Microphone Array Channel Combination Algorithms for Overlapped Speech Detection

Théo Mariotte, Anthony Larcher, Silvio Montrésor, Jean-Hugh Thomas

Microphone Array Channel Combination Algorithms for Overlapped Speech Detection

Théo Mariotte[1,2]*, Anthony Larcher*[2]*, Silvio Montrésor*[1]*, Jean-Hugh Thomas* [1]

[1]LAUM, UMR CNRS 6613, IA-GS, Le Mans Université, Av. Olivier Messiaen 72085 Le Mans, France
[2]LIUM, Le Mans Université, Av. René Laennec 72085 Le Mans, France

theo.mariotte@univ-lemans.fr

Abstract

Overlapped speech occurs when multiple speakers are simultaneously active. This may lead to severe performance degradation in automatic speech processing systems such as speaker diarization. Overlapped speech detection (OSD) aims at detecting time segments in which several speakers are simultaneously active. Recent deep neural network architectures have shown impressive results in the close-talk scenario. However, performance tends to deteriorate in the context of distant speech. Microphone arrays are often considered under these conditions to record signals including spatial information. This paper investigates the use of the self-attention channel combinator (SACC) system as a feature extractor for OSD. This model is also extended in the complex space (cSACC) to improve the interpretability of the approach. Results show that distant OSD performance with self-attentive models gets closer to the near-field condition. A detailed analysis of the cSACC combination-weights is also conducted showing that the self-attention module focuses attention on the speakers' direction.

Index Terms: overlapped speech detection, multi-microphone, distant speech, interpretability

1. Introduction

Speaker diarization in the multi-party scenario is still a challenging task [1–3]. Diarization systems are subject to severe performance degradation when several speakers are overlapping, which may naturally occur in spontaneous speech. Overlapped speech detection (OSD) is thus needed for robust speaker diarization [4] to process overlapped speech segments separately.

Recent advances in neural network based OSD have shown impressive results in near-field conditions compared to classical approaches [5–8]. Notably, recurrent neural networks such as Long Short-Term Memory (LSTM) [9, 10] reached state-of-the-art near-field OSD performance. Few researches have however been produced on the distant speech scenario [11–13] while this configuration offers practical benefits by avoiding speakers to wear their own microphone.

On distant OSD, Cornell et al. [12] proposed a Temporal Convolutional Network (TCN) [14] based OSD and speaker counting architecture. The authors also proposed a detailed benchmark of several OSD architectures [13] and showed that distant OSD can be improved by fusing spatial features with acoustic features. Only handcrafted spatial features were investigated, which may not be as optimal as spatial features learned in an end-to-end manner.

Spatial information available in multi-microphone signals can be exploited using spatial filtering. Many algorithms based on classical signal processing [15–18] or on neural networks [19–21] have been proposed in the literature. Signal-based approaches often require extracting explicit information about the input signal (e.g. speaker location, noise statistics). On neural approaches, Gong *et al.* [22] proposed the Self Attention Channel Combinator (SACC) which learns how to combine channels using self-attention [23]. Combination-weights are learned on the magnitude of the multichannel Short-Time Fourier Transform (STFT) of the input signal. This approach has shown significant improvement in the context of distant Automatic Speech Recognition (ASR).

Building on these previous works, we propose a multichannel OSD system based on the SACC algorithm. SACC acts as a feature extractor from the raw multi-microphone input signal. Two types of OSD systems are implemented, respectively based on Bidirectional LSTM (BLSTM) and TCN sequence modeling. As combination-weights learned by SACC offer limited interpretation, we propose its extension in the complex space, cSACC. cSACC learns complex combination-weights to preserve the phase information in the STFT and get closer to standard spatial filtering algorithms (e.g. beamforming [15]). We also show that cSACC weights are easily interpretable and provide information on the spatial directions exploited by the OSD system. To the best of our knowledge, this work is the first application of SACC for the task of OSD and its extension to use complex weights is the first attempt in the domain.

The paper is organized as follows. In Section 2, we introduce the supervised OSD framework. The BLSTM and TCN architectures are also presented. In Section 3, we describe the SACC and the cSACC architectures and their integration into the OSD system. The dataset, experiments and results are presented in Sections 4 and 5. Finally, a detailed analysis of the cSACC combination-weights is conducted in Section 6. Conclusions and perspectives are drawn in Section 7.

2. Overlapped Speech Detection

2.1. Principle

We formulate OSD as a binary classification task. Let $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N\}$ be a sequence of feature vectors, with N being the number of frames, and its associated sequence of binary labels $\boldsymbol{y} = \{y_1, y_2, \ldots, y_N\}$. We wish to determine the optimal parameters $\hat{\boldsymbol{\theta}}$ of a model $f(\boldsymbol{X}, \boldsymbol{\theta})$ to predict the sequence $\hat{\boldsymbol{y}} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_N\}$, with $\hat{y}_n \in \{0, 1\}$ being the predicted binary label of the n^{th} frame.

The sequence of feature vectors $\boldsymbol{X}$ is extracted from the raw audio time signal $\boldsymbol{x} \in \mathbb{R}^{M \times T}$ with M being the number of microphones and T the number of samples. Feature extraction is

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 101007666, the Agency is not responsible for this results or use that may be made of the information and from the French ANR Extensor (ANR-19-CE23-0001-01)

performed by a function $\boldsymbol{X} = g(\boldsymbol{x})$ which may be handcrafted (e.g. MFCC, MVDR beamforming) or jointly optimized with the parametric model f in an end-to-end manner (e.g. SACC, cSACC). Similarly to [10, 12, 13], the feature extraction applies a downsampling operation to the input data to reduce the computational cost. The overall OSD pipeline is presented in Figure 1.

2.2. Sequence modeling and frame classification

2.2.1. BLSTM

Sequence modeling is first performed using the BLSTM architecture as described in [9, 10]. Two BLSTM layers composed of $P = 256$ cells are stacked. The resulting sequence is post processed using a three-layers feed forward network (FFN) with output sizes $L_1 = 128$, $L_2 = 128$ and $L_3 = 2$ respectively. FFN layers are followed by $\tanh$ activation functions except for the last one. A $\mathrm{softmax}$ activation is applied to the output logits to compute classification probabilities.

2.2.2. TCN

TCN architecture has been proposed as an alternative to BLSTM and FFN for OSD [12]. This architecture is composed of causal convolutional layers with residual connections. The key feature of the TCN is the dilated convolution, allowing to learn a large temporal context. We employed the same TCN architecture as [12] composed of $R = 5$ residual convolutional blocks repeated $P = 3$ times[1]. Classification is performed by a 1-d convolutional layer followed by a $\mathrm{softmax}$ activation function to compute classification probabilities.

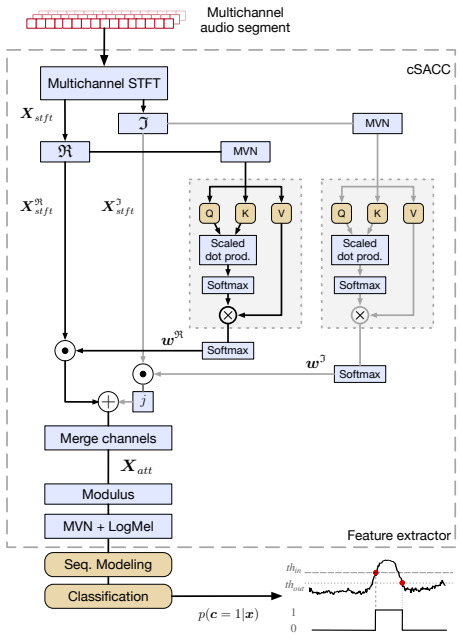


Figure 1: *Flowchart of the OSD task with the proposed cSACC feature extractor. Sequence modeling is performed with BLSTM or TCN architecture. The bottom part of the figure presents the inference procedure: two detection thresholds are applied to the positive-class output of the model to extract a binary sequence.*

[1] https://github.com/popcornell/OSDC

3. Multichannel feature extraction

Combining and weighting channels coming from multiple microphones allows to select spatial information in a multi-microphone input signal (e.g. beamforming [15]). In order to automatically combine multiple channels, we apply the SACC method [22] as a feature extractor for the OSD task. This method is also extended in the complex space (cSACC) to preserve all of the STFT information during the weight-estimation procedure.

3.1. Self-Attention Channel Combinator

Let $\boldsymbol{X}_{stft} \in \mathbb{C}^{M \times N \times K}$ be the multichannel STFT of the input signal $\boldsymbol{x}$, where K is the number of frequency bins. The SACC algorithm computes combination-weights $\boldsymbol{w} \in \mathbb{R}^{M \times N \times 1}$ applied to each STFT channel before combining them. Those weights are determined from the log-magnitude of the STFT using self-attention [23]. Let q, k and v be three linear transformations, mapping the input STFT log-magnitude $\boldsymbol{X}_{log}$ to the query and the key $\boldsymbol{Q}, \boldsymbol{K} \in \mathbb{R}^{M \times N \times D}$ and the value $\boldsymbol{V} \in \mathbb{R}^{M \times N \times 1}$. The combination weights are computed as follows:

$$\boldsymbol{w} = \mathrm{softmax}\left(\mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{D}}\right)\boldsymbol{V}\right), \tag{1}$$

with D being the output dimension of the linear layers mapping the STFT log-magnitude to $\boldsymbol{Q}$ and $\boldsymbol{K}$. The last $\mathrm{softmax}$ activation constrains the weights to be within the interval $[0, 1]$. Mean and Variance Normalization (MVN) is applied on each frequency bin before feeding the self attention module to reduce the variation range of the input data. The combined STFT magnitude $\boldsymbol{X}_{att}$ is finally obtained as the weighted sum of the different channels:

$$\boldsymbol{X}_{att} = \sum_{m=1}^{M} \boldsymbol{w} \odot \|\boldsymbol{X}_{stft}\|. \tag{2}$$

Since channels combination can lead to a large variation in the data, MVN is applied to the combined STFT for each frequency bin, and converted to the log-mel scale using $F = 64$ filters as in [22]. The process of the SACC algorithm is presented in Figure 1. The use of multiple self-attention heads [23] was investigated to compute combination-weights but did not bring significant improvement.

The SACC combination-weights can be visualized as a function of the speaker direction of arrival (DOA) [22]. However, since weights belong to the real space, the lack of information about the phase limits their interpretation. In order to preserve the phase information of the STFT along the process and to better understand weights learned by SACC, we propose its extension in the complex space.

3.2. Complex Self-Attention Channel Combinator

We propose an extension of the SACC algorithm in the complex space (cSACC). This model learns complex weights directly from the incoming multichannel STFT $\boldsymbol{X}_{stft}$. We separate the real and imaginary parts $\boldsymbol{X}^{\Re}_{stft}$ and $\boldsymbol{X}^{\Im}_{stft}$ respectively. Weights are then computed separately on the real part $\boldsymbol{w}^{\Re}$ and the imaginary part $\boldsymbol{w}^{\Im}$ using equation (1). Two self-attention modules are thus needed. The complex combined STFT can be written as:

$$\boldsymbol{X}_{att} = \sum_{m=1}^{M} \boldsymbol{w}^{\Re} \odot \boldsymbol{X}^{\Re}_{stft} + j \times \boldsymbol{w}^{\Im} \odot \boldsymbol{X}^{\Im}_{stft}, \tag{3}$$

where $j = \sqrt{-1}$. MVN is applied to to each frequency bin on both real and imaginary parts of the STFT before computing

combination-weights. The magnitude of the complex weighted STFT is then converted to the log-mel scale using $F = 64$ filters before being fed to the sequence modeling network. Similarly to SACC, MVN is applied before log-mel conversion.

4. Experimental study

4.1. Dataset

Experiments are conducted on the AMI[2] meeting corpus [24] using two types of audio signals: close-talk speech material recorded with the speakers' headset and distant speech signals recorded by the microphone array (Array 1). It consists of a uniform circular array (UCA) composed of $M = 8$ omni-directional microphones placed on the table during meetings. To train and evaluate our models, the AMI corpus is split into training, development and evaluation subsets, containing about 80 h, 10 h and 10 h of human-annotated audio signals respectively. The data partition follows the protocol proposed in [25] which guarantees different speakers across subsets. Overlapped speech binary labels are computed from the manual segment annotation provided. Audio signals are sampled at 16 kHz.

4.2. Baselines

Both SACC and cSACC feature extractors are compared to four different baselines. (**i**) Close talk MFCC: MFCC are extracted from close-talk recordings from the AMI headset mix. This model is referred to as the close-talk reference. (**ii**) Single Distant Microphone (SDM): MFCC are extracted from the signal coming from the first channel of the UCA. (**iii**) Channels sum: time signals coming from each microphone of the UCA are directly merged in the time domain without any weighting scheme. MFCC are then extracted. (**iv**) MVDR beamformer: signals are weighted and merged using the MVDR solution proposed in [16]. We use the recent MVDR implementation from the `torchaudio` toolkit [26]. Since we are dealing with signals recorded under real conditions, speech and noise covariance matrices are estimated using the Coherent-to-Diffuse Ratio (CDR). The CDR is estimated under diffuse noise and unknown DOA conditions [27] (eq. 25). The STFT magnitude of the MVDR beamformed signal is then converted to mel-scale using $F = 64$ filters similarly to the SACC and cSACC approaches.

4.3. Training and evaluation procedures

OSD models are trained on 245k two-second segments randomly sampled from the AMI train set. Features are extracted on 25 ms sliding window with 10 ms shift. The learning rate is set to $l_r = 10^{-3}$ since scheduling schemes did not bring any improvement. Binary cross-entropy is used as a training objective with stochastic gradient descent (SGD) optimizer [28]. To counteract class imbalance, 50% of the training segments are augmented on-the-fly by summing them to another randomly sampled training segment. Associated labels of each segment are also combined [9, 10]. No other data augmentation procedure is applied to make the comparison between monochannel and multichannel approaches easier. Furthermore, multichannel data augmentation (e.g. additional reverberation) was investigated but did not bring significant improvement. This may be due to different locations of the speakers in the real acquisitions and in the room impulse response (RIR) simulations. After training, detection thresholds are tuned on the development set. Model performance is evaluated using F1-score and average precision (AP) [13] and reported on both development and evaluation subset.

[2]https://groups.inf.ed.ac.uk/ami/corpus/

5. Results

Overlapped speech detection performance with the BLSTM architecture are presented in table 1. As expected, the use of a single distant microphone drastically degrades OSD performance with an absolute 23,5% loss on the evaluation F1-score. Direct sum of the channels offers similar performance. MVDR beamforming improves detection by an absolute 22% F1-score gain, probably because it better takes advantage of spatial information. Self-attention based channel combination also reduces the gap between close talk and distant OSD. The SACC and cSACC methods reach about 23% and 19% absolute improvement on the F1-score respectively compared to the SDM configuration. The same behavior is observed on the AP scores with about 19% and 13% improvement in the distant speech scenario. The SACC model slightly outperforms MVDR without requiring to extract explicit information from the multi-microphone input signal (e.g. noise statistics, speakers location). The self-attention mechanism automatically extract useful spatial information for distant OSD from the multichannel audio data. The extension of the SACC approach in the complex space does not improve the results compared to SACC. However, this formulation allows to better interpret the combination-weights learned by the model, as shown in Section 6.

Table 1: *OSD performance on the AMI meeting corpus for each feature extraction method used as BLSTM architecture front-end. Bold value indicates the best model in the distant speech scenario.*

	F1-score (%)		AP (%)	
AMI	Dev	Eval	Dev	Eval
Close talk MFCC	67,9	63,1	71,7	63,6
Single channel MFCC	49,1	39,6	50,6	42,6
Sum channels MFCC	49,1	31,5	42,4	34,2
MVDR	62,7	60,1	67,0	59,6
SACC	**64,8**	**62,4**	**67,9**	**61,6**
cSACC	62,2	58,9	65,0	55,8

Table 2 presents OSD performance with each feature extractor using the TCN architecture. The use of TCN globally improves detection compared to the BLSTM architecture. For example, AP is improved by 7,6% in the close-talk scenario and by 12% in the SDM scenario. Furthermore, SACC and cSACC still improve detection performance compared to SDM, reaching similar performance as MVDR beamforming. The score gap between SACC and cSACC still appears with TCN sequence modeling. The difference in performance with cSACC could be explained by the fact that the real and imaginary parts of the weight are learned separately. To tackle this, complex-valued deep neural networks [29, 30] or learnable analytic filterbank [31] are going to be investigated. Preliminary work has also shown that the use of a different optimizer could improve the performance with cSACC architecture.

6. Analysis

Hereafter, we conduct an analysis of the combination-weights learned by the cSACC algorithm. The analysis shows that the model is focusing on the speakers' direction when overlapped speech is detected.

Table 2: *OSD performance on the AMI meeting corpus for each feature extraction method used as TCN architecture front-end.*

	F1-score (%)		AP (%)	
AMI	Dev	Eval	Dev	Eval
Close talk MFCC	71,6	68,1	76,7	71,2
Single channel MFCC	62,0	48,5	59,4	54,6
Sum channels MFCC	57,0	46,6	61,3	56,2
MVDR	**68,3**	**64,2**	72,4	**66,1**
SACC	**68,2**	**64,2**	**72,7**	64,3
cSACC	65,0	60,8	68,7	60,1

6.1. Beampattern

The response of standard spatial filters, the so-called beampattern, can be computed based on filter weights and the array geometry [15, 18]. The magnitude of the beampattern represents the gain applied in every θ azimuth angular direction. Hence, the beampattern computed on a set of cSACC combination-weights $\boldsymbol{w}_n$ provides information about the directions in which the attention is focused.

Let $\psi_m = 2\pi(m-1)/M$ be the m^{th} microphone angular position and r the radius of the UCA. The beampattern is defined as [18]:

$$\mathcal{B}_n[\boldsymbol{w}_n, \theta] = \sum_{m=1}^{M} w_{m,n}^{*} \mathrm{e}^{j\bar{\omega}\cos(\theta-\psi_m)}, \tag{4}$$

with $\bar{\omega} = \omega r/c$ and $c = 340$ m/s being the speed of the sound. The n subscript represents the frame-index since combination-weights are time-variant.

The Time-Averaged Beampattern (TAB) can be computed from (4). It represents the average steering direction of the cSACC algorithm on a given time window of length N_T. The TAB can be formulated as follows:

$$\hat{\mathcal{B}} = \frac{1}{N_T} \sum_{n=0}^{N_T-1} \mathcal{B}_n[\boldsymbol{w}_n(\omega, \theta_s), \theta]. \tag{5}$$

6.2. Simulated utterances

The analysis of the cSACC combination-weights is performed on simulated data to analyze them in a controlled environment. First, clean utterances are taken from the Librispeech dataset [32]. We simulate a $L = 5$ m $\times$ $l = 4$ m $\times$ $h = 3$ m room with $T_{60} = 0.8$ s reverberation time. RIRs are then computed between two sources and an 8-microphone UCA of $r = 5$ cm radius using the gpuRIR toolkit [33]. Finally, each clean utterance from each speaker is convolved with its own set of RIRs. Convolved signals from each location are delayed from each other and summed to generate the artificial distant overlapped speech.

6.3. cSACC combination-weights analysis

The cSACC combination-weights are analyzed within the best performing TCN based OSD architecture. An example of beampattern magnitude computed on those weights is proposed in Figure 2. The beampattern is presented as a heat-map function of time and DOA. DOA associated with higher amplitude indicates the steering direction of the cSACC algorithm. This figure shows that the model is switching attention between the angular direction of each speaker.

Two examples of TAB for two active speakers are presented in Figure 3. The TAB is only computed on time segments where

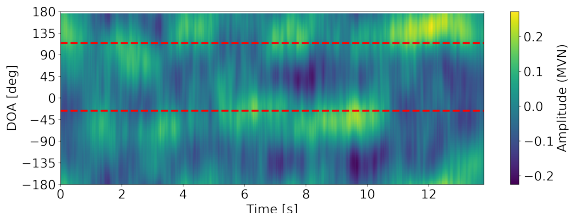
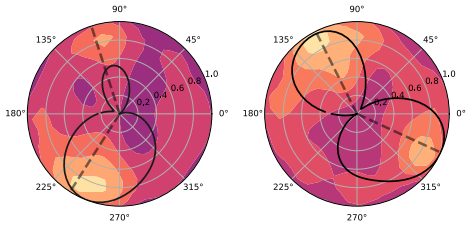


Figure 2: *Beampattern magnitude computed from* (4) *as a function of time and DOA for two active speakers located in different angular locations (- - -). Frame-wise MVN has been applied as well as a 100 frames moving average for better readability.*

the output probability of the model verifies $p(\boldsymbol{c} = 1|\boldsymbol{x}) > 0.6$. Hence, the main lobes of the TAB informs us on the angular directions where cSACC draws attention when overlapped speech is detected. The utterance-averaged SRP-PHAT [15] is also presented as a heatmap. It has been computed on a circular plane of two meters radius and allows to compare the cSACC steering directions to the distribution of the acoustic energy. Figure 3 shows that, over the utterance, cSACC model steers towards the direction of the two active speakers for these two examples. A quantitative study may be conducted to confirm these observations.



(a) (b)

Figure 3: *TAB computed from* (5) *(—) on two overlapped speech utterances compared to SRP-PHAT energy map and ground-truth (- - -) speakers location with two simultaneously active speakers. TAB is normalized between* $[0, 1]$ *for better visualization.*

7. Conclusion

In this paper, we investigated the use of self-attention channel combinators as distant overlapped speech detection front-end. We showed that self attention-based algorithms reach state-of-the-art performance without requiring handcrafted feature extraction. Furthermore, the original self-attention channel combinator (SACC) [22] slightly outperforms MVDR beamformer without requiring to estimate speech signal statistics. The SACC model was also extended in the complex space to preserve all of the information in the STFT. Even if this approach does not improve detection performance, it allows a better interpretation of the learned combination-weights. Combination-weight analysis showed that the model seems to draw attention to the angular directions of the active speakers.

Further work will be conducted on the use of self-attention to combine channels in a full diarization pipeline in order to evaluate the benefits of this kind of approach on the overall task. Other signal representations such as learnable filterbank [31] or pre-trained models (e.g. WavLM [34]) will be investigated as an alternative to the STFT.

8. References

[1] N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman, "The Second DIHARD Diarization Challenge: Dataset, Task, and Baselines," in *Interspeech*, 2019, pp. 978–982.

[2] N. Ryant, P. Singh, V. Krishnamohan, R. Varma, K. Church, C. Cieri, J. Du, S. Ganapathy, and M. Liberman, "The Third DIHARD Diarization Challenge," in *Interspeech*, 2021, pp. 3570–3574.

[3] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe, and S. Khudanpur, "Diarization is Hard: Some Experiences and Lessons Learned for the JHU Team in the Inaugural DIHARD Challenge," in *Interspeech*, 2018, pp. 2808–2812.

[4] L. P. Garcia Perera, J. Villalba, H. Bredin, J. Du, D. Castan, A. Cristia, L. Bullock, L. Guo, K. Okabe, P. S. Nidadavolu, S. Kataria, S. Chen, L. Galmant, M. Lavechin, L. Sun, M.-P. Gill, B. Ben-Yair, S. Abdoli, X. Wang, W. Bouaziz, H. Titeux, E. Dupoux, K. A. Lee, and N. Dehak, "Speaker Detection in the Wild: Lessons Learned from JSALT 2019," in *The Speaker and Language Recognition Workshop (Odyssey 2020)*, 2020, pp. 415–422.

[5] J. T. Geiger, F. Eyben, B. Schuller, and G. Rigoll, "Detecting overlapping speech with long short-term memory recurrent neural networks," in *Interspeech*, 2013, pp. 1668–1672.

[6] V. Andrei, H. Cucu, and C. Burileanu, "Detecting overlapped speech on short timeframes using deep learning." in *Interspeech*, 2017, pp. 1198–1202.

[7] N. Sajjan, S. Ganesh, N. Sharma, S. Ganapathy, and N. Ryant, "Leveraging lstm models for overlap detection in multi-party meetings," in *ICASSP*, 2018, pp. 5249–5253.

[8] J. weon Jung, H.-S. Heo, Y. Kwon, J. S. Chung, and B.-J. Lee, "Three-Class Overlapped Speech Detection Using a Convolutional Recurrent Neural Network," in *Interspeech*, 2021, pp. 3086–3090.

[9] L. Bullock, H. Bredin, and L. P. Garcia-Perera, "Overlap-Aware Diarization: Resegmentation Using Neural End-to-End Overlapped Speech Detection," in *ICASSP*, 2020, pp. 7114–7118.

[10] H. Bredin, R. Yin, J. M. Coria, G. Gelly, P. Korshunov, M. Lavechin, D. Fustes, H. Titeux, W. Bouaziz, and M.-P. Gill, "Pyannote.Audio: Neural Building Blocks for Speaker Diarization," in *ICASSP*, 2020, pp. 7124–7128.

[11] Z. Chen, X. Xiao, T. Yoshioka, H. Erdogan, J. Li, and Y. Gong, "Multi-channel overlapped speech recognition with location guided speech extraction network," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 558–565.

[12] S. Cornell, M. Omologo, S. Squartini, and E. Vincent, "Detecting and Counting Overlapping Speakers in Distant Speech Scenarios," in *Interspeech*, 2020, pp. 3107–3111.

[13] ——, "Overlapped speech detection and speaker counting using distant microphone arrays," *Computer Speech & Language*, vol. 72, p. 101306, 2022. [Online]. Available: https://doi.org/10.1016/j.csl.2021.101306

[14] S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," *arXiv:1803.01271 [cs]*, 2018.

[15] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*. Springer, 2008.

[16] M. Souden, J. Benesty, and S. Affes, "On optimal frequency-domain multichannel linear filtering for noise reduction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 2, pp. 260–276, 2010.

[17] S. Applebaum, "Adaptive arrays," *IEEE Transactions on Antennas and Propagation*, vol. 24, no. 5, pp. 585–598, 1976.

[18] J. Benesty, J. Chen, and I. Cohen, *Design of Circular Differential Microphone Arrays*, ser. Springer Topics in Signal Processing. Cham: Springer International Publishing, 2015, vol. 12.

[19] B. Li, T. N. Sainath, R. J. Weiss, K. W. Wilson, and M. Bacchiani, "Neural network adaptive beamforming for robust multichannel speech recognition," 2016.

[20] W. Minhua, K. Kumatani, S. Sundaram, N. Ström, and B. Hoffmeister, "Frequency domain multi-channel acoustic modeling for distant speech recognition," in *ICASSP*, 2019, pp. 6640–6644.

[21] T. Park, K. Kumatani, M. Wu, and S. Sundaram, "Robust multi-channel speech recognition using frequency aligned network," in *ICASSP*, 2020, pp. 6859–6863.

[22] R. Gong, C. Quillen, D. Sharma, A. Goderre, J. Laínez, and L. Milanović, "Self-Attention Channel Combinator Frontend for End-to-End Multichannel Far-Field Speech Recognition," in *Interspeech*, 2021, pp. 3840–3844.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, 2017, p. 6000–6010.

[24] I. Mccowan, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, D. Reidsma, and P. Wellner, "The ami meeting corpus," in *In: Proceedings Measuring Behavior 2005, 5th International Conference on Methods and Techniques in Behavioral Research.*, 2005.

[25] F. Landini, J. Profant, M. Diez, and L. Burget, "Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: Theory, implementation and analysis on standard tasks," *Computer Speech & Language*, vol. 71, p. 101254, 2022.

[26] Y.-Y. Yang, M. Hira, Z. Ni, A. Chourdia, A. Astafurov, C. Chen, C.-F. Yeh, C. Puhrsch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Mahadeokar, J. Hwang, J. Chen, P. Goldsborough, P. Roy, S. Narenthiran, S. Watanabe, S. Chintala, V. Quenneville-Bélair, and Y. Shi, "Torchaudio: Building blocks for audio and speech processing," *arXiv preprint arXiv:2110.15018*, 2021.

[27] A. Schwarz and W. Kellermann, "Coherent-to-Diffuse Power Ratio Estimation for Dereverberation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 6, pp. 1006–1018, 2015.

[28] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," *The Annals of Mathematical Statistics*, pp. 462–466, 1952.

[29] Y.-S. Lee, C.-Y. Wang, S.-F. Wang, J.-C. Wang, and C.-H. Wu, "Fully complex deep neural network for phase-incorporating monaural source separation," in *ICASSP*, 2017. [Online]. Available: https://doi.org/10.1109/icassp.2017.7952162

[30] H.-S. Choi, J.-H. Kim, J. Huh, A. Kim, J.-W. Ha, and K. Lee, "Phase-aware speech enhancement with deep complex u-net," in *International Conference on Learning Representations*, 2018.

[31] M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, "Filterbank design for end-to-end speech separation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6364–6368.

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

[33] 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, no. 4, pp. 5653–5671, 2021.

[34] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *arXiv preprint arXiv:2110.13900*, 2021.