INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Overlap Aware Continuous Speech Separation without Permutation Invariant Training

Linfeng Yu, Wangyou Zhang, Chenda Li, Yanmin Qian[†]

MoE Key Lab of Artificial Intelligence, AI Institute
X-LANCE Lab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{ylf2017, wyz-97, lichenda1996, yanminqian}@sjtu.edu.cn

Abstract

Continuous speech separation (CSS) aims to separate a long-form signal with multiple partially overlapped utterances into a set of non-overlapped speech signals. While most existing CSS methods rely on the permutation invariant training (PIT) algorithm for training and inference, we argue that one may not need PIT at all to achieve promising CSS performance. In this paper, we propose a novel overlap aware CSS method, which explicitly identifies the non-overlapped segments in the long-form input to guide the separation of overlapped segments. We show that with the help of an external overlapping speech detection (OSD) model, an overlap-aware CSS model can be trained without PIT. In addition, an overlap-aware inference algorithm is proposed to greatly reduce the computational cost while preserving strong performance. Experiment results show that our proposed methods outperform the conventional stitching-based CSS approach, with over 1 dB signal-to-noise ratio (SNR) improvement.

Index Terms: Continuous speech separation, overlapping speech detection, permutation-free training

1. Introduction

With the vigorous development of speech-related human-machine interactive techniques, speech enhancement and separation-based frontend processing has played an increasingly important role in real-world speech processing. While speech enhancement focuses on removing the background noise and reverberation from the speech signal, speech separation aims at extracting the speech signals of different speakers from the mixture signal. Compared to the former, speech separation is especially important in tackling the well-known cocktail party problem [1, 2], where multiple speakers are active simultaneously. In this scenario, speech separation faces an additional challenge known as the permutation problem. In the literature, two representative methods have been proposed to solve this problem, i.e., deep clustering (DC) [3] and permutation invariant training (PIT) [4, 5]. Most existing deep learning-based speech separation techniques have been developed based on these two methods.

Although much progress has been achieved in the traditional speech separation task, it assumes a short utterance-level input signal with a high overlap ratio. In contrast, real-world conversational speech is often partially overlapped with a relatively low overlap ratio, and it can be very long in scenarios such as meetings [6]. Based on this observation, continuous speech separation (CSS) [7, 8] has been proposed to explicitly address the long-form speech separation problem in realistic conditions. It aims at separating partially overlapped long-form speech into several overlap-free outputs of the same length. One popular and representative CSS approach is the extension of utterance-level PIT (uPIT) [4], namely uPIT-CSS, which consists of three stages. First, the long-form input is divided into overlapped short chunks via a fixed-length sliding window. Second, uPIT-based separation is applied to each chunk. Finally, these chunk-level separated signals are combined to form full-length separation outputs via a stitching algorithm, where the PIT method is applied. Albeit simple and effective, the uPIT-CSS method is usually computationally expensive due to the highly-overlapped sliding windows required by the stitching algorithm.

Recently, researchers have been interested in improving the CSS method from different aspects. Some works [9, 10, 11, 12, 13] use additional modules to obtain speaker information in the long-form speech to assist separation. Some works [14, 15, 16, 17] divide the long-form speech into blocks and utilize the history information for separation in future blocks. Some other works [18, 19] focus on enhancing the permutation invariant training algorithm in the CSS task. Other works [20, 21, 22, 23] leverages multi-modal information such as spatial correlation and visual cues for CSS. In [24], Wang et al. propose to use an enhancement model and a separation model to respectively process non-overlapped and overlapped frames in the long-form signal. However, the tight relationship between non-overlapped and overlapped segments is not utilized.

In this paper, we propose a novel training framework called Overlap Aware CSS. Our system includes an overlapping speech detection (OSD) model and a speaker-biased separation model with an auxiliary network. The OSD model takes the long-form speech as input to detect overlapped and non-overlapped segments. Then each non-overlapped single-speaker segment is leveraged as a condition to help the separation model separate the subsequent overlapped speech and decide the permutation. This procedure is naturally free of the permutation problem, thus getting rid of the PIT method. With the help of the OSD model, we also propose an overlap-aware inference algorithm to focus the speaker-biased separation model on processing overlapped segments, while non-overlapped segments will be directly stitched to the separated overlapped segments. In this way, we can significantly reduce the computation cost during inference. We evaluated the proposed methods on simulated meeting-style data. Our experiments show that our proposed methods outperform the traditional stitching-based CSS approach by more than 1 dB signal-to-noise ratio (SNR) improvement on the meeting-style simulation data.

2. Overlap aware CSS

Our proposed system consists of two models: a frame-level overlapping speech detection model and a speaker-biased separation model.

[†] Yanmin Qian is the corresponding author.

10.21437/Interspeech.2023-1530

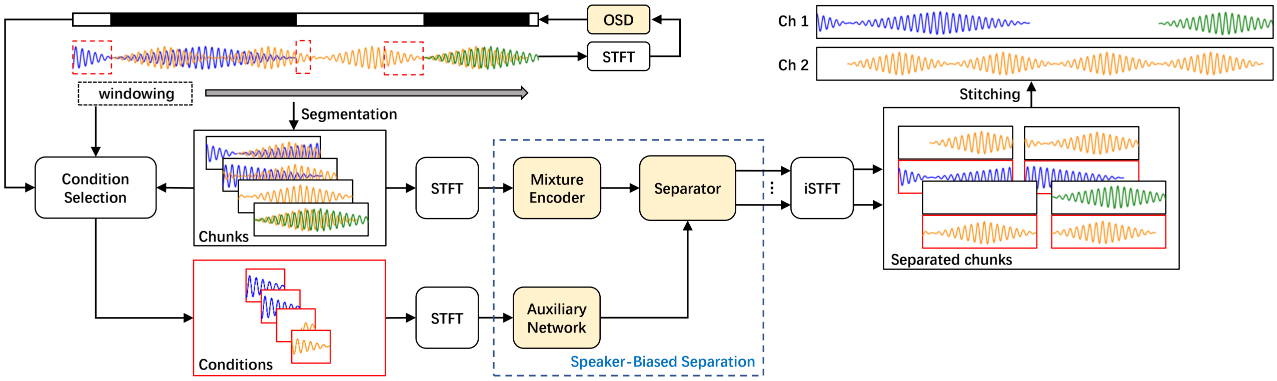

Figure 1: *Overall process of proposed overlap aware continuous speech separation. Overlapped segments are painted in black while non-overlapped segments are painted in white in the output of the OSD model. Trainable modules are painted in yellow.*

2.1. Overlapping Speech Detection

In our proposed method, we follow the basic assumption in CSS [7] that at most C speakers are active simultaneously, where C is the number of output channels. Here, we adopt $C = 2$. Based on the above assumption, the permutation problem in CSS can be directly resolved by using a deterministic permutation $\mathcal{P}$ for the separated signals in each overlapped segment. Our intuition is that each non-overlapped segment and the subsequent overlapped segment next to it are likely to both contain the same speaker. Consequently, $\mathcal{P}$ can be obtained by conditioning on the neighboring non-overlapped segment before the current overlapped segment such that the shared speaker is always separated in the first output channel.

To this end, a frame-level overlapping speech detection (OSD) model can be used to identify overlapped and non-overlapped segments, which is essentially a binary classifier. While any binary classifier that provides frame-level overlap information can be used here, we adopt the block-based convolutional neural network (CNN) based model [25] in this paper.

2.2. Speaker-Biased Separation

In the following discussion, we use the term “chunk” to represent the model input selected from long-form speech through a sliding window, which may contain both overlapped and non-overlapped “segments”. After the non-overlapped single-speaker segments have been identified by the OSD model, we can develop a speaker-biased separation model that is free of the permutation problem. The proposed model takes as input a mixture signal $\mathbf{X}$ and the adjacent single-speaker segment $\mathbf{C_a}$ before it and generates the corresponding separated speech. As mentioned in Section 2.1, the output permutation is deterministic by ensuring that the speaker in the first output channel is the same as in the single-speaker condition $\mathbf{C_a}$. Inspired by the success of target speech extraction [15, 16], we adopt the structure of frequency-domain SpeakerBeam [26] as our speaker-biased separation model, which contains Mixture Encoder $\mathrm{Enc}_{\mathrm{Mix}}(\cdot)$ for processing mixture, Auxiliary Network $\mathrm{AuxNet}(\cdot)$ for processing condition segment and Separator for generating outputs. In order to generate multiple outputs, we increase the number of output heads at the last layer of the original extraction model. The training process can be formulated as follows:

$$\left[\hat{\mathbf{S}}_a, \hat{\mathbf{S}}_b\right] = \mathrm{Separator}\left(\mathrm{Enc}_{\mathrm{Mix}}(\mathbf{X}), \mathrm{AuxNet}(\mathbf{C}_a)\right), \quad (1)$$

$$\mathcal{L} = -\,\mathrm{SNR}(\mathbf{S}_a, \hat{\mathbf{S}}_a) - \mathrm{SNR}(\mathbf{S}_b, \hat{\mathbf{S}}_b), \quad (2)$$

where $\mathbf{X}$ and $\mathbf{C}_a$ are the input mixture and the corresponding condition of speaker a, respectively. $\hat{\mathbf{S}}_a$ and $\hat{\mathbf{S}}_b$ denote the separated speech for speakers a and b, respectively, while $\mathbf{S}_a$ and $\mathbf{S}_b$ are the corresponding reference signals. $\mathcal{L}$ is the loss function.

The construction of the single-speaker condition is illustrated in Fig. 1. More specifically, for each windowed chunk, a corresponding single-speaker segment is chosen as the condition by the following rule. If there is an adjacent single-speaker segment before the chunk, it is used as the condition. When no adjacent single-speaker segment is found before the chunk, we will try to use the non-overlapped segment from the last chunk if it exists. If a chunk contains the beginning of a single-speaker segment, we use itself as its own condition. In our experiment, all condition segments are padded or chopped to a fixed length. We use oracle information to select conditions according to the above rule during training.

2.3. Overlap-aware Inference

In the inference stage, our proposed model can be directly used with the conventional chunk-wise stitching-based CSS method [7]. During stitching, the permutation for each chunk is adjusted by placing the first output speech in the same channel as the conditioning speech. To find the channel of conditioning speech, we will first use the output of overlapping speech detection model to get the timestamp of the conditioning speech, and the non-silence channel is the channel of conditioning speech of this chunk.

However, with the help of the OSD model, we can further propose a novel overlap-aware stitching method, where the separation model only processes overlapped segments.[1] Then we just stitch the separated signals with unprocessed non-overlapped segments. Since non-overlapped segments are used as the additional condition, the permutation problem during stitching can also be avoided.

3. Experiments

3.1. Dataset

We simulated a reverberant meeting-style data based on LibriSpeech [27], which is mostly the same as that in [28, 29], except that we did not include noise in the generated data. The training, development, and evaluation sets contain 30,000, 900, and 900 samples, respectively. Each sample is 90s long, containing 3–5 speakers with an overlap ratio between 50% and 80%. The reverberation time ranges from 100 ms to 500 ms.

3.2. Model Configurations and Training Details

We adopt the T-F masking method for speech separation. The size of short-time Fourier transformation (STFT) is 512-point while the hop length is 256. The sliding window size for CSS is 3.2 seconds. All single-speaker conditions are padded or chopped to 1 second. The proportion of windowed chunks with

[1] We leave denoising and dereverberation in the future work.

Table 2: *Window-level SNR (dB) (3.2s) with different overlap ratios for different models with different stitching methods. "chunk-wise" denotes the conventional CSS approach where each chunk is processed independently via a sliding window. "overlap-aware" denotes only overlapped segments will be processed and then stitched with non-overlapped segments. "MACs of 90s" denotes the computational cost of the separation model to process a 90s-long speech, while numbers in brackets show the additional computational cost of the OSD model. The proposed methods are marked with* *.

Model	Stitching Type	Overlap ratio between windows	Overlap ratio (%)					MACs of 90s (G)
			0–25	25–50	50–75	75–100	Avg	
No Separation	-	-	0.00	0.00	0.00	0.00	0.00	-
BLSTM	chunk-wise	50%	13.67	9.94	8.61	7.75	9.08	292.36
		0%	11.73	8.27	7.59	7.43	8.12	148.50
	*overlap-aware	50%	16.99	8.86	7.60	5.38	7.98	129.94 (+ 5.57)
		0%	14.62	7.55	6.68	6.98	7.88	116.02 (+ 5.57)
*Speaker-biased BLSTM	chunk-wise	50%	14.46	**11.56**	**9.75**	**8.66**	**10.17**	290.51
		0%	11.36	9.31	8.50	8.09	8.81	147.56
	overlap-aware	50%	17.39	10.41	8.91	8.28	9.90	133.73 (+ 5.57)
		0%	**17.44**	10.33	8.83	8.20	9.83	110.67 (+ 5.57)
Conformer	chunk-wise	50%	17.60	12.01	9.85	8.71	10.66	270.24
		0%	11.79	9.35	8.23	8.19	8.82	137.26
	*overlap-aware	50%	14.05	8.17	7.34	7.90	8.50	120.11 (+ 5.57)
		0%	13.93	7.92	7.05	7.55	8.21	107.24 (+ 5.57)
*Speaker-biased Conformer	chunk-wise	50%	**19.71**	**14.17**	**11.89**	**10.73**	**12.73**	194.95
		0%	13.75	11.07	10.14	9.76	10.57	99.02
	overlap-aware	50%	18.38	11.72	10.33	9.89	11.33	86.64 (+ 5.57)
		0%	18.32	11.66	10.26	9.80	11.25	77.36 (+ 5.57)

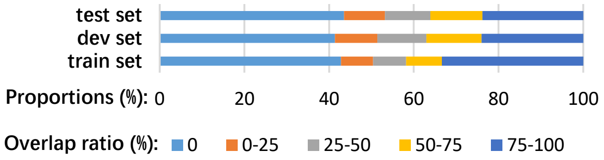

Figure 2: *Proportion of windowed chunks with different overlap ratios. A 3.2s sliding window with 50% overlap ratio is applied.*

different overlap ratios in the simulated data is shown in Fig. 2. All experiments were conducted using the ESPnet [30] toolkit.

For the recurrent neural network (RNN)-based model, we adopt the structure of frequency-domain SpeakerBeam [26] in the speaker-biased separation model. Both baseline and proposed models have 5 bi-directional long short-term memory (BLSTM) blocks. Each block consists of a BLSTM layer followed by a linear projection layer, a global LayerNorm layer, and a tanh activation layer with the residual connection. The input dimension is 257 for the first block and 256 for the rest blocks. The hidden dimension of the BLSTM block is 515 for the baseline model and 512 for the speaker-biased separation model. The auxiliary network of the speaker-biased separation model has the same configuration as in the frequency-domain SpeakerBeam [26]. As a result, the baseline model has 23.05M parameters and the proposed model has 23.01M parameters. We set the initial learning rate to $1e^{-4}$ and use the StepLR scheduler, where the learning rate is decayed by a factor of 0.98 every two epochs.

For the Conformer-based model, we use the same Conformer-base configuration as in [8], which has 16 Conformer encoder layers with 4 attention heads, 256 attention dimensions, and 1024 feed-forward network (FFN) dimensions. For the speaker-biased separation model, we adopt the cross-attention conformer structure proposed in [31, 32]. It consists of two independent conformer encoders for processing the input mixture and the condition respectively, followed by 8 cross-attention conformer blocks to obtain the separation masks. Each conformer encoder includes 8 conformer layers with 4 attention heads, 228 attention dimensions, and 512 FFN dimensions. Similarly, each cross-attention conformer block is comprised of a cross-attention conformer layer with 4 attention heads, 228 attention dimensions, and 512 FFN dimensions. As a result, the baseline model has 21.54M parameters and the speaker-biased separation model has 21.43M parameters. We set the learning rate to $2e^{-4}$ and use the warm-up learning rate scheduler with 20000 warm-up steps.

During training, each minibatch includes 8 long-form samples. All separation models are trained for 150 epochs with the Adam optimizer [33], and the patience for early stopping is 10.

The OSD model follows the same structure and training configuration as in [25], with 888.58K parameters. Different from above, we directly use the 90s-long samples to train and evaluate the OSD model. The learning rate is $1e^{-3}$ and the batch size is 1.

Table 1: *Evaluation of the overlapping speech detection model.*

Class	Acc	Recall	Precision	F1
Overlapped speech	0.900	0.887	0.790	0.828
Non-overlapped speech	0.900	0.906	0.951	0.927

3.3. Evaluation on Simulated Data

Table 1 shows the performance of our OSD model. Since the model is not completely accurate in predicting overlapped segments and some extremely short overlapped segments may occur, inspired by the one-dimensional dilation-erosion algorithm in [34], we perform a similar dilation-erosion post-process smoothing strategy on the predicted overlapped segments with the kernel size set to 5 frames. In this way, we can eliminate extremely short overlapped segments.

Table 2 shows the results of baseline models and our speaker-biased separation models on the simulated dataset[2]. We compare different stitching strategies when processing the long-form speech, where a stitching window with a 0% or 50% overlap ratio can be used. After generating the long-form separated signals, the window-level SNR is computed by dividing

[2] Results on LibriCSS: `https://earthmanylf.github.io/oacss/libricss.pdf`

Table 3: *Window-level SNR (dB) (3.2s) for oracle and predicted overlapping speech information. A sliding window with a 50% overlap ratio is applied.*

Model	Stitch Type	Oracle	Overlap ratio (%)			
			0–25	25–50	50–75	75–100
Speaker-biased BLSTM	chunk-wise	✓	15.17	11.82	9.89	8.76
		✗	14.46	11.56	9.75	8.66
	overlap-aware	✓	19.85	10.67	9.23	8.53
		✗	17.39	10.41	8.91	8.28
Speaker-biased Conformer	chunk-wise	✓	20.27	14.05	11.8	10.71
		✗	20.05	14.03	11.79	10.61
	overlap-aware	✓	20.41	11.94	10.64	10.14
		✗	18.38	11.72	10.33	9.89

the long-form signal into chunks using a 3.2s sliding window. We calculate the average SNR based on the speaker overlap ratio in each chunk.

The results show that both proposed Speaker-biased BLSTM and Speaker-biased Conformer models outperform their baseline counterparts, with over 1 dB SNR improvement. In addition, when the conventional chunk-wise stitching method is applied with a 0% window overlap, our proposed models can still achieve strong performance that is comparable to or even better than the baseline performance when a 50% window overlap is used. Furthermore, we evaluate the effectiveness of the proposed overlap-aware stitching method. It can be seen that the computational cost is largely reduced, while the overall performance is only moderately degraded. It is interesting to see that the SNR performance in chunks with low overlap ratios (0–25%) is even better than in the chunk-wise stitching method, especially for the BLSTM-based models. This might attribute to the energy leakage problem when processing an almost single-speaker signal with a speech separation model. Also, we can observe that the proposed stitching method is not sensitive to the window overlap ratio, which allows further reduction of computation.

Table 3 compares the performance of the speaker-biased separation model when using oracle and predicted overlap information as the condition. While better performance can be achieved by using the oracle overlap information, we can see that the SNR performance gap is less than 0.3 dB in most conditions with a high overlap ratio ($> 25\%$). This also shows the superiority of the proposed methods.

3.4. Ablation Study

We base Speaker-biased Conformer model to make ablation studies. All the experiments use the chunk-wise stitch type, and a 50% overlap ratio is applied. Table 4 shows the performance of different condition fusion methods. We use three widely used techniques: cross-attention [32], conditional layer normalization (cLN) [35], and scaled activation based SpeakerBeam [26]. All the speaker-biased separation models have similar sizes of parameters and were trained under the same optimizer, which is shown in Section 3.2. It can be seen that all the speaker-biased separation models outperform their baseline counterparts, with over 1 dB SNR improvement. Cross-attention model achieves the best performance among all three speaker-biased separation models. This suggests that the cross-attention architecture allows conformer to achieve stronger fusion capabilities, compared to the other two techniques.

Table 5 shows the effects of different condition lengths and chunk sizes on the same trained model as in Section 3.3, which was trained with 1s condition length and 3.2s chunk size. Although longer conditioning speech contains more information about a speaker, we can see that shorter conditioning speech does not have a significant impact on the performance of the speaker-biased separation model. When we use 0.25s of speech as a condition, which is a quarter of the length of the conditioning speech used during training, we only get a performance loss of 0.2 dB SNR compared to the default setting. On the other hand, we can observe that a sufficient chunk size is crucial for the speaker-biased separation model. Reducing the chunk size from 2s to 1s leads to significant performance degradation in all conditions. In addition, the model with a shorter chunk size (1s) loses more performance at lower overlap ratios (0–25%) than at higher overlap ratios ($> 25\%$). This may attribute to the energy leakage problem when processing almost overlap-free speech with a separation model, where the separated signal in one output channel is highly similar to that in the other. In contrast, using a longer chunk can effectively mitigate this issue by providing more contextual information.

Table 4: *Window-level SNR (dB) (3.2s) of ablation studies on the performance of different condition fusion methods. A sliding window with a 50% overlap ratio is applied. Stitch type is chunk-wise. "CA" denotes cross-attention, "cLN" denotes conditional layer normalization, "SB" denotes scaled activation based SpeakerBeam.*

Model	Fusion Method	Overlap ratio (%)				
		0–25	25–50	50–75	75-100	Avg
Conformer	-	17.60	12.01	9.85	8.71	10.66
Speaker-biased Conformer	CA [32]	**19.71**	**14.71**	**11.89**	**10.73**	**12.73**
	cLN [35]	17.56	13.31	11.16	10.01	11.81
	SB [26]	17.82	13.29	11.15	10.01	11.84

Table 5: *Window-level SNR (dB) (3.2s) of ablation studies on the performance of condition length and chunk size. The experiments are based on the same cross-attention Speaker-biased Conformer. Stitch type is chunk-wise. A sliding window with a 50% overlap ratio is applied.*

Condition Length	Chunk Size	Overlap ratio (%)				
		0–25	25–50	50–75	75-100	Avg
1s	3.2s	19.71	14.71	11.89	10.73	12.73
0.5s	3.2s	19.40	14.12	11.87	10.71	12.67
0.25s		18.58	14.03	11.83	10.68	12.53
1s	2s	9.96	12.59	11.34	10.37	11.01
	1s	0.80	9.07	9.20	8.91	8.09

4. Conclusions

In this paper, we propose a novel CSS training framework called overlap-aware CSS based on speaker-biased speech separation and overlapping speech detection. The proposed framework trains the speaker-biased speech separation model by providing a single-speaker segment to help the model determine the permutation of outputs, thus getting rid of the PIT method. In addition, we introduce an overlap-aware inference algorithm to generate separated long-form speech with the help of the OSD model. Experimental results demonstrate that our framework outperforms the traditional stitching-based CSS approach, with over 1 dB SNR improvement. Moreover, our models can maintain strong performance while largely reducing the computational cost using the proposed inference algorithm.

Future work includes investigating the denoising and dereverberation capabilities of the proposed model and exploring the use of the proposed model for automatic speech recognition.

5. Acknowledgement

This work was supported in part by China STI 2030-Major Projects under Grant No. 2021ZD0201500, in part by China NSFC projects under Grants 62122050 and 62071288, and in part by Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102.

6. References

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

[2] Y. Qian, C. Weng, X. Chang, S. Wang, and D. Yu, "Past review, current progress, and challenges ahead on the cocktail party problem," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 1, pp. 40–63, 2018.

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

[4] 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. IEEE ICASSP*, 2017, pp. 241–245.

[5] M. Kolbaek, 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. ASLP.*, vol. 25, no. 10, pp. 1901–1913, 2017.

[6] Ö. Çetin and E. Shriberg, "Analysis of overlaps in meetings by dialog factors, hot spots, speakers, and collection site: Insights for automatic speech recognition," in *Ninth International Conference on Spoken Language Processing*, 2006.

[7] 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. IEEE ICASSP*, 2020, pp. 7284–7288.

[8] S. Chen, Y. Wu, Z. Chen, J. Wu, J. Li, T. Yoshioka, C. Wang, S. Liu, and M. Zhou, "Continuous speech separation with conformer," in *Proc. IEEE ICASSP*, 2021, pp. 5749–5753.

[9] C. Han, Y. Luo, C. Li, T. Zhou, K. Kinoshita, S. Watanabe, M. Delcroix, H. Erdogan, J. R. Hershey, N. Mesgarani, and Z. Chen, "Continuous speech separation using speaker inventory for long recording," in *Proc. Interspeech*, 2021, pp. 3036–3040.

[10] R. Paturi, S. Srinivasan, K. Kirchhoff, and D. Garcia-Romero, "Directed speech separation for automatic speech recognition of long form conversational speech," in *Proc. Interspeech*, 2022, pp. 5388–5392.

[11] Y. Takashima, Y. Fujita, S. Watanabe, S. Horiguchi, P. García, and K. Nagamatsu, "End-to-end speaker diarization conditioned on speech activity and overlap detection," in *Proc. IEEE SLT*, 2021, pp. 849–856.

[12] 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. IEEE SLT*, 2023, pp. 480–487.

[13] B. Zeng, W. Wang, Y. Bao, and M. Li, "Simultaneous speech extraction for multiple target speakers under the meeting scenarios," *arXiv preprint arXiv:2206.08525*, 2022.

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

[15] M. Ge, C. Xu, L. Wang, E. S. Chng, J. Dang, and H. Li, "Multi-stage speaker extraction with utterance and frame-level reference signals," in *Proc. IEEE ICASSP*, 2021, pp. 6109–6113.

[16] A. Pandey and D. Wang, "Attentive Training: A New Training Framework for Talker-independent Speaker Extraction," in *Proc. Interspeech*, 2022, pp. 201–205.

[17] 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. IEEE ICASSP*, 2022, pp. 6017–6021.

[18] T. v. Neumann, K. Kinoshita, C. Boeddeker, M. Delcroix, and R. Haeb-Umbach, "Graph-PIT: Generalized permutation invariant training for continuous separation of arbitrary numbers of speakers," in *Proc. Interspeech*, 2021, pp. 3490–3494.

[19] W. Zhang, Z. Chen, N. Kanda, S. Liu, J. Li, S. Emre Eskimez, T. Yoshioka, X. Xiao, Z. Meng, Y. Qian, and F. Wei, "Separating long-form speech with group-wise permutation invariant training," in *Proc. Interspeech*, 2022, pp. 5383–5387.

[20] Z.-Q. Wang and D. Wang, "Localization based sequential grouping for continuous speech separation," in *Proc. IEEE ICASSP*, 2022, pp. 281–285.

[21] T. Yoshioka, X. Wang, D. Wang, M. Tang, Z. Zhu, Z. Chen, and N. Kanda, "VarArray: Array-geometry-agnostic continuous speech separation," in *Proc. IEEE ICASSP*, 2022, pp. 6027–6031.

[22] 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. IEEE ICASSP*, 2022, pp. 6032–6036.

[23] A. Rahimi, T. Afouras, and A. Zisserman, "Reading to listen at the cocktail party: Multi-modal speech separation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 493–10 502.

[24] Z.-Q. Wang and D. Wang, "Count and separate: Incorporating speaker counting for continuous speaker separation," in *Proc. IEEE ICASSP*, 2021, pp. 11–15.

[25] M. Yousefi and J. H. L. Hansen, "Block-based high performance CNN architectures for frame-level overlapping speech detection," *IEEE/ACM Trans. ASLP.*, vol. 29, pp. 28–40, 2021.

[26] K. Žmolíková, M. Delcroix, K. Kinoshita, T. Ochiai, T. Nakatani, L. Burget, and J. Černocký, "SpeakerBeam: Speaker aware neural network for target speaker extraction in speech mixtures," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 800–814, 2019.

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

[28] C. Li, Z. Chen, Y. Luo, C. Han, T. Zhou, K. Kinoshita, M. Delcroix, S. Watanabe, and Y. Qian, "Dual-path modeling for long recording speech separation in meetings," in *Proc. IEEE ICASSP*, 2021, pp. 5739–5743.

[29] C. Li, Z. Chen, and Y. Qian, "Dual-path modeling with memory embedding model for continuous speech separation," *IEEE/ACM Trans. ASLP.*, vol. 30, pp. 1508–1520, 2022.

[30] C. Li, J. Shi, W. Zhang, A. S. Subramanian, X. Chang, N. Kamo, M. Hira, T. Hayashi, C. Boeddeker, Z. Chen, and S. Watanabe, "ESPnet-SE: End-to-end speech enhancement and separation toolkit designed for ASR integration," in *Proc. IEEE SLT*, 2021, pp. 785–792.

[31] A. Narayanan, C.-C. Chiu, T. O'Malley, Q. Wang, and Y. He, "Cross-attention conformer for context modeling in speech enhancement for ASR," in *Proc. IEEE ASRU*, 2021, pp. 312–319.

[32] T. R. O'Malley, A. Narayanan, and Q. Wang, "A universally-deployable ASR frontend for joint acoustic echo cancellation, speech enhancement, and voice separation," in *Proc. Interspeech*, 2022, pp. 3829–3833.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[34] G.-B. Wang and W.-Q. Zhang, "An RNN and CRNN based approach to robust voice activity detection," in *Proc. APSIPA ASC*, 2019, pp. 1347–1350.

[35] M. Chen, X. Tan, B. Li, Y. Liu, T. Qin, sheng zhao, and T.-Y. Liu, "Adaspeech: Adaptive text to speech for custom voice," in *Proc. ICLR*, 2021.