wang Y, and Wang DL, "Complex ratio masking for monaural speech separation," IEEE/ACM Transactions on Audio, Speech and Language Processing, vol. 24, pp. 483–492, 2016. [PubMed: 27069955]

[13]. Choi H-S, Kim J-H, Huh J, Kim A, Ha J-W, and Lee K, "Phaseaware speech enhancement with deep complex U-Net," in ICLR, 2019.

[14]. Hu Y, Liu Y, Lv S, Xing M, Zhang S, Fu Y, Wu J, Zhang B, and Xie L, "DCCRN: Deep complex convolution recurrent network for phase-aware speech enhancement," in INTERSPEECH, 2020, pp. 2472–2476.

[15]. Zhou L, Gao Y, Wang Z, Li J, and Zhang W, "Complex spectral mapping with attention based convolution recurrent neural network for speech enhancement," arXiv:2104.05267, 2021.

[16]. Fu S-W, Hu T.-y., Tsao Y, and Lu X, "Complex spectrogram enhancement by convolutional neural network with multi-metrics learning," in Workshop on Machine Learning for Signal Processing, 2017, pp. 1–6.

[17]. Pandey A and Wang DL, "Exploring deep complex networks for complex spectrogram enhancement," in ICASSP, 2019, pp. 6885–6889.

[18]. Tan K and Wang DL, "Learning complex spectral mapping with gated convolutional recurrent networks for monaural speech enhancement," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 380–390, 2019. [PubMed: 33748323]

[19]. Pandey A and Wang DL, "Learning complex spectral mapping for speech enhancement with improved cross-corpus generalization," in INTERSPEECH, 2020, pp. 4511–4515.

[20]. Yu G, Li A, Wang H, Wang Y, Ke Y, and Zheng C, "Dbtnet: Dual-branch federative magnitude and phase estimation with attention-in-attention transformer for monaural speech enhancement," arXiv:2202.07931, 2022.

[21]. Fu S-W, Tsao Y, Lu X, and Kawai H, "Raw waveform-based speech enhancement by fully convolutional networks," arXiv:1703.02205, 2017.

[22]. Pascual S, Bonafonte A, and Serrà J, "SEGAN: Speech enhancement generative adversarial network," in INTERSPEECH, 2017, pp. 3642–3646.

[23]. Rethage D, Pons J, and Serra X, "A wavenet for speech denoising," in ICASSP, 2018, pp. 5069–5073.

[24]. Qian K, Zhang Y, Chang S, Yang X, Florêncio D, and Hasegawa-Johnson M, "Speech enhancement using bayesian wavenet," in INTERSPEECH, 2017, pp. 2013–2017.

[25]. Fu S-W, Wang T-W, Tsao Y, Lu X, and Kawai H, "End-to-end waveform utterance enhancement for direct evaluation metrics optimization by fully convolutional neural networks," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 26, pp. 1570–1584, 2018.

[26]. Pandey A and Wang DL, “A new framework for CNN-based speech enhancement in the time domain,” IEEE/ACM Transactions on Audio, Speech and Language Processing, vol. 27, pp. 1179–1188, 2019. [PubMed: 34262993]

[27]. ——, “TCNN: Temporal convolutional neural network for real-time speech enhancement in the time domain,” in ICASSP, 2019, pp. 6875–6879.

[28]. Wang K, He B, and Zhu W-P, “TSTNN: Two-stage transformer based neural network for speech enhancement in the time domain,” in ICASSP, 2021, pp. 7098–7102.

[29]. Pandey A and Wang DL, “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. [PubMed: 36245814]

[30]. Cherry EC, “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.

[31]. Broadbend D, “Perception and communication,” Pergamon Press, New York, 1958.

[32]. Hershey JR, Chen Z, Le Roux J, and Watanabe S, “Deep clustering: Discriminative embeddings for segmentation and separation,” in ICASSP, 2016, pp. 31–35.

[33]. Kolbæk M, Yu D, Tan Z-H, and Jensen J, “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.

[34]. Luo Y, Chen Z, and Mesgarani N, “Speaker-independent speech separation with deep attractor network,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 26, no. 4, pp. 787–796, 2018.

[35]. Liu Y and Wang DL, “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. [PubMed: 33748322]

[36]. Luo Y and Mesgarani N, “Conv-TasNet: Surpassing ideal time-frequency magnitude masking for speech separation,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 27, pp. 1256–1266, 2019. [PubMed: 31485462]

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

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

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

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

[41]. Wang Q, Muckenhirn H, Wilson K, Sridhar P, Wu Z, Hershey J, Saurous RA, Weiss RJ, Jia Y, and Moreno IL, “Voicefilter: Targeted voice separation by speaker-conditioned spectrogram masking,” arXiv:1810.04826, 2018.

[42]. Xu C, Rao W, Chng ES, and Li H, “Spex: Multi-scale time-domain speaker extraction network,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 1370–1384, 2020.

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

[44]. Zhang Z, He B, and Zhang Z, “X-tasnet: Robust and accurate time-domain speaker extraction network,” arXiv:2010.12766, 2020.

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

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

[47]. Afouras T, Chung JS, and Zisserman A, “The conversation: Deep audio-visual speech enhancement,” arXiv:1804.04121, 2018.

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

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

[50]. Brendel A, Haubner T, and Kellermann W, “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.

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

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

[53]. Bregman AS, Auditory scene analysis: The perceptual organization of sound. MIT press, 1994.

[54]. Treisman AM and Gelade G, “A feature-integration theory of attention,” Cognitive Psychology, vol. 12, no. 1, pp. 97–136, 1980. [PubMed: 7351125]

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

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

[57]. Pandey A and Wang DL, “Attentive training: A new training framework for talker-independent speaker extraction,” in INTERSPEECH, 2022, pp. 201–205.

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

[59]. Desplanques B, Thienpondt J, and Demuynck K, “ECAPA-TDNN: emphasized channel attention, propagation and aggregation in TDNN based speaker verification,” in INTERSPEECH, 2020, pp. 3830–3834.

[60]. Ravanelli M, Parcollet T, Plantinga P, Rouhe A, Cornell S, Lugosch L, Subakan C, Dawalatabad N, Heba A, Zhong J, Chou J-C, Yeh S-L, Fu S-W, Liao C-F, Rastorgueva E, Grondin F, Aris W, Na H, Gao Y, Mori RD, and Bengio Y, “SpeechBrain: A generalpurpose speech toolkit,” 2021, arXiv:2106.04624.

[61]. Panayotov V, Chen G, Povey D, and Khudanpur S, “Librispeech: an ASR corpus based on public domain audio books,” in ICASSP, 2015, pp. 5206–5210.

[62]. Wichern G, Antognini J, Flynn M, Zhu LR, McQuinn E, Crow D, Manilow E, and Roux JL, “WHAM!: Extending speech separation to noisy environments,” arXiv:1907.01160, 2019.

[63]. Grimm E, Van Everdingen R, and Schöpping M, “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.

[64]. Kingma D and Ba J, “Adam: A method for stochastic optimization,” in ICLR, 2015.

[65]. Nagrani A, Chung JS, and Zisserman A, “VoxCeleb: A Large-Scale Speaker Identification Dataset,” in INTERSPEECH, 2017, pp. 2616–2620.

[66]. Chung JS, Nagrani A, and Zisserman A, “Voxceleb2: Deep speaker recognition,” in INTERSPEECH, 2018.

[67]. Park DS, Chan W, Zhang Y, Chiu C-C, Zoph B, Cubuk ED, and Le QV, “SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition,” in INTERSPEECH, 2019, pp. 2613–2617.

[68]. Deng J, Guo J, Xue N, and Zafeiriou S, “Arcface: Additive angular margin loss for deep face recognition,” in CVPR, 2019, pp. 4690–4699.

[69]. Xiang X, Wang S, Huang H, Qian Y, and Yu K, “Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition,” in Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, 2019, pp. 1652–1656.

[70]. Smith LN, “Cyclical learning rates for training neural networks,” in Winter Conference on Applications of Computer Vision, 2017, pp. 464–472.

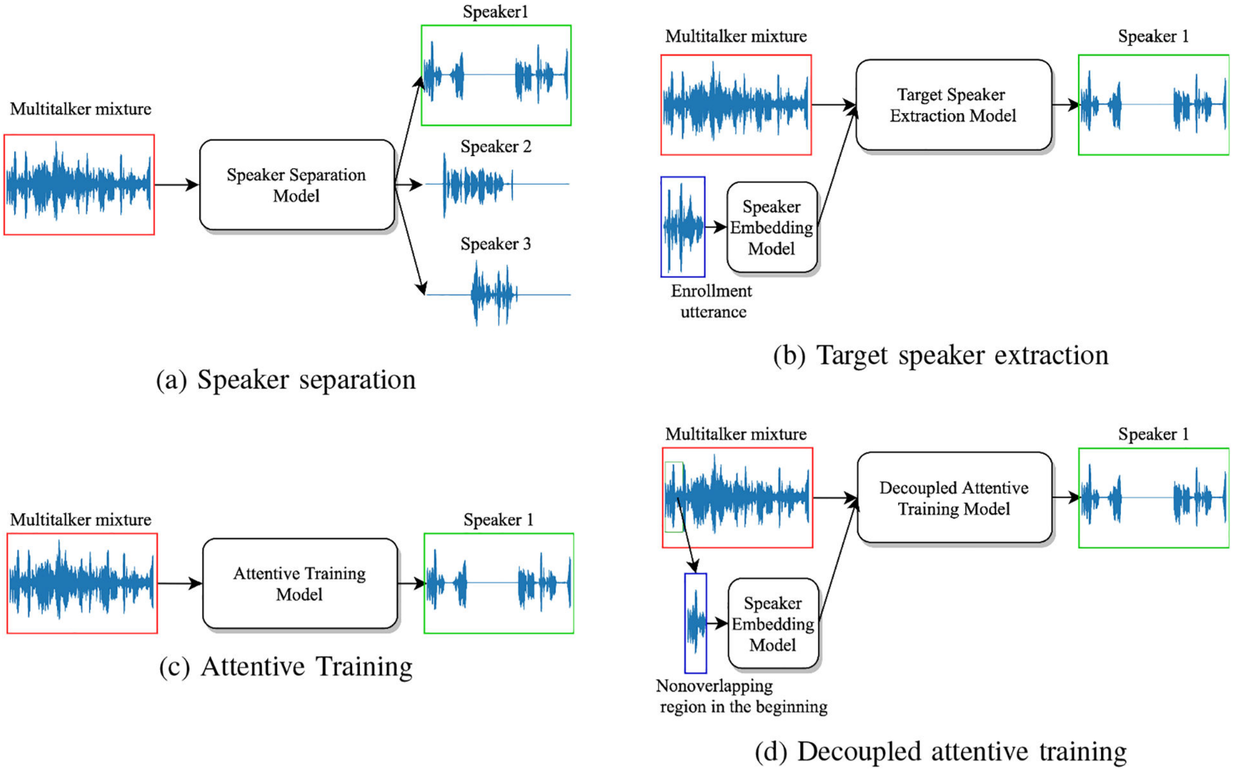
[71]. Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L, and Lerer A, “Automatic differentiation in PyTorch,” 2017.

[72]. Micikevicius P, Narang S, Alben J, Diamos G, Elsen E, Garcia D, Ginsburg B, Houston M, Kuchaiev O, Venkatesh G, and Wu H, “Mixed precision training,” in ICLR, 2018.

[73]. Jensen J and Taal CH, “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.

[74]. Rix AW, Beerends JG, Hollier MP, and Hekstra AP, “Perceptual evaluation of speech quality (PESQ) - a new method for speech quality assessment of telephone networks and codecs,” in ICASSP, 2001, pp. 749–752.

[75]. Ge M, Xu C, Wang L, Chng ES, Dang J, and Li H, “SpEx+: A complete time domain speaker extraction network,” in INTERSPEECH, 2020, pp. 1406–1410.



(a) Speaker separation

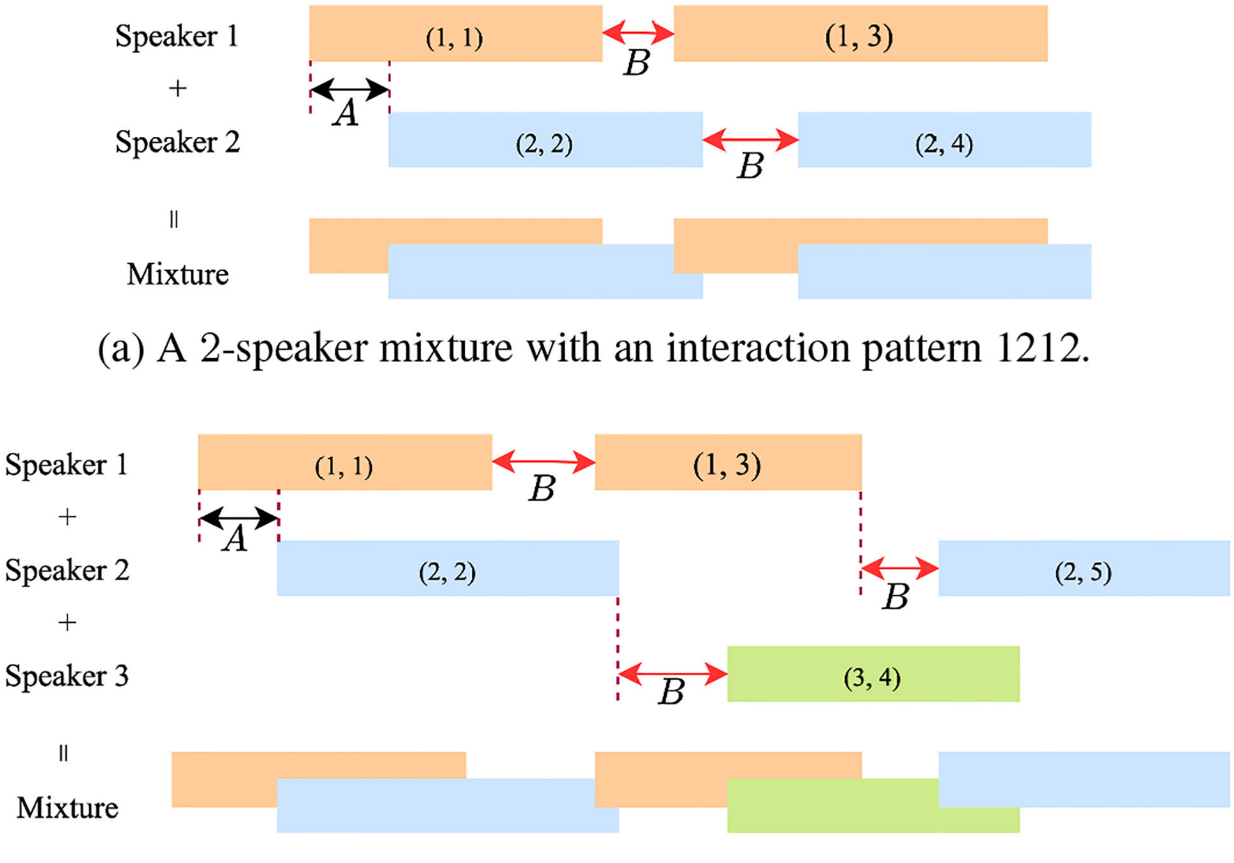
(b) Target speaker extraction

(c) Attentive Training

(d) Decoupled attentive training

Fig. 1:

Different methods for extracting the first speaker from a multitalker mixture.



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

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

Fig. 2:

Examples of interaction patterns with 2 and 3 speakers, and an initial minimum onset gap of A between the first and the second speaker. In pair (a, b) inside a box, a and b respectively index the speaker order and the segment order.

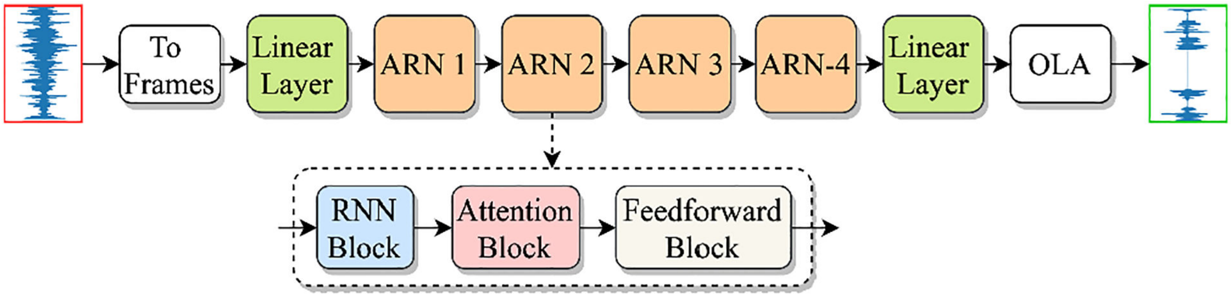


Fig. 3:
The model architecture used for attentive training.

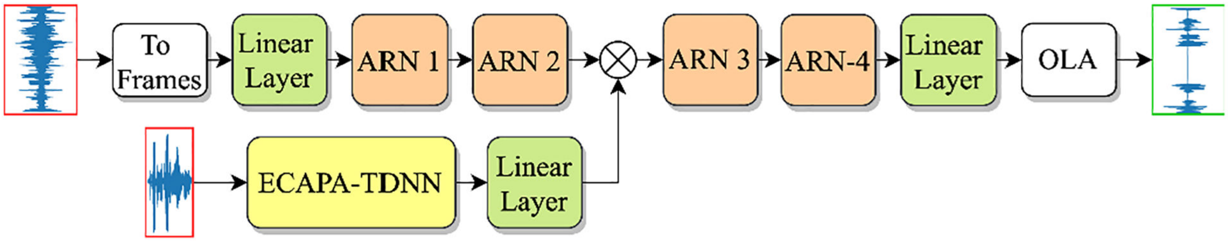


Fig. 4:
The model architecture used for target speaker extraction and decoupled attentive training.

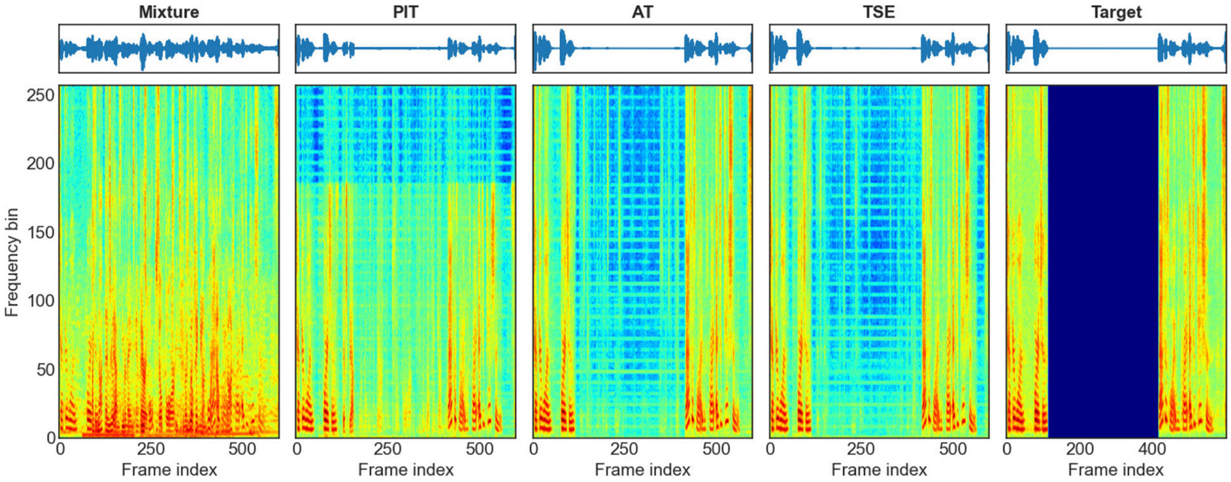


Fig. 5:

Spectrograms of a sample multitalker mixture enhanced using different methods.

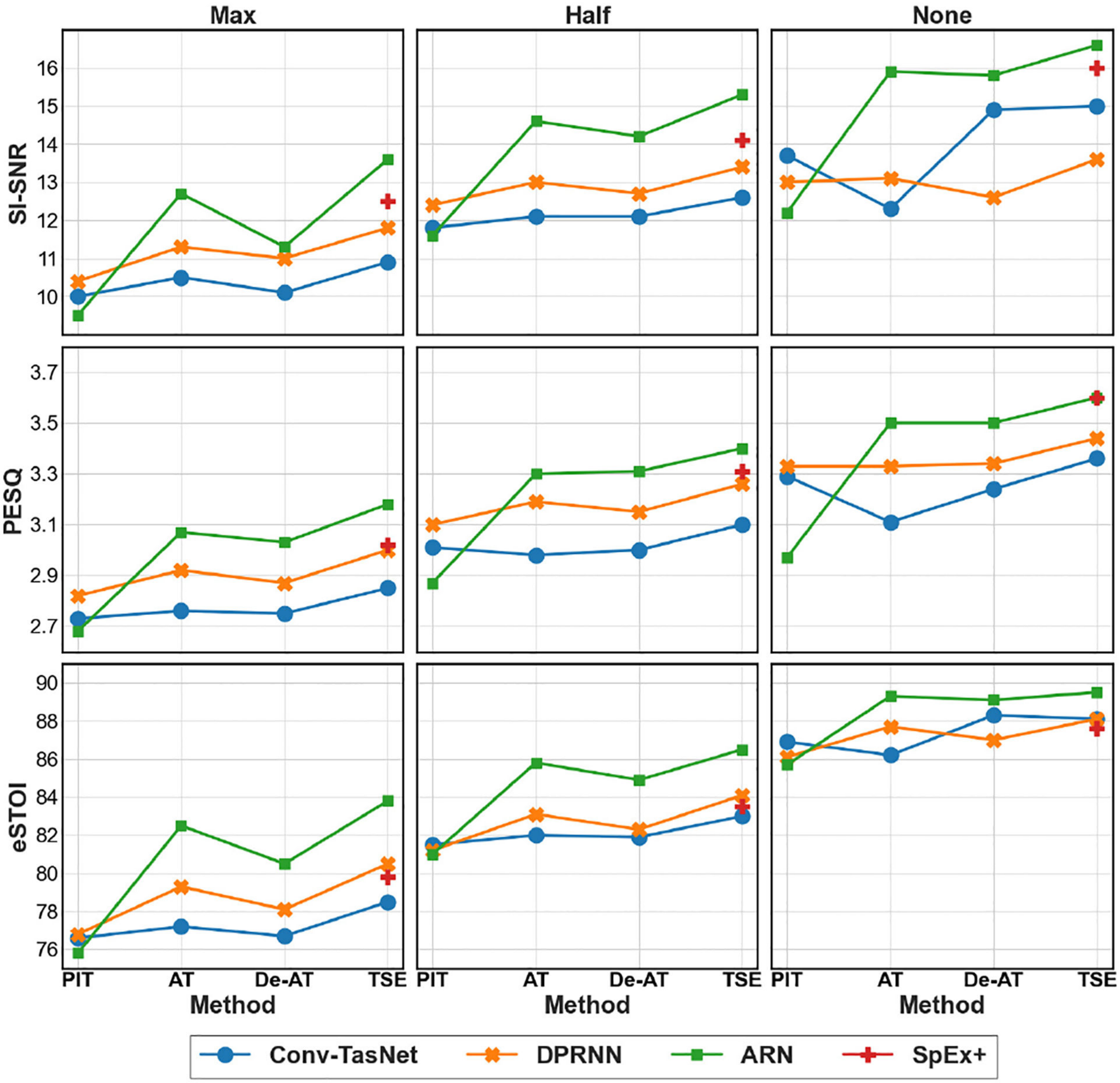


Fig. 6:

Comparison of Conv-TasNet, DPRNN, ARN and SpEx+ on interaction pattern 123231 for three types of overlap.

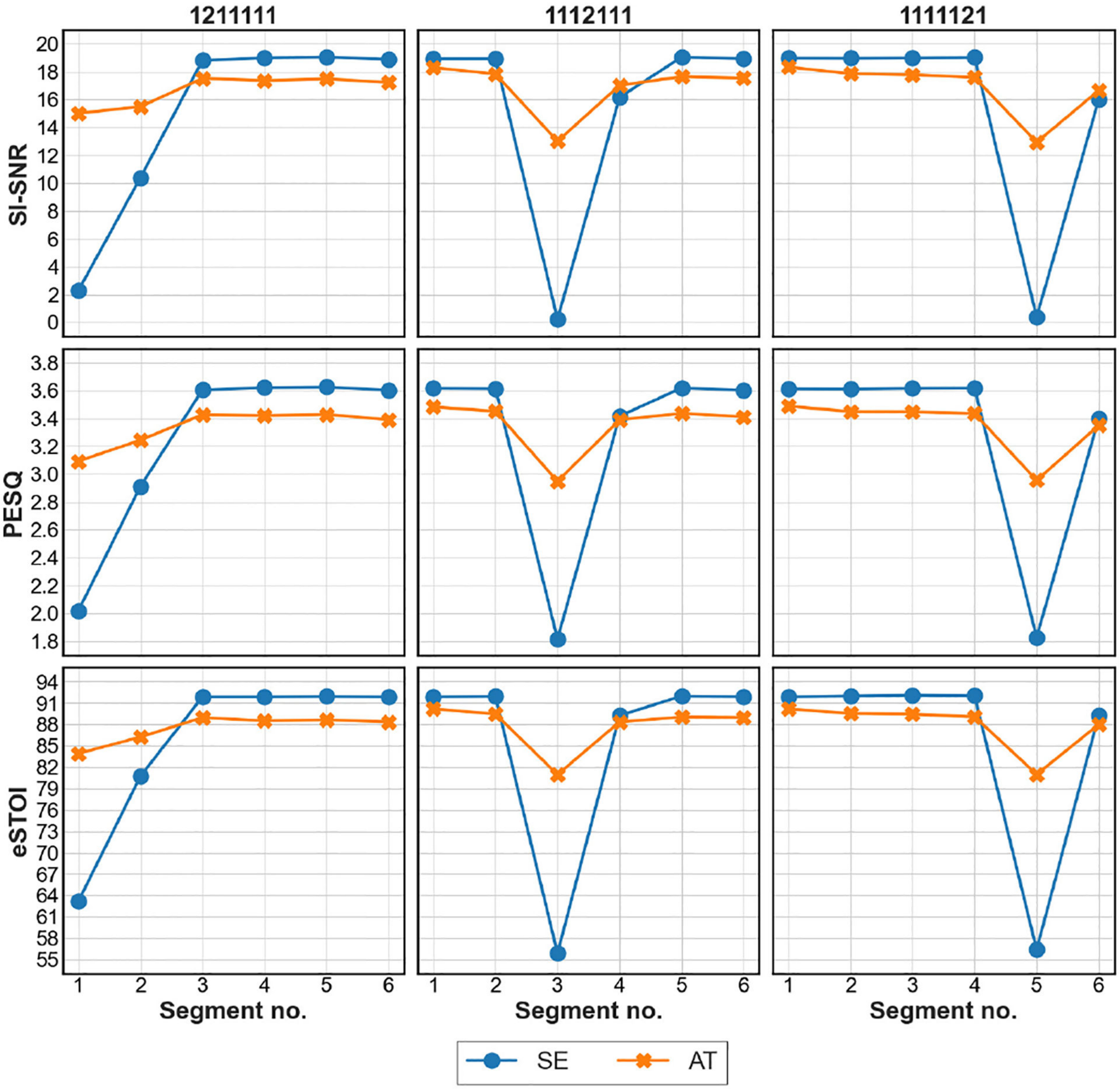


Fig. 7:

Comparing AT and SE on 6 segments of the target speaker. Results are plotted for interaction patterns 1211111, 1112111, and 1111121 with overlap type *Max*.

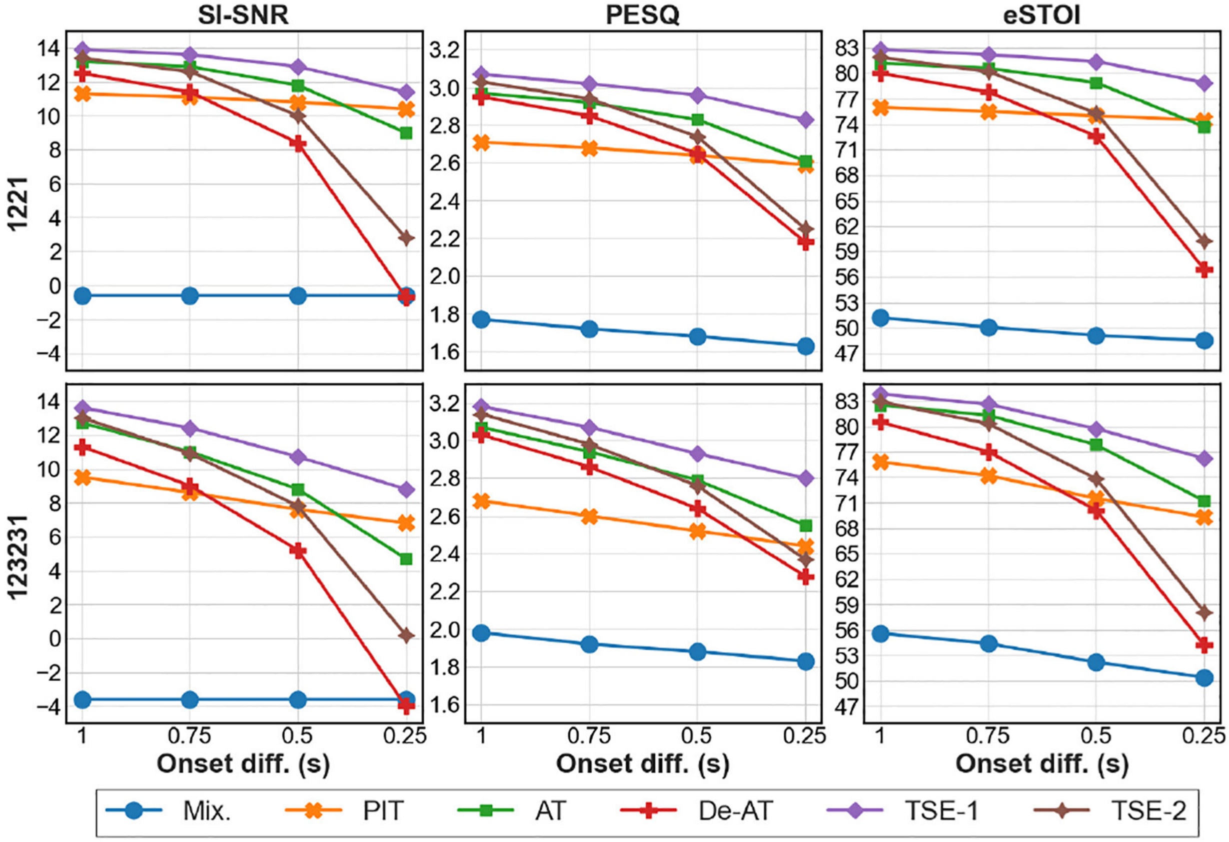


Fig. 8:

Performance comparisons with gradually decreasing onset difference.

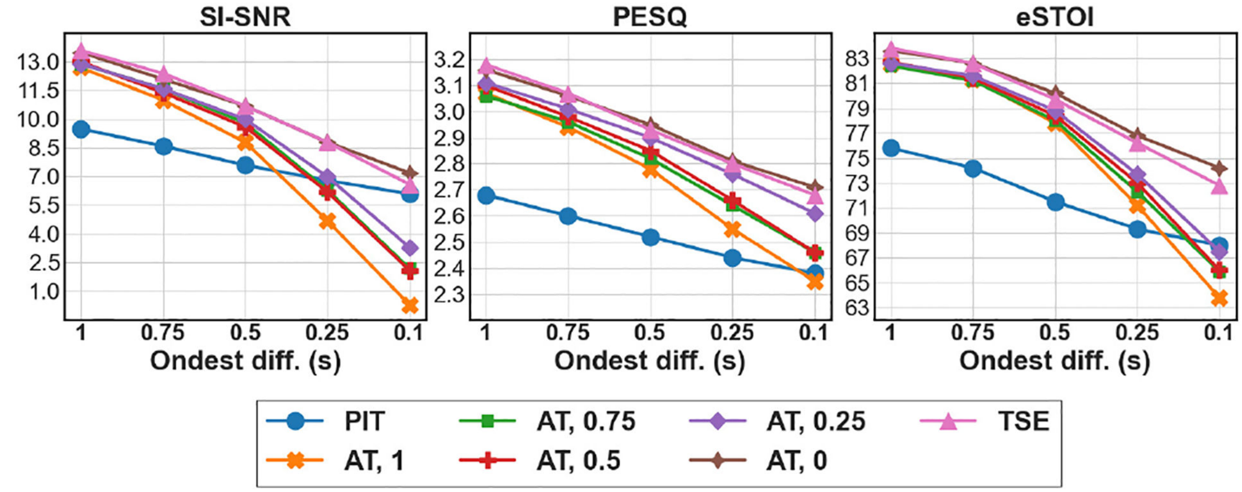


Fig. 9:

Comparing ARN trained with AT using a gradually decreasing value of A. AT, a denotes an ARN trained with AT using $A = a$.

TABLE I:

Comparing different methods on the interaction pattern 1111.

Metric	Mix.	PIT	AT	De-AT	TSE	SE
SI-SNR	9.5	15.2	**17.5**	17.4	17.3	**19.1**
PESQ	2.33	3.24	**3.40**	3.38	3.38	**3.61**
ESTOI	72.2	87.6	**89.8**	89.2	89.3	**92.6**

TABLE II:

Comparing different methods for trained numbers of speakers. (a) number of speakers, (b) interaction pattern, (c) whether trained on interaction pattern.

			Type	***Max***			***Half***			***None***		
(a)	**(b)**	**(c)**	**Metric**	**SI-SNR**	**PESQ**	**ESTOI**	**SI-SNR**	**PESQ**	**ESTOI**	**SI-SNR**	**PESQ**	**ESTOI**
2	1212	✓	Mix.	−0.6	1.67	51.4	−0.7	1.86	59.5	−1.0	2.28	71.9
			PIT	11.4	2.64	76.4	12.4	2.79	80.3	14.2	3.04	86.4
			AT	**13.4**	**2.90**	**81.7**	**14.8**	**3.09**	**85.0**	**16.4**	3.38	**89.6**
			De-AT	12.6	2.86	80.3	14.5	3.06	84.3	**16.4**	**3.39**	89.1
			TSE	**13.8**	**2.97**	**82.7**	**15.0**	**3.12**	**85.4**	**16.9**	**3.43**	**89.3**
	1221	✓	Mix.	−0.6	1.77	51.2	−0.8	2.06	60.1	−1.0	2.31	71.6
			PIT	11.3	2.71	76.0	12.5	2.92	80.5	14.3	3.06	86.0
			AT	**13.2**	**2.97**	**81.2**	**14.7**	**3.28**	**85.0**	16.5	3.47	**89.3**
			De-AT	12.5	2.95	80.0	14.5	3.25	84.4	**16.7**	**3.49**	88.9
			TSE	**13.9**	**3.07**	**82.8**	**15.1**	**3.32**	**85.6**	**17.1**	**3.55**	**89.1**
	122221	×	Mix.	−3.7	2.00	51.1	−3.8	2.16	60.2	−3.9	2.32	71.6
			PIT	11.2	2.83	76.1	12.5	2.94	80.5	14.1	3.03	85.7
			AT	**12.9**	**3.23**	**80.9**	**14.6**	**3.44**	**84.9**	16.2	3.52	**89.2**
			De-AT	12.4	3.22	80.0	14.5	3.40	84.5	**16.6**	**3.53**	89.0
			TSE	**13.8**	**3.33**	**82.7**	**15.2**	**3.49**	**85.9**	**17.1**	**3.62**	**89.5**
3	1231	✓	Mix.	−0.6	1.86	55.3	−0.8	2.08	62.7	−1.0	2.32	71.7
			PIT	10.3	2.66	76.0	12.5	2.90	81.5	13.3	3.04	86.4
			AT	**13.2**	**2.97**	**82.4**	**15.2**	**3.26**	**86.0**	**16.3**	3.46	**89.3**
			De-AT	12.2	2.93	80.8	14.7	3.23	85.1	16.2	**3.48**	88.9
			TSE	**14.0**	**3.08**	**83.8**	**15.4**	**3.31**	**86.4**	**16.8**	**3.54**	**89.3**
	123231	×	Mix.	−3.6	1.98	55.6	−3.7	2.14	62.6	−3.9	2.32	71.6
			PIT	9.5	2.68	75.8	11.6	2.87	81.0	12.2	2.97	85.7
			AT	**12.7**	**3.07**	**82.5**	**14.6**	3.30	**85.8**	**15.9**	**3.50**	**89.3**
			De-AT	11.3	3.03	80.5	14.2	**3.31**	84.9	15.8	3.50	89.1
			TSE	**13.6**	**3.18**	**83.8**	**15.3**	**3.40**	**86.5**	**16.6**	**3.60**	**89.5**

TABLE III:

Comparing different methods for the case of untrained number of speakers. (a) number of speakers, (b) interaction pattern.

		Type	***Max***			***Half***			***None***		
(a)	**(b)**	**Metric**	**SI-SNR**	**PESQ**	**ESTOI**	**SI-SNR**	**PESQ**	**ESTOI**	**SI-SNR**	**PESQ**	**ESTOI**
4	12341	Mix.	−2.4	1.98	57.5	−2.5	2.12	62.6	−2.8	2.31	71.8
		PIT	7.6	2.65	76.1	7.6	2.77	79.5	9.1	2.92	85.4
		AT	**13.0**	**3.07**	**83.3**	**14.7**	**3.28**	**85.9**	**16.0**	3.49	**89.3**
		De-AT	12.0	3.02	81.7	14.2	3.28	85.0	15.7	**3.50**	89.0
		TSE	**13.9**	**3.17**	**84.4**	**15.2**	**3.36**	**86.3**	**16.5**	**3.57**	**89.3**
5	123451	Mix.	−3.6	1.98	55.7	−3.7	2.14	62.8	−4.0	2.32	71.8
		PIT	4.9	2.55	73.5	4.3	2.67	77.4	6.0	2.85	85.0
		AT	**12.2**	**3.04**	**82.4**	**14.4**	**3.31**	**86.1**	**15.8**	**3.51**	**89.6**
		De-AT	11.4	3.02	81.0	14.0	3.30	85.2	15.5	**3.51**	89.3
		TSE	**13.4**	**3.17**	**83.7**	**15.1**	**3.40**	**86.4**	**16.3**	**3.59**	**89.3**

TABLE IV:

Comparing SE and AT for sporadic speech interruptions. (a) interaction pattern.

	Type	***Max***			***Half***			***None***		
(a)	**Metric**	**SI-SNR**	**PESQ**	**eSTOI**	**SI-SNR**	**PESQ**	**eSTOI**	**SI-SNR**	**PESQ**	**eSTOI**
1211111	Mix.	5.6	2.19	68.5	5.5	2.23	69.8	5.2	2.33	72.4
	SE	8.7	3.07	85.8	8.2	3.12	87.9	7.7	3.29	**92.5**
	AT	**15.9**	**3.26**	**88.0**	**16.2**	**3.31**	**88.6**	**16.2**	**3.37**	89.4
1112111	Mix.	5.5	2.18	68.5	5.5	2.21	69.4	5.2	2.33	72.5
	SE	8.5	3.05	86.0	8.3	3.09	87.3	7.7	3.28	**92.5**
	AT	**16.0**	**3.29**	**88.4**	**16.5**	**3.33**	**88.9**	**16.9**	**3.42**	90.0
1111121	Mix.	5.5	2.18	68.4	5.5	2.21	69.3	5.2	2.33	72.4
	SE	8.5	3.05	86.1	8.3	3.09	87.3	7.7	3.27	**92.6**
	AT	**16.0**	**3.28**	**88.4**	**16.4**	**3.31**	**88.8**	**17.1**	**3.42**	90.0

TABLE V:

Performance of speaker verification systems trained on top of the hidden layers in the ARN model.

Layer	**Raw**	**Lin-inp**	**ARN-1**	**ARN-2**	**ARN-3**	**ARN-4**
EER (%)	4.5	4.2	3.8	4.7	**3.4**	4.5