

# Attractor-Based Speech Separation of Multiple Utterances by Unknown Number of Speakers

Yuzhu Wang<sup>1</sup>, Archontis Politis<sup>1</sup>, Konstantinos Drossos<sup>2</sup>, Tuomas Virtanen<sup>1</sup>

<sup>1</sup>Signal Processing Research Center, Tampere University, Finland

<sup>2</sup>Nokia Technologies, Finland

yuzhu.wang@tuni.fi, archontis.politis@tuni.fi, konstantinos.drossos@nokia.com,  
tuomas.virtanen@tuni.fi

## Abstract

This paper addresses the problem of single-channel speech separation, where the number of speakers is unknown, and each speaker may speak multiple utterances. We propose a speech separation model that simultaneously performs separation, dynamically estimates the number of speakers, and detects individual speaker activities by integrating an attractor module. The proposed system outperforms existing methods by introducing an attractor-based architecture that effectively combines local and global temporal modeling for multi-utterance scenarios. To evaluate the method in reverberant and noisy conditions, a multi-speaker multi-utterance dataset was synthesized by combining Librispeech speech signals with WHAM! noise signals. The results demonstrate that the proposed system accurately estimates the number of sources. The system effectively detects source activities and separates the corresponding utterances into correct outputs in both known and unknown source count scenarios.

**Index Terms:** speech separation, speaker counting, speaker diarization, attractors, transformers

## 1. Introduction

Speech separation aims to isolate individual speech signals from mixtures containing multiple speakers. This task is particularly challenging in real-world environments due to noise, reverberation, and the highly time-varying nature of source activities, especially in single-channel scenarios [1, 2, 3, 4].

Most existing research assumes a known and fixed number of speakers with each contributing a single utterance [5, 6, 7, 8, 9, 10, 11, 12]. However, this assumption does not match with real-world scenarios, where the number of speakers is often unknown and varies over time, with each speaker potentially contributing multiple utterances that may overlap in variable ways. To handle unknown source numbers, one direct approach is recursive separation [13, 14]. In contrast, deep clustering methods [15, 16] create high-dimensional embeddings for each time-frequency bin, which are then clustered to group together the embeddings corresponding to the same speaker, thereby avoiding the dependency on the prior number of speakers. A similar approach is applied in speaker clustering [17]. Deep attractor networks [18, 19] try to create a representation vector for each sound source in feature space as a reference point. Following the success of attractor networks, systems that combine attractors and separators have gradually emerged as the mainstream solution. RNN-based attractors were explored in joint speaker diarization and speech separation systems [20, 21]. More recently, a transformer attractor was proposed in [22], which demonstrated excellent performance.

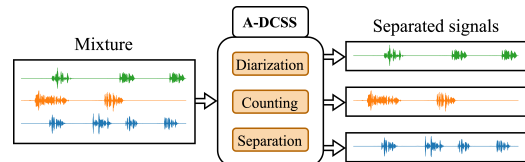


Figure 1: The illustration of the A-DCSS in multi-speaker with multi-utterance scenarios.

The aforementioned studies focus only on single-utterance scenarios. For situations with an unknown number of speakers and multiple utterances, several methods have shown potential effectiveness, including recurrent attention networks [23], spatial filtering [24], and dual-path model consisting of LSTM and transformers [25]. However, most of these approaches have concentrated on speaker counting and long-sequence modeling, overlooking source activity detection. Moreover, they primarily target speech recognition, with limited evaluations of separation performance and the impact of noise and reverberation.

Our work aims to address the novel problem of separating an unknown number of speakers, each with multiple utterances, a scenario that has not been evaluated in previous research, as shown in Fig. 1. Related research exists in continuous speech separation (CSS) [26, 27], which focuses on producing output streams where speech signals do not overlap, typically assuming a known number of sources. In contrast, our work emphasizes separation of unknown number of speakers while ensuring all utterances from the same speaker are consistently assigned to a single output channel, rather than merely de-overlapping the signals. We propose an attractor-based joint diarization, counting, and separation system (A-DCSS) that leverages multi-task learning to simultaneously minimize errors across these tasks. Our system makes four major contributions: (1) We are the first to test an attractor design for multi-utterance separation, validating its effectiveness in source modeling and potential source identification. (2) We integrate RNN attractors with feature-wise linear modulation (FiLM) [28] and develop an improved multi-path module with transformer-LSTM blocks [10]. (3) Our ablation studies demonstrate that source activity detection significantly improves separation performance. (4) We retrain and evaluate existing methods [13, 20, 21, 22] on the synthesized dataset, demonstrating their adaptability to multi-utterance scenarios. Our experimental results show that A-DCSS outperforms baseline systems in both fixed and varying speaker scenarios under anechoic, noisy, and reverberant conditions.

## 2. Signal Model

The observed signal  $\mathbf{y}$  of length  $T^*$  can be represented as

$$\mathbf{y} = \sum_{c=1}^C \mathbf{x}_c + \mathbf{n}. \quad (1)$$

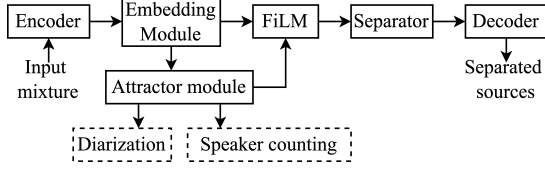


Figure 2: A-DCSS architecture.

Here, the observed signal  $\mathbf{y}$  consists of speech signals  $\mathbf{x}_c$  from  $C$  sources and noise  $\mathbf{n}$ . The signal  $\mathbf{x}_c$  from the  $c$ -th speaker, consists of multiple utterances and potential pauses between them. The speaker counting task involves estimating the number of speakers  $C$ . We assume that by extracting features from the time-domain signals, we can predict binary speaker activities  $\hat{\mathbf{P}}$  of size  $C \times T$ , where  $T$  denotes the number of time segments in which the activity is predicted. In realistic scenarios the number of speakers  $C$  is unknown, and the speech signal  $\mathbf{x}_c$  consists of multiple utterances. Our objective is to estimate  $\mathbf{x}_c$  by separating all utterances belonging to the same speaker into a single output. This process leverages the estimated speaker count  $\hat{C}$  and speaker activity  $\hat{\mathbf{P}}$ , resulting in  $\hat{C}$  separated speech signals, as illustrated in Fig. 1.

### 3. Attractor-based joint diarization, counting, and separation system (A-DCSS)

The proposed joint separation system consists of three core processes: feature embedding, attractor generation, and speech separation. The system architecture is illustrated in Fig. 2. An encoder and a decoder are used to transform the signal between the time domain and the time-feature domain. The embedding module maps the time-feature representations into a latent embedding space through a dual-path transformer-based network. The attractors are generated by learning source-level information from the embeddings. As shown in Fig. 2, the speaker diarization results and the prediction of the attractor existence probability can be obtained by the attractors, respectively. Then, the generated attractors are used as conditioning input, modulating the mixture features into speaker-specific features through a FiLM operation. Finally, we employ a triple-path separator to achieve speech separation.

**Encoder and decoder:** The encoder applies a one-dimensional convolutional layer with rectified linear unit (ReLU) activation to transform the input  $\mathbf{y} \in \mathbb{R}^{T^*}$  into a time-feature representation  $\mathbf{Y} \in \mathbb{R}^{T \times F}$ ,  $F$  being the feature dimension. The kernel size is  $L$  samples and the stride is  $L/2$  samples. The decoder reconstructs each separated signal using a transposed convolutional layer with the same hyperparameters as the encoder.

**Feature Embedding:** The feature embedding module of A-DCSS uses a dual-path design, as shown in Fig. 3. A linear layer is used to map the encoded features from dimension  $F$  to  $D$ . Following the Sepformer [9], we employ a chunking operation that segments the time dimension  $T$  into  $S$  chunks of equal length  $K$ , with a hop size of  $K/2$ , resulting in the input tensor  $\mathbf{D}_{in} \in \mathbb{R}^{K \times S \times D}$ . This dual-path structure allows the module to model both local dependencies within chunks through intra-chunk processing ( $S \times K \times D$ ) and global dependencies across chunks through inter-chunk processing ( $K \times S \times D$ ). In Fig. 3, P/R represents the permute/reshape operations. The intra-chunk and inter-chunk blocks follow the design in [9], consisting of a multi-head attention layer followed by a feed-forward layer, with layer normalization before each layer.

**Attractor module:** The attractor module aims to learn source-

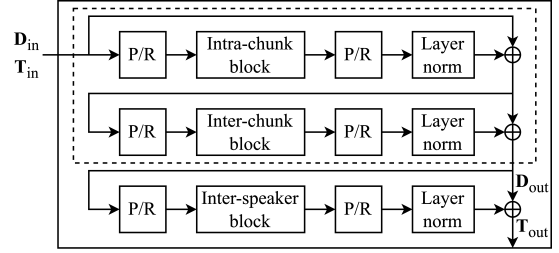


Figure 3: Dual-path (dashed line) and triple-path modules.  $\mathbf{D}_{out}$  and  $\mathbf{T}_{out}$  are their outputs with inputs  $\mathbf{D}_{in}$  and  $\mathbf{T}_{in}$ , respectively. P and R represent the permute and reshape operations.

level representations that characterize individual speakers in the mixture. Unlike the attractors in [18, 19] obtained through ideal mask and k-means clustering, our approach adopts a BLSTM-based encoder-decoder architecture for attractor generation as shown in Fig. 4, following [29]. The output  $\mathbf{D}_{out} \in \mathbb{R}^{K \times S \times D}$  from the dual-path module is processed by an overlap-add operation, reconstructing it into a sequence  $\{\mathbf{d}_t\}_{t=1}^T \in \mathbb{R}^{T \times D}$ . The LSTM-encoder encodes the obtained sequence to generate hidden state  $\mathbf{h} \in \mathbb{R}^D$  and cell states  $\mathbf{c} \in \mathbb{R}^D$ , which are subsequently used by the LSTM-decoder to generate  $J+1$  attractors  $\{\mathbf{a}_j\}_{j=1}^{J+1}$ , where  $\mathbf{a}_j \in \mathbb{R}^{1 \times D}$ . During the decoding step, a sequence of  $D$ -dimensional zero vectors  $\{\mathbf{h}_0\} \in \mathbb{R}^{(J+1) \times D}$  is used as input. The first  $J$  attractors  $\{\mathbf{a}_j\}_{j=1}^J \in \mathbb{R}^{J \times D}$  represent the identified  $J$  sources, while the final attractor  $\mathbf{a}_{J+1}$  is responsible for determining the non-existence of the speaker. As shown in Fig. 2, the attractor module produces three outputs: (1) The generated attractors  $\{\mathbf{a}_j\}_{j=1}^{J+1} \in \mathbb{R}^{(J+1) \times D}$  are passed through a linear layer with a sigmoid activation function to get the prediction of the attractor existence probability  $\hat{\mathbf{q}} \in \mathbb{R}^{J+1}$ . Binary prediction values are obtained by comparing each element of  $\hat{\mathbf{q}}$  with the predefined threshold  $\tau_{exist}$ . The estimated number of speakers  $\hat{C}$  corresponds to the number of existing attractors; (2) The inner product between the first  $J$  attractors  $\{\mathbf{a}_j\}_{j=1}^J \in \mathbb{R}^{J \times D}$  and the input embeddings  $\{\mathbf{d}_t\}_{t=1}^T \in \mathbb{R}^{T \times D}$  are passed through a linear layer with a sigmoid activation function to predict the probability of activity of each source at each time step, which is represented using matrix  $\hat{\mathbf{P}} \in \mathbb{R}^{J \times T}$ . This process follows the attractor-based diarization method in [29]. Binary activity predictions are obtained by comparing each element of  $\hat{\mathbf{P}}$  with the predefined threshold  $\tau_{diar}$ ; (3) The generated first  $J$  attractors  $\{\mathbf{a}_j\}_{j=1}^J \in \mathbb{R}^{J \times D}$  are fused using FiLM conditioning [28] with the dual-path module output  $\mathbf{D}_{out} \in \mathbb{R}^{K \times S \times D}$  to get the input of separator  $\mathbf{T}_{in} \in \mathbb{R}^{J \times K \times S \times D}$ . During training,  $J$  is set to the true number of speakers  $C$ . During inference, we define a maximum possible number of sources  $J_{max}$  and generate  $J_{max} + 1$  attractors. Then, an estimate  $\hat{C}$  is obtained based on the predicted probability of attractor existence, and the first  $\hat{C}$  attractors are passed to the speaker diarization and separation branches.

**Triple-path Separator:** The triple-path module extends the dual-path module by incorporating an additional inter-speaker block, following the design in [22], as shown in Fig. 3. After FiLM modulation, the feature tensor  $\mathbf{T}_{in} \in \mathbb{R}^{J \times K \times S \times D}$  incorporates an additional dimension  $J$  corresponding to the first  $J$  attractors. The inter-speaker block processes information along the  $J$  dimension, treating the  $K$  and  $S$  as a batch dimension. Each block of the triple-path module employs an enhanced transformer-LSTM block instead of the simple transformer block used in the dual-path module. This enhanced block, which has demonstrated superior performance for long

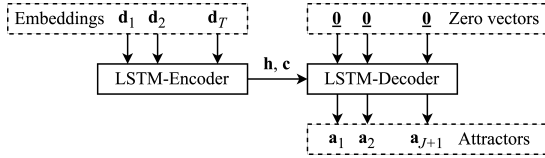


Figure 4: Attractor module.

speech sequence modeling [10], comprises a multi-head attention layer, an LSTM layer, and a linear layer. Furthermore, the three-path module is stacked  $N_{\text{triple}}$  times, which enhances the capacity to capture complex patterns. The output of the triple-path module  $\mathbf{T}_{\text{out}} \in \mathbb{R}^{J \times K \times S \times D}$  is first processed by overlap-add to obtain  $\mathbf{T}'_{\text{out}} \in \mathbb{R}^{J \times T \times D}$ . Then, a linear output layer transforms the feature dimension from  $D$  to  $F$ , and the  $J$  separated time-domain signals are obtained after decoding.

**Loss Function:** A-DCSS is trained using a joint loss function,  $\mathcal{L}_{\text{joint}} = \lambda_s \mathcal{L}_{\text{SI-SDR}} + \lambda_d \mathcal{L}_{\text{diar}} + \lambda_e \mathcal{L}_{\text{exist}}$ .  $\lambda_s$ ,  $\lambda_d$ , and  $\lambda_e$  denote the weights assigned to each loss item. Here,  $\mathcal{L}_{\text{SI-SDR}}$  is the SI-SDR loss that measures the reconstruction error between the separated signal  $\hat{\mathbf{x}}$  and reference signal  $\mathbf{x}$ .  $\mathcal{L}_{\text{diar}}$  represents the binary prediction loss for speaker activity. The permutation invariant training (PIT) method [7, 8] is used to calculate the diarization loss as

$$\mathcal{L}_{\text{diar}} = \min_{\text{PIT}} \sum_t \text{BCE}(\mathbf{p}_t, \hat{\mathbf{p}}_t), \quad (2)$$

where  $\mathbf{p}_t$  is a vector of reference labels at time  $t$ .  $\text{BCE}(\cdot, \cdot)$  is the binary cross entropy loss.  $\mathcal{L}_{\text{exist}}$  is the attractor existence loss,  $\mathcal{L}_{\text{exist}} = \text{BCE}(\mathbf{q}, \hat{\mathbf{q}})$ , where  $\mathbf{q} = [1, \dots, 1, 0] \in \mathbb{R}^{C+1}$ .

## 4. Experimental settings

### 4.1. Datasets

In our research scenario, multiple speakers coexist in a mixed signal, each contributing multiple utterances. To synthesize data consistent with this scenario, we use speech from the LibriSpeech corpus [30], which provides the diarization labels for model training and evaluation. Noise signals are from the WHAM! dataset [31]. We consider four scenarios: (a) anechoic, (b) noisy, (c) reverberant, and (d) noisy and reverberant mixtures. Each scenario involves mixtures containing either two or three speakers, with a random number of utterances per speaker, resulting in eight dataset configurations. Each configuration comprises a training set, a validation set, and a test set, with 20,000, 2,000, and 2,000 mixtures, respectively.

When synthesizing each anechoic mixture, we randomly select  $C$  speakers from LibriSpeech. For each selected speaker, we randomly extract between 1 and 5 speech utterances, where the number of utterances follows a uniform distribution. We use the original Librispeech signal levels without scaling, where the relative level between any two speech signals varies within  $[0, 5]$  dB. A random silence interval is inserted in the beginning and between each utterance which duration is drawn uniformly from  $[0, 3]$  seconds. Concatenating all speech utterances and silences for each speaker yields their complete speech signal  $\mathbf{x}_c$ . The anechoic multi-speaker, multi-utterance mixture is then obtained by superimposing these signals. This synthesis approach results in an average overlap ratio of 22.5% in our 2-speaker dataset, which aligns with natural conversational scenarios. To incorporate noise, we randomly extract a segment of equal length from the WHAM! noise dataset and adjust its amplitude according to a randomly generated signal-to-noise ratio (SNR) between 0 and 10 dB. In the SNR calculation, the signal level is calculated as the average power of all speakers in the

Table 1:  $\Delta\text{SI-SDR}$  (dB)( $\uparrow$ ) on the 2-speaker dataset.

Models	Anechoic	Noisy	Reverb	Noisy & Reverb
Conv-TasNet [5]	6.1	6.0	3.5	4.0
Sepformer [9]	8.1	7.4	7.1	6.8
Recursive-SS [13]	7.1	6.4	4.5	3.5
EEND-SS [20]	7.7	7.2	5.7	5.5
SepEDA [21]	10.1	8.4	7.5	7.6
SepTDA [22]	8.5	7.7	6.9	6.8
A-DCSS	<b>11.2</b>	<b>9.7</b>	<b>8.9</b>	<b>8.7</b>

logarithmic scale. This noise is then added to the anechoic mixture to produce a noisy multi-speaker, multi-utterance mixture. To simulate reverberant mixtures, we randomly generate room sizes and reverberation time (RT60), along with random positions for the microphone and the  $C$  speakers. Using the gpuRIR toolkit [32], we apply reverberation to each speaker. Then we can generate a reverberant, multi-speaker, multi-utterance mixture. The noise is added using the same method as before, resulting in a noisy, reverberant, multi-speaker, multi-utterance mixture. All synthesized signals are single-channel with a sampling rate of 16 kHz. The training and test sets do not share speakers, noise signals, or rooms.

We modeled rooms with dimensions randomly selected from the following ranges: length  $[4.0, 8.0]$  m, width  $[4.0, 8.0]$  m, and height  $[3.0, 4.0]$  m. The RT60 was varied between 0.2 and 0.6 seconds. Microphones were positioned at heights ranging from 1.0 to 1.5 m, while speakers were generated at heights between 1.5 and 2.0 m. Microphones and speakers were restricted to be at least 0.5 m away from any walls and each other.

### 4.2. Training Configurations

The A-DCSS model was trained in two phases: first with two-speaker samples, and then with a dataset containing a varying number of speakers. We utilized the ESPNet toolkit [33] for model training, employing the Adam optimizer. For the first phase, the initial learning rate was set to  $1.0 \times 10^{-3}$ , while in the second phase, it was reduced to  $1.0 \times 10^{-5}$ . Both training phases were configured to run for a maximum of 200 epochs with an early stopping strategy. The model architecture was configured with an encoder/decoder kernel size  $L$  of 16 and a stride of 8 samples. The feature dimensions  $D$  and  $F$  were both set to 256. The chunk length  $K$  is 96. We used 4 attention heads in each transformer layer. The  $N_{\text{triple}}$  was set to 6. The thresholds  $\tau_{\text{exist}}$  and  $\tau_{\text{diar}}$  were both set to 0.5. The  $\lambda_s$ ,  $\lambda_d$ , and  $\lambda_e$  were set to 0.8, 0.1, and 0.1, respectively. All these hyperparameters were determined through experiments on the validation set. During training, the batch size was set to 4, each training sample was split into 10 second segments in a batch.

### 4.3. Evaluation Metrics

We evaluate the separation performance of the A-DCSS using the scale-invariant source-to-distortion ratio improvement ( $\Delta\text{SI-SDR}$ ) [34]. Diarization performance is measured by the diarization error rate (DER) [35], which assesses the accuracy of detecting speaker activity. We also utilize speaker counting accuracy (SCA) [36] to evaluate the counting performance.

## 5. Experimental Results

We evaluated the performance of the A-DCSS in two scenarios: one with a fixed number of speakers and the other with a varying number of speakers. Besides, an ablation study was conducted

Table 2: Evaluation on 2-speaker & 3-speaker dataset. DER (%) ( $\downarrow$ ) is diarization error rate. SCA (%) ( $\uparrow$ ) is speaker counting accuracy.

Models	Anechoic			Noisy			Reverb			Noisy & Reverb		
	$\Delta$ SI-SDR	DER	SCA	$\Delta$ SI-SDR	DER	SCA	$\Delta$ SI-SDR	DER	SCA	$\Delta$ SI-SDR	DER	SCA
Recursive-SS [13]	6.2	–	–	5.5	–	–	3.9	–	–	3.1	–	–
EEND-SS [20]	6.7	8.4	96.7	6.0	9.7	94.1	4.6	8.5	96.2	4.5	10.2	93.7
SepEDA [21]	8.6	–	97.5	7.3	–	96.3	6.4	–	96.9	6.7	–	96.1
SepTDA [22]	7.8	–	<b>98.7</b>	6.9	–	<b>97.8</b>	6.0	–	<b>98.2</b>	5.9	–	<b>97.5</b>
A-DCSS	<b>9.7</b>	<b>5.9</b>	97.9	<b>8.8</b>	<b>6.4</b>	96.7	<b>7.8</b>	<b>6.2</b>	97.1	<b>7.7</b>	<b>7.1</b>	96.4

Table 3: Ablation experiment.

#	Attractor	Diarization	Counting	$\Delta$ SI-SDR
1	–	–	–	8.2
2	Transformer	–	✓	8.8
3	Transformer	✓	✓	10.3
4	RNN	–	✓	10.1
5	RNN	✓	✓	<b>11.2</b>

to analyze the importance of architectural choices. We compare the representative single-channel source separation methods, including Conv-Tasnet [5] and Sepformer [9]. In addition, we evaluate four advanced models designed for speech separation in scenarios where the number of speakers is unknown, as shown in Table 1.

- Recursive-SS [13]: Employs a recursive structure. The separation component is based on Conv-Tasnet [5].
- EEND-SS [20]: A baseline system integrating speaker diarization, counting, and separation. Both EEND-SS and A-DCSS adopt a three-task system with similar RNN attractors. The difference is that EEND-SS uses TCN [5] for feature extraction and separation, while A-DCSS employs transformer-based feature embedding module and separator, along with FiLM [28] for feature fusion.
- SepEDA [21]: Integrates speaker counting and separation. Uses an RNN attractor and a separator based on Sepformer [9]. In SepEDA [21], the attractor is used only for speaker counting, whereas in A-DCSS, it is utilized for both speaker counting and activity detection. Furthermore, the design of the separation component and the fusion of the attractor features differs between SepEDA [21] and A-DCSS.
- SepTDA [22]: Combines speaker counting and separation using a transformer attractor and transformer-based separator. Like A-DCSS, its separation module is inspired by [9]. The difference is that SepTDA employs transformer attractors for joint counting and separation, while A-DCSS uses RNN attractors for diarization, counting, and separation.

### 5.1. Fixed Number of Speakers

We conducted experiments using synthesized datasets of two-speaker mixtures in four acoustic scenarios, as shown in Table 1. In the anechoic condition, the A-DCSS achieves the best performance. Comparing Conv-TasNet [5], Recursive-SS [13], and EEND-SS [20], we observe that incorporating recursive structures (Recursive-SS [13]) or attractor modules (EEND-SS [20]) enhances separation performance in multi-speaker multi-utterance scenarios. The attractor outperforms the recursive process. In noisy, reverberant, and noisy-reverberant environments, we observe similar trends. Both noise and reverberation cause performance degradation of the A-DCSS, and the impact of reverberation is greater than that of noise.

### 5.2. Unknown Number of Speakers

We evaluated the A-DCSS on a varying number of speakers dataset combining 2-speaker and 3-speaker scenarios. Due to potential speaker counting errors, the number of reference signals may differ from the number of separated speech signals. To address this, we added silent audio signals to either the reference or separated speech signals to match the signal count. Table 2 presents the results. In all conditions tested, our proposed method achieved the best performance. All models showed performance degradation compared to the scenario with the fixed number of speakers. For the DER evaluation, the A-DCSS significantly outperformed the EEND-SS [20] baseline. Regarding the accuracy of source counting, the SepTDA method [22] performed best, with our proposed method ranking second. This comparison suggests that the transformer attractor outperforms the RNN attractor for the source count estimation subtask. For speaker diarization and counting, reverberation had little impact, while noise had a clearer effect.

### 5.3. Ablation experiment

In multi-utterance scenarios, the attractor module is essential to the A-DCSS. We conducted ablation experiments using the two-speaker anechoic dataset, with results in Table 3. The transformer attractor follows the SepTDA [22], and the RNN attractor is detailed in this paper. Diarization and counting represent two possible output branches of the attractor. All other modules and parameters remained unchanged during ablation.

When comparing row 1 to the others, the effect of the attractor is significant. Comparing row 2 and row 3, it can be observed that the  $\Delta$ SI-SDR improves by 1.5 dB when additional diarization constraints are added to the transformer attractors. Similarly, comparing row 4 and row 5 (A-DCSS), the SI-SDR increases by 1.1 dB when diarization output is incorporated into the RNN attractors. This demonstrates the importance of speaker diarization outputs from attractors for separation in multiple utterance scenarios. Further analysis of the results between row 2 and row 4, as well as between row 3 and row 5, indicates that the RNN-based attractor outperforms the transformer-based attractor in our experimental setup. This observation is also supported by the results presented in Table 1.

## 6. Conclusions

In this paper, we proposed a novel single-channel joint speech separation system that effectively handles scenarios with an unknown number of speakers, where each speaker may contribute multiple utterances. The system employs an RNN attractor module to estimate the number of sources and detect source activity, resulting in significant improvements in separation performance. Additionally, the proposed method demonstrates excellent robustness in noisy and reverberant environments.

## 7. References

- [1] E. Vincent, T. Virtanen, and S. Gannot, *Audio Source Separation and Speech Enhancement*. John Wiley & Sons, 2018.
- [2] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath, “Deep learning for audio signal processing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, 2019.
- [3] M. Cobos, J. Ahrens, K. Kowalczyk, and A. Politis, “An overview of machine learning and other data-based methods for spatial audio capture, processing, and reproduction,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2022, no. 1, p. 10, 2022.
- [4] D. Wang and J. Chen, “Supervised speech separation based on deep learning: An overview,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 10, pp. 1702–1726, 2018.
- [5] Y. Luo and N. Mesgarani, “Conv-tasnet: Surpassing ideal time-frequency magnitude masking for speech separation,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 8, pp. 1256–1266, 2019.
- [6] Y. Luo, Z. Chen, and T. Yoshioka, “Dual-Path RNN: Efficient long sequence modeling for time-domain single-channel speech separation,” in *Proc. ICASSP*, 2020, pp. 46–50.
- [7] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, “Permutation invariant training of deep models for speaker-independent multi-talker speech separation,” in *Proc. ICASSP*, 2017, pp. 241–245.
- [8] M. Kolbæk, D. Yu, Z.-H. Tan, and J. Jensen, “Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 10, pp. 1901–1913, 2017.
- [9] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, “Attention is all you need in speech separation,” in *Proc. ICASSP*, 2021, pp. 21–25.
- [10] J. Chen, Q. Mao, and D. Liu, “Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation,” in *Proc. Interspeech*, 2020, pp. 2642–2646.
- [11] Z.-Q. Wang, S. Cornell, S. Choi, Y. Lee, B.-Y. Kim, and S. Watanabe, “TF-GridNet: Making time-frequency domain models great again for monaural speaker separation,” in *Proc. ICASSP*, 2023, pp. 1–5.
- [12] —, “TF-GridNet: Integrating full- and sub-band modeling for speech separation,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 3221–3236, 2023.
- [13] N. G. Naoya Takahashi, Sudarsanam Parthasaarathy and Y. Mitsu-fuji, “Recursive speech separation for unknown number of speakers,” in *Proc. Interspeech*, 2019, pp. 1348–1352.
- [14] K. Kinoshita, L. Drude, M. Delcroix, and T. Nakatani, “Listening to each speaker one by one with recurrent selective hearing networks,” in *Proc. ICASSP*, 2018, pp. 5064–5068.
- [15] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, “Deep Clustering: Discriminative embeddings for segmentation and separation,” in *Proc. ICASSP*, 2016, pp. 31–35.
- [16] S. Wang, G. Naithani, and T. Virtanen, “Low-latency deep clustering for speech separation,” in *Proc. ICASSP*, 2019, pp. 76–80.
- [17] N. Zeghidour and D. Grangier, “Wavesplit: End-to-end speech separation by speaker clustering,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 2840–2849, 2021.
- [18] Z. Chen, Y. Luo, and N. Mesgarani, “Deep attractor network for single-microphone speaker separation,” in *Proc. ICASSP*, 2017, pp. 246–250.
- [19] Y. Luo, Z. Chen, and N. Mesgarani, “Speaker-independent speech separation with deep attractor network,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 4, pp. 787–796, 2018.
- [20] S. Maiti, Y. Ueda, S. Watanabe, C. Zhang, M. Yu, S.-X. Zhang, and Y. Xu, “EEND-SS: Joint end-to-end neural speaker diarization and speech separation for flexible number of speakers,” in *Proc. SLT*, 2023, pp. 480–487.
- [21] S. R. Chetupalli and E. A. P. Habets, “Speech separation for an unknown number of speakers using transformers with encoder-decoder attractors,” in *Proc. Interspeech*, 2022, pp. 5393–5397.
- [22] Y. Lee, S. Choi, B.-Y. Kim, Z.-Q. Wang, and S. Watanabe, “Boosting unknown-number speaker separation with transformer decoder-based attractor,” in *Proc. ICASSP*, 2024, pp. 446–450.
- [23] Y. Zhang, Z. Chen, J. Wu, T. Yoshioka, P. Wang, Z. Meng, and J. Li, “Continuous speech separation with recurrent selective attention network,” in *Proc. ICASSP*, 2022, pp. 6017–6021.
- [24] Z. Zhang, T. Yoshioka, N. Kanda, Z. Chen, X. Wang, D. Wang, and S. E. Eskimez, “All-neural beamformer for continuous speech separation,” in *Proc. ICASSP*, 2022, pp. 6032–6036.
- [25] D. Raj, L. Lu, Z. Chen, Y. Gaur, and J. Li, “Continuous streaming multi-talker asr with dual-path transducers,” in *Proc. ICASSP*, 2022, pp. 7317–7321.
- [26] 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 *Proc. ICASSP*. IEEE, 2020, pp. 7284–7288.
- [27] C. Li, Y. Luo, C. Han, J. Li, T. Yoshioka, T. Zhou, M. Delcroix, K. Kinoshita, C. Boeddeker, Y. Qian *et al.*, “Dual-path rnn for long recording speech separation,” in *Proc. SLT*. IEEE, 2021, pp. 865–872.
- [28] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. C. Courville, “FiLM: Visual reasoning with a general conditioning layer,” in *Proc. AAAI*, 2018.
- [29] S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, and K. Nagamatsu, “End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors,” *arXiv:2005.09921*, 2020.
- [30] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “LibriSpeech: an ASR corpus based on public domain audio books,” in *Proc. ICASSP*, 2015.
- [31] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. Le Roux, “WHAM!: Extending speech separation to noisy environments,” in *Proc. Interspeech*, 2019.
- [32] D. Diaz-Guerra, A. Miguel, and J. R. Beltran, “gpuRIR: A python library for room impulse response simulation with gpu acceleration,” *Multimedia Tools and Applications*, vol. 80, pp. 5653–5671, 2021.
- [33] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, “ESPnet: End-to-end speech processing toolkit,” in *Proc. Interspeech*, 2018, pp. 2207–2211.
- [34] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, “SDR – half-baked or well done?” in *Proc. ICASSP*, 2019, pp. 626–630.
- [35] J. G. Fiscus, J. Ajot, M. Michel, and J. S. Garofolo, “The rich transcription 2006 spring meeting recognition evaluation,” in *Proc. MLMI*, 2006, pp. 309–322.
- [36] S. R. Chetupalli and E. A. Habets, “Speaker counting and separation from single-channel noisy mixtures,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 1681–1692, 2023.