Interspeech 2024
1-5 September 2024, Kos, Greece

Graph Attention Based Multi-Channel U-Net for Speech Dereverberation With Ad-Hoc Microphone Arrays

Hongmei Guo[*1,2,3], *Yijiang Chen*[*1], *Xiao-Lei Zhang*[1,2,3], *Xuelong Li*[2]

[1]School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China
[2]Institute of Artificial Intelligence (TeleAI), China Telecom, China
[3]Research & Development Institute of Northwestern Polytechnical University in Shenzhen, China

guohongmei@mail.nwpu.edu.cn, orangechen@mail.nwpu.edu.cn, xiaolei.zhang@nwpu.edu.cn

Abstract

Speech dereverberation with ad-hoc microphone arrays seems not studied sufficiently, particularly in the scenario where the reverberation time is large. In this paper, we propose a novel multi-channel U-Net model for speech dereverberation with ad-hoc microphone arrays, where an attention module is integrated into the model in an end-to-end training manner to do channel selection and fusion. Specifically, we first train a single-channel U-Net model. Then, we replicate the U-Net model to each channel. Finally, we train the attention module for aggregating the information of the channels, where the parameters of the U-Net model are fixed at this stage. To our knowledge, this is the first work that U-Net was used for dereverberation with ad-hoc microphone arrays. We studied two attention mechanism, which are the self-attention and graph-attention; moreover, we integrated the attention module into either the bottleneck layer or the output layer of the multi-channel U-Net, which results in four implementations.

Experimental results demonstrate that the proposed method achieves the state-of-the-art performance, and the attention module is very important in channel selection and fusion for improving the performance against long reverberation time.

Index Terms: Speech dereverberation, ad-hoc microphone arrays, attention mechanism

1. Introduction

The reverberation of speech is the speech signal that reaches a microphone after many reflections from obstacles, such as walls. Reverberation degrades the clarity and comprehensibility of speech. It does great harm to intelligent speech systems, such as speech recognition and speaker recognition. Therefore, we need to do dereverberation. Dereverberation aims to remove the reverberant component from the noisy speech signal at the microphone receiver, keeping only the direct speech for applications. According to the number of microphones in an array, the speech dereverberation approaches can be categorized into single-channel-based and multi-channel-based ones. Compared to the single-channel approaches, the multi-channel methods are able to leverage abundant spatial information for better performance [1, 2, 3], which is the focus of this paper.

Currently, one prevalent way of multi-channel dereverberation is to integrate deep learning with the conventional hand-crafted models that are designed to mathematically simulate the reverberation process, e.g. spatial filters [4]. For instance, the DESNet architecture employs a deep neural network (DNN)-based weighted prediction error (WPE) module for dereverberation [5]. An alternative strategy is to employ DNN to calculate the weights of a beamformer, such as the MC-CSM [6], EaBNet [7], PCG-AIID System [2], TPARN [8] and FasNet-TAC [9]. The spatial patterns of the microphone array between the training and test stages are consistent, which make the learned beamforming weights applicable in the test stage.

However, the aforementioned methods were designed for fixed microphone arrays, where all microphones are contained in a single device. The way of grouping multiple distributed devices into an ad-hoc network, which is named the ad-hoc microphone array, has received some attention beyond the conventional fixed microphone arrays [10, 11, 12]. A core property of the ad-hoc microphone array is that the devices can be placed randomly and flexibly without having to know their positions. However, this topic seems not developed sufficiently, particularly on speech dereverberation. To our knowledge, existing

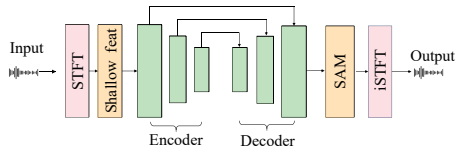
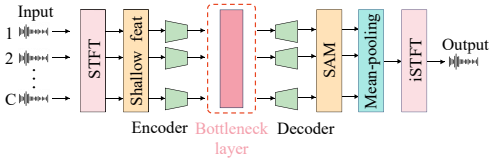
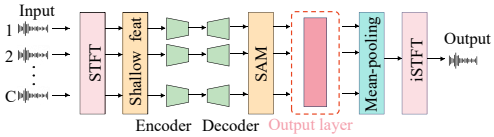


Figure 1: *Single-channel dereverberation model.*



(a) Attention module integrated into the bottleneck layer of the multi-channel U-Net Model.



(b) Attention module integrated into the output layer of the multi-channel U-Net Model.

Figure 2: *The proposed multi-channel U-Net architecture for ad-hoc array speech dereverberation.*

∗ Equal contribution.
Xiao-Lei Zhang is the corresponding author.

This work was supported in part by the National Science Foundation of China (NSFC) under Grant 62176211, and in part by the Project of the Science, Technology, and Innovation Commission of Shenzhen Municipality, China under Grant JCYJ20210324143006016 and JSGG20210802152546026

10.21437/Interspeech.2024-336

works mostly focused on speech enhancement with limited reverberation time, e.g. no more than 0.6 second [13, 14, 15]. [13] presents a method for multi-channel speech enhancement using graph neural networks. [14] presents the Tango algorithm, which uses ad-hoc nodes to compute and transmit compressed signals, and then uses both local and compressed signals to estimate the desired signal. [15] integrates traditional signal processing techniques with deep neural networks to handle unconstrained microphone arrays with varying node numbers.

In this paper, we propose an end-to-end multi-channel U-Net-based speech dereverberation method for ad-hoc microphone arrays. The novelties of the proposed method lie in the following respects. (i) It is, to our knowledge, the first U-Net-based speech dereverberation model for ad-hoc microphone arrays. The reason why we adopt U-Net is that U-Net has been proven to be an effective model in single-channel speech dereverberation. (ii) An attention module, which is used to aggregate information of the channels at the both time and spatial dimensions, is integrated into the multi-channel U-Net model smoothly, which leads to an end-to-end model. The attention module is responsible for channel selection and fusion. (iii) We have used two kinds of attention module for the information aggregation, i.e. self-attention and graph-attention, and integrated it either into the bottleneck layer of the multi-channel U-Net model or the output layer, which results in four implementations of the proposed model.

Additionally, the proposed model doesn't constrain the number of channels or spatial pattern of an ad-hoc microphone array, making it flexible in real-world scenarios. Extensive experimental results demonstrate the effectiveness of the proposed model in challenging scenarios with long reverberation time.

2. Proposed method

2.1. Signal model

Suppose an ad-hoc microphone array contains C randomly distributed microphones. For the c-th microphone, given its corresponding room impulse response (RIR) $h_c(t) = h_{cd}(t) + h_{cr}(t)$ where t denotes time, and $h_{cd}(t)$ and $h_{cr}(t)$ represent the impulse response functions for direct sound and reverberation respectively. The reverberant speech signal at the microphone can be described as:

$$\begin{aligned} y_c(t) &= s(t) * h_c(t) = s(t) * h_{cd}(t) + s(t) * h_{cr}(t) \\ &= x_c(t) + r_c(t) \end{aligned} \tag{1}$$

where $*$ represents the convolution operation, $s(t)$ represents the clean speech at the source point, $y_c(t)$ represents the reverberant speech, $x_c(t)$ denotes the direct sound at the c-th microphone, and $r_c(t)$ denotes the reverberant noise of $y_c(t)$. Speech dereverberation with ad-hoc microphone arrays is to get the direct speech of any channel $x_c(t)$, which is an amplification of the clean speech $s(t)$.

2.2. Model architecture

The proposed method is an end-to-end multi-channel dereverberation model. The novelties of the model lie in that (i) it generalizes a single-channel U-Net model [16] to the multi-channel case for ad-hoc microphone arrays, and then (ii) integrates an attention module into the multi-channel U-Net in an end-to-end manner. The architecture of the proposed model is show in Figure 2. It is a method working in the time-frequency domain where the magnitude spectrogram of short-time Fourier transform (STFT) is used as the acoustic feature. The training of the proposed model contains two-stages:

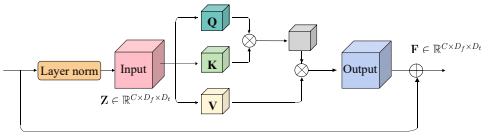


Figure 3: *Aggregation module based on self-attention*

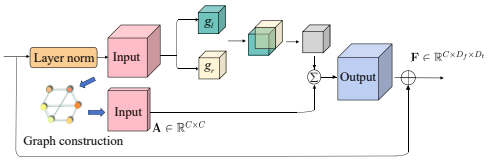


Figure 4: *Aggregation module based on graph attention*

The first stage is to train a single-channel U-Net-based dereverberation model [16] as shown in Figure 1, which consists of a shallow feat, an encoder, a decoder and a Spatial Attention Module (SAM). See [16] for the details.

In the second stage, we replicate the single-channel model in the first stage to each channel of the ad-hoc microphone array, which yields a multi-channel U-Net model. Then, we add an attention module, in the places of either at the bottleneck layer in Figure 2(a) or the output layer in Figure 2(b), to learn channel-invariant feature for all channels, and finally average the dereverberant features of all channels which yields a single-channel dereverberant speech. In this training stage, we keep the parameters of the multi-channel U-Net models fixed, and focus on using the ad-hoc data to train the information-exchange module only. We have developed two implementations of the attention module: (i) SA-aggregation, which is a multi-head attention module, and GAT-aggregation, which is based on the graph attention mechanism. The two implementations are introduced as follows.

2.3. Information aggregation based on self-attention

The self-attention mechanism aims to capture global dependencies and has been successfully utilized in various deep learning domains such as speech recognition [17] and speech separation [18]. In our architecture, we incorporate the SA-aggregation module to facilitate the fusion of inter-channel information by learning temporal-spatial characteristics among different channels. Figure 3 demonstrates the architecture of the self-attention module used in Figure 2. For the integrity of this paper, we describe the self-attention module as follows.

Given the input of the self-attention module, denoted as $\mathbf{Z} = [\mathbf{Z_0}, \cdots, \mathbf{Z_{D_t}}] \in \mathbb{R}^{C\times D_f\times D_t}$, which is also the output of the encoder or SAM. For the m-th attention head $\forall m = 1, \ldots, M$, the query ($\mathbf{Q}_t^m$), key ($\mathbf{K}_t^m$), and value ($\mathbf{V}_t^m$) subspaces of dimension E are computed through convolution. They are all in $\mathbb{R}^{C\times d_m}$, where $d_m = E/M$, and M denotes the number of attention heads. Subsequently, the output of is computed as follows:

$$\mathbf{H}_t^m = \text{softmax}\left(\frac{\mathbf{Q}_t^m \cdot (\mathbf{K}_t^m)^\top}{\sqrt{d_m}}\right)\mathbf{V}_t^m \tag{2}$$

Table 1: *Experimental results of the single channel dereverberation model [16]*

	STOI	PESQ	fwSegSNR
Reverberant speech	0.692	2.106	6.232
Single channel Unet	0.848	2.547	9.395

with $\mathbf{H}_t^m \in \mathbb{R}^{C \times d_m}$. The outputs of all attention heads are:

$$\mathbf{F}_t = \mathbf{Z}_t + \text{concat}\left[\mathbf{H}_t^1, \mathbf{H}_t^2, \ldots, \mathbf{H}_t^M\right] \mathbf{W} \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{E \times D_f}$ is the weight matrix of the linear projection layer, and $\mathbf{F}_t \in \mathbb{R}^{C \times D_f}$. To prevent gradient vanishing, a residual connection is established.

2.4. Information aggregation based on graph attention

As shown in Figure 4, another information aggregation module is the GAT-aggregation. GAT-aggregation uses a self-attention method based on graph convolutional network (GCN) layers to learn the attention weights for the channels of a graph. The adjacent matrix $\mathbf{A} \in \mathbb{R}^{C \times C}$, constructed by the graph, enables each channel to focus on other channels by utilizing its own representation as the query. Here, we assume that all channels are mutually correlated:

$$\mathbf{A}_{[c,j]} = 1, \quad \forall c = 1, \ldots, C, \quad \forall j = 1, \ldots, C \tag{4}$$

Given the input of the GAT module, denoted as $\mathbf{Z}_t$, where $\mathbf{Z}_t \in \mathbb{R}^{C \times D_f}$, for the m-th attention head, we initiate the process by projecting into a d_m dimensional space using learnable parameters $\mathbf{W}_l^m \in \mathbb{R}^{D_f \times d_m}$ and $\mathbf{W}_r^m \in \mathbb{R}^{D_f \times d_m}$. In this context, we denote the query and key matrices as $\mathbf{g}_l^m$ and $\mathbf{g}_l^m$.

$$\mathbf{g}_l^m = \mathbf{Z}_t \mathbf{W}_l^m, \mathbf{g}_r^m = \mathbf{Z}_t \mathbf{W}_r^m \tag{5}$$

The score for the query-key pair from channel c to channel j is computed using the following formula:

$$\mathbf{E}^m[c,j] = \alpha^\top \text{LeakyReLU}(\text{concat}(\mathbf{g}_{lc}^m, \mathbf{g}_{rj}^m)) \tag{6}$$

$$a_{cj}^m = \frac{\exp\left(\mathbf{E}^m[c,j]\right)}{\sum_{\mathbf{A}[c,j]=1} \exp\left(\mathbf{E}^m[c,j]\right)} \tag{7}$$

where $\mathbf{E}^m \in \mathbb{R}^{C \times C}$, and $\alpha \in \mathbb{R}^{2d_k}$ is a learnable vector. The softmax function is used to normalize the attention ratings over all of the adjacent channels. The aggregated output of channel c is represented by $\mathbf{h}_c^m$ for the m-th head are: $\mathbf{h}_c^m = \sum_{j=1}^C a_{cj}^m \mathbf{g}_{rj}^m$.

The aggregated characteristics of all nodes are concatenated. Then, we obtain the output of the aggregation as follows:

$$\mathbf{H}_t^m = [\mathbf{h}_1^m, \ldots, \mathbf{h}_c^m, \ldots, \mathbf{h}_C^m] \tag{8}$$

$$\mathbf{F_t} = \mathbf{Z}_t + \text{concat}[\mathbf{H}_t^1, \ldots, \mathbf{H}_t^m, \ldots, \mathbf{H}_t^M] \tag{9}$$

3. Experiments

3.1. Datasets

We constructed a simulated LibriSIMU-reverb dataset from Librispeech[22]. Specifically, we randomly selected 10000, 3000 and 2000 utterances from the 'train-clean-100', 'dev-clean' and 'test-clean' subsets of Librispeech respectively, as the speech source for constructing the training, validation, and test datasets respectively. For each utterance, we simulated a room with its length randomly generated from a range of of $[12, 14]$ meters, width from $[8, 10]$ meters, and height from $[3, 5]$ meters. The reverberation time T_{60} of the room was randomly generated from a range of $[0.2, 1.2]$ seconds. The gpuRIR [23] was used to generate the room impulse response function. The positions of the single speaker and a number of microphones were sampled randomly within the room, where the training data uses ad-hoc microphone arrays of 8 microphones, while the number of microphones in the test data were set to 4, 8, 12, and 16.

3.2. Experimental setup

When the attention module is at the bottleneck layer, we denote the proposed method based on the SA-aggregation as **UNet-Bottleneck-SA**, and the method based on GAT-aggregation as **UNet-Bottleneck-GAT**. Similarly, when the attention module is at the output layer, we denote the proposed methods as **UNet-Output-SA** and **UNet-Output-GAT** respectively.

For each proposed model, the information aggregation module contains two spatial-temporal blocks, each of which has four attention heads. To train the proposed model, we first train the single-channel U-Net dereverberation model [16] by 100 epochs with randomly selected single-channel data from the training set of the LibriSIMU-reverb dataset. The best model among the 100 epochs was selected to initialize the proposed multi-channel U-Net model. The performance of the single-channel dereverberation model are listed in Table 1. In the second training stage, we trained the attention module by 100 training epochs with 8 randomly selected channels of each training utterance. The batch size at the second stage was set to 3. We used the short-time objective intelligibility (STOI) [24], perceptual evaluation of speech quality (PESQ) [25], and frequency-weighted segmental signal-to-noise ratio (fwSegSNR) [26] to evaluate the dereverberation performance.

3.3. Comparison methods

The comparison baselines are categorized into three classes. The first class is a conventional signal processing method **weighted prediction error (WPE)** [19], which is implemented by the NARA-WPE algorithm, a highly recognized traditional dereverberation method, without constraints on the number of channels or array designs.

The second class is single-channel U-Net based methods. (i) **Oracle one-best**: It first selects physically the nearest microphone to the speaker, under the assumption that the microphone positions are known as a prior. Then, it uses the single-channel U-Net to do dereverberation on the selected channel. (ii) **Beamforming** [21]: It first uses the conventional delay-and-sum algorithm to fuse all channels into a single channel, where GCC-PHAT is used to estimate the time delay, and then uses the single-channel U-Net to do dereverberation on the single channel output. (iii) **EV** [20]: It chooses the microphone whose received signal has the highest envelope variance among all microphones, and then uses the single-channel U-Net to do dereverberation.

The third class of comparison methods is multi-channel deep learning methods for ad-hoc microphone arrays. (i) **FaSNet-TAC:** It is a representative multi-channel speech separation model for ad-hoc microphone arrays [9]. (ii) **Mean pooling:** It assigns equal weights to the outputs of all channels in the proposed multi-channel U-Net instead of using the attention mechanism.

Table 2: *Results of the comparison methods on different numbers of microphones*

Algorithm	8Mic			4Mic			12Mic			16Mic		
	STOI	**PESQ**	**fwSegSNR**	**STOI**	**PESQ**	**fwSegSNR**	**STOI**	**PESQ**	**fwSegSNR**	**STOI**	**PESQ**	**fwSegSNR**
WPE [19]	0.758	2.264	7.704	0.736	2.191	6.928	0.775	2.323	8.326	0.786	2.364	8.643
Oracle one-best	0.867	2.543	9.362	0.865	2.436	9.070	0.869	2.559	9.468	0.872	2.588	9.214
EV [20]	0.854	2.524	9.299	0.855	2.527	9.319	0.848	2.567	10.079	0.851	2.518	9.258
Beamforming [21]	0.849	**2.639**	8.482	0.840	**2.795**	8.676	0.840	**2.650**	8.676	0.842	2.576	8.523
FaSNet-TAC [9]	0.787	2.285	8.528	0.785	2.282	8.508	0.789	2.285	8.529	0.789	2.286	8.532
Mean pooling	0.850	2.639	9.482	0.846	2.695	9.276	0.840	2.560	9.367	0.843	2.553	9.323
UNet-Bottleneck-SA	**0.887**	2.600	9.635	**0.882**	2.626	9.469	**0.891**	2.612	9.789	**0.892**	2.614	9.823
UNet-Bottleneck-GAT	**0.888**	2.599	**9.643**	0.879	2.571	**9.619**	0.888	2.606	**10.072**	0.889	2.608	**10.104**
UNet-Output-SA	**0.888**	2.610	9.536	**0.883**	2.597	9.371	**0.891**	2.622	9.687	**0.893**	**2.626**	9.799
UNet-Output-GAT	0.877	2.503	9.144	0.874	2.483	9.009	0.879	2.512	9.352	0.882	2.516	9.783

Table 3: *Results of the comparison methods on the reverberant scenarios with different T_{60} values.*

Algorithm	0.2s-0.4s			0.4s-0.6s			0.6s-0.8s			0.8s-1.0s			1.0s-1.2s		
	STOI	**PESQ**	**fwSegSNR**	**STOI**	**PESQ**	**fwSegSNR**	**STOI**	**PESQ**	**fwSegSNR**	**STOI**	**PESQ**	**fwSegSNR**	**STOI**	**PESQ**	**fwSegSNR**
Reverberant speech	0.831	2.672	8.884	0.718	2.119	6.561	0.658	1.903	5.547	0.595	1.714	4.565	0.551	1.590	3.821
WPE [19]	**0.941**	**3.283**	**12.569**	0.843	2.378	8.524	0.735	2.012	6.448	0.680	1.794	5.118	0.629	1.629	4.572
Oracle one-best	0.891	2.787	10.201	0.896	2.859	10.273	0.877	2.647	9.577	0.832	2.360	8.573	0.829	2.356	8.825
EV [20]	0.910	3.066	11.165	0.875	2.641	9.505	0.850	2.429	8.765	0.833	2.341	8.725	0.809	2.181	7.626
Beamforming [21]	0.917	2.988	11.780	0.848	2.353	8.537	0.775	2.077	6.814	0.722	1.857	6.134	0.674	1.677	4.929
FaSNet-TAC [9]	0.849	2.499	7.788	0.752	2.073	6.324	0.667	1.856	5.042	0.617	1.745	4.107	0.578	1.635	3.750
Mean pooling	0.918	2.426	11.188	0.862	2.532	9.234	0.827	2.186	8.023	0.814	2.053	7.562	0.786	1.875	7.021
UNet-Bottleneck-SA	0.928	3.158	12.083	**0.903**	2.724	9.941	0.879	2.551	9.143	**0.865**	2.421	9.112	0.842	2.290	8.139
UNet-Bottleneck-GAT	0.928	3.151	12.455	**0.903**	**2.724**	**10.528**	**0.880**	**2.550**	**9.567**	**0.866**	2.438	**9.423**	**0.844**	2.293	**8.491**
UNet-Output-SA	0.928	3.165	12.002	**0.903**	**2.743**	9.761	**0.880**	**2.570**	9.001	**0.866**	**2.456**	9.169	**0.845**	**2.308**	8.034
UNet-Output-GAT	0.915	3.134	12.162	0.894	2.707	10.289	0.871	2.546	9.455	0.857	2.444	9.141	0.839	2.302	8.491

3.4. Main results

Table 2 lists the performance of the comparison methods on the LibriSIMU-reverb test set, where the test scenario "8Mic" matches with the training data. From the result, we see that the proposed methods achieve the best performance in terms of STOI and fwSegSNR in all test scenarios, and perform slightly worse than "beamforming" in terms of PESQ when the number of microphones is smaller than 16. The experiment found that 'channel selection' is of significant importance. Specifically, "oracle one-best" outperforms "EV" and "beamforming" in terms of STOI and fwSegSNR, while "EV" behaves better than "beamforming" where beamforming does not utilize channel selection. Moreover, the proposed methods outperform "mean pooling", which also demonstrate the importance of the attention module in channel selection and fusion.

We also see the strong generalization ability of the proposed methods in mismatching test scenarios. Specifically, the results on the mismatched test scenarios with 4, 12, and 16 microphones behave similarly with those with the matched scenario with 8 microphones.

Finally, we find that the four proposed variants behave similarly. If we have to rank the four methods, "UNet-Bottleneck-SA" and "UNet-Bottleneck-GAT" behave similarly and outperform the other two, which indicates that placing the attention module in the bottleneck layer is better than the other choice, and that the graph attention and self-attention behave similar.

3.5. Effects of reverberant time on performance

Table 3 lists the effects of different reverberant time on the performance of the comparison methods. From the table, we see that, the proposed methods outperform all comparison methods in all evaluation metrics when the reverberant time is longer than 0.4 second. The only case that the proposed methods perform worse than the conventional WPE is when the reverberant time is shorter than 0.4 second. Comparing Table 2 with Table 3, we see that the reason why the average PESQ performance of WPE is better than that of the proposed methods in Table 2 is just caused by its outstanding performance in the scenario with little reverberation.

We also observe that, when the reverberant time increases, the performance of all comparison methods drop, however, the decrease rates of the comparison methods are different. The methods with channel selection, including "oracle one-best", "EV", and the proposed methods, drop much slower than the methods without a channel selection module. Particularly, when the reverberant time is large, we see that (i) the channel selection and weighted fusion demonstrates its exceptionally importance in improving the performance, and (ii) the attention module is very important when comparing the results of "mean pooling" and the proposed methods.

4. Conclusion

In this paper, we have proposed a multi-channel U-Net model for dereverberation with ad-hoc microphone arrays, where the attention module was used to do channel selection and fusion. The core novelty is that proposed method generalizes the single-channel U-Net dereverberation model to ad-hoc microphone arrays, which is to our knowledge the first U-Net based dereverberation model for ad-hoc arrays. We also integrate the attention module into the multi-channel U-Net model in an end-to-end training manner, where two variants of attention mechanism was studied. The comparison results show that, the proposed multi-channel U-Net model yields outstanding performance in speech dereverberation with ad-hoc microphone arrays, and the attention module is very important in channel selection and fusion, particularly in the challenging scenarios with large reverberant time.

5. References

[1] A. Pandey, B. Xu, A. Kumar, J. Donley, P. Calamia, and D. Wang, "Tparn: Triple-path attentive recurrent network for time-domain multichannel speech enhancement," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6497–6501.

[2] J. Li, Y. Zhu, D. Luo, Y. Liu, G. Cui, and Z. Li, "The pcg-aiid system for l3das22 challenge: Mimo and miso convolutional recurrent network for multi channel speech enhancement and speech recognition," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 9211–9215.

[3] S. LV, Y. Fu, Y. JV, L. Xie, W. Zhu, W. Rao, and Y. Wang, "Spatial-dccrn: Dccrn equipped with frame-level angle feature and hybrid filtering for multi-channel speech enhancement," in *2022 IEEE Spoken Language Technology Workshop (SLT)*, 2023, pp. 436–443.

[4] M. Togami, "End to end learning for convolutive multi-channel wiener filtering," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 8032–8036.

[5] Y. Fu, J. Wu, Y. Hu, M. Xing, and L. Xie, "Desnet: A multi-channel network for simultaneous speech dereverberation, enhancement and separation," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, 2021, pp. 857–864.

[6] K. Tan, Z.-Q. Wang, and D. Wang, "Neural spectrospatial filtering," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 605–621, 2022.

[7] A. Li, W. Liu, C. Zheng, and X. Li, "Embedding and beamforming: All-neural causal beamformer for multichannel speech enhancement," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6487–6491.

[8] A. Pandey, B. Xu, A. Kumar, J. Donley, P. Calamia, and D. Wang, "Multichannel speech enhancement without beamforming," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6502–6506.

[9] Y. Luo, Z. Chen, N. Mesgarani, and T. Yoshioka, "End-to-end microphone permutation and number invariant multi-channel speech separation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6394–6398.

[10] A. Pandey, B. Xu, A. Kumar, J. Donley, P. Calamia, and D. Wang, "Time-domain ad-hoc array speech enhancement using a triple-path network," *arXiv preprint arXiv:2110.11844*, 2021.

[11] J. Chen and X.-L. Zhang, "Scaling sparsemax based channel selection for speech recognition with ad-hoc microphone arrays," *arXiv preprint arXiv:2103.15305*, 2021.

[12] L. Feng, Y. Gong, and X.-L. Zhang, "Soft label coding for end-to-end sound source localization with ad-hoc microphone arrays," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[13] P. Tzirakis, A. Kumar, and J. Donley, "Multi-channel speech enhancement using graph neural networks," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3415–3419.

[14] N. Furnon, R. Serizel, S. Essid, and I. Illina, "Dnn-based mask estimation for distributed speech enhancement in spatially unconstrained microphone arrays," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2310–2323, 2021.

[15] N. Furnon, R. Serizel, S. Essid, and Illina, "Attention-based distributed speech enhancement for unconstrained microphone arrays with varying number of nodes," in *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 1095–1099.

[16] L. Zhao, W. Zhu, S. Li, H. Luo, X.-L. Zhang, and S. Rahardja, "Multi-resolution convolutional residual neural networks for monaural speech dereverberation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 2338–2351, 2024.

[17] R. Wang, J. Ao, L. Zhou, S. Liu, Z. Wei, T. Ko, Q. Li, and Y. Zhang, "Multi-view self-attention based transformer for speaker recognition," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6732–6736.

[18] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 21–25.

[19] L. Drude, J. Heymann, C. Boeddeker, and R. Haeb-Umbach, "Nara-wpe: A python package for weighted prediction error dereverberation in numpy and tensorflow for online and offline processing," in *Speech Communication; 13th ITG-Symposium*. VDE, 2018, pp. 1–5.

[20] M. Wolf and C. Nadeu, "Channel selection measures for multi-microphone speech recognition," *Speech Communication*, vol. 57, pp. 170–180, 2014.

[21] X. Anguera, C. Wooters, and J. Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2011–2022, 2007.

[22] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[23] 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.

[24] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011.

[25] I.-T. Recommendation, "Perceptual evaluation of speech quality (pesq): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," *Rec. ITU-T P. 862*, 2001.

[26] Y. Hu and P. C. Loizou, "Evaluation of objective quality measures for speech enhancement," *IEEE Transactions on audio, speech, and language processing*, vol. 16, no. 1, pp. 229–238, 2007.