

TPARN: TRIPLE-PATH ATTENTIVE RECURRENT NETWORK FOR TIME-DOMAIN MULTICHANNEL SPEECH ENHANCEMENT

Ashutosh Pandey^{1*}, Buye Xu¹, Anurag Kumar¹, Jacob Donley¹, Paul Calamia¹ and DeLiang Wang²

¹Facebook Reality Labs Research, USA

²Department of Computer Science and Engineering, The Ohio State University, USA

ABSTRACT

In this work, we propose a new model called triple-path attentive recurrent network (TPARN) for multichannel speech enhancement in the time domain. TPARN extends a single-channel dual-path network to a multichannel network by adding a third path along the spatial dimension. First, TPARN processes speech signals from all channels independently using a dual-path attentive recurrent network (ARN), which is a recurrent neural network (RNN) augmented with self-attention. Next, an ARN is introduced along the spatial dimension for spatial context aggregation. TPARN is designed as a multiple-input and multiple-output architecture to enhance all input channels simultaneously. Experimental results demonstrate the superiority of TPARN over existing state-of-the-art approaches.

Index Terms— multichannel, time-domain, MIMO, self-attention, triple-path, fixed array

1. INTRODUCTION

Speech enhancement is concerned with improving the intelligibility and quality of a speech signal degraded by noise and reverberation. The most basic approach to speech enhancement is monaural processing, where recordings from a single microphone are utilized [1]. Single-channel methods can obtain good enhancement, but are limited in capability to utilize only time-frequency (T-F) information. Multichannel speech enhancement aims at utilizing both T-F and spatial information by using recordings from multiple microphones [2], [3].

Supervised speech enhancement using deep neural networks (DNNs) represents the mainstream methodology for speech enhancement [4]. For multichannel processing, a popular approach is to incorporate DNNs with traditional spatial filters, such as an MVDR beamformer [5], [6]. A DNN is first used to estimate second-order statistics of speech and noise which are then used for computing beamformer weights. Another approach is to train a DNN with spatial features, such as inter-channel time, phase or level difference [7], [8]. A more recent trend is to use end-to-end supervised learning, where spatial information becomes an implicit part of supervised learning [9], [10]. Wang et al. [10] proposed a dense convolutional recurrent network (DCRN) for multi-microphone complex spectral mapping, where real and imaginary components of the clean spectrum are directly predicted from the multichannel noisy spectrum. In [9], authors used inspiration from complex beamforming to propose a novel channel-attention mechanism inside a dense UNet.

Moreover, time-domain speech enhancement using DNNs has also gained considerable attention in recent years [11]–[14]. Time

domain networks directly map noisy speech samples to clean speech samples, and as a result, feature extraction becomes an implicit part of the learning process. Even though highly effective in removing additive interference, time-domain approaches have not yet been established for removing room reverberation, a convolutive interference [15]. Time-domain networks have also been explored for end-to-end multichannel speech enhancement [16]–[18], however reported performances are far from satisfactory.

In this work, we propose a novel approach for end-to-end time-domain multichannel speech enhancement. We refer to it as *TPARN: Triple-path Attentive Recurrent Network*. The key idea in the TPARN design is to extend a dual-path network [14] with a third path along the spatial dimension. The audio input signals from all channels are first divided into short chunks which are then processed by the TPARN system in three stages. These three stages for processing include: *intra-chunk processing* for local temporal modeling, *inter-chunk processing* for global temporal modeling, and *inter-channel processing* for spatial modeling.

Intra-chunk and inter-chunk processing are performed independently for all the channels using a dual-path attentive recurrent networks (ARN) [19], which are RNNs augmented with self-attention [20], [21]. A combination of RNN and self-attention has been proven to be effective for speech processing tasks [19], [21]–[23]. The inter-channel processing introduced by us can be modeled using different methods and we explore RNN, self-attention network, and ARN along the spatial dimension. We find ARN to be slightly superior compared to the other two. Besides, with an explicit capability to capture spatial information through neural network based inter-channel processing, our TPARN framework has additional desirable characteristics. For example, TPARN is designed as a multiple-input and multiple-output (MIMO) architecture to enhance all input channels simultaneously. These multi-channel outputs can be further processed by a downstream system if needed. We train and evaluate TPARN on two different datasets with varying degrees of reverberation and noise. We show that TPARN obtains better results than state-of-the-art approaches on both datasets. We also find performance improvements to be more significant on a difficult dataset.

2. MODEL DESCRIPTION

2.1. Problem Definition

A multichannel noisy signal $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_P\} \in \mathbb{R}^{P \times N}$, where P is the number of microphones and N is the number of samples, is

*Work done during internship at Facebook Reality Labs Research.

modeled as

$$\begin{aligned} x_p(n) &= y_p(n) + z_p(n) \\ &= d_p(n) + r_p(n) + z_p(n) \\ &= d_p(n) + u_p(n) \end{aligned} \quad (1)$$

where $p = 1, 2, \dots, P$ and $n = 0, 1, \dots, N - 1$. x_p represents noisy signal at microphone p . y is the received speech including direct-path speech d and reverberated speech r . z is the noise and u is the overall interference including noise and reverberation. The goal of multichannel speech enhancement is to get a close estimate \hat{d}_r of the direct-path clean speech at a reference microphone r from \mathbf{X} .

2.2. Triple-path Attentive Recurrent Network

The overall schema of the proposed TPARN architecture is shown in Fig. 1. It consists of an input linear layer, four TPARN blocks, and an output linear layer.

An input signal $\mathbf{X} \in \mathbb{R}^{P \times N}$ is first converted into frames, $\mathbf{T} = [\mathbf{X}_1, \dots, \mathbf{X}_T] \in \mathbb{R}^{P \times T \times L}$, using a frame size of L samples and frame shift of K . T is the total number of frames. The frames in \mathbf{T} are arranged into chunks with a chunk size of R and chunk shift of S , leading the input being represented as $\mathbf{C} = [\mathbf{C}_1, \dots, \mathbf{C}_C] \in \mathbb{R}^{P \times C \times R \times L}$, where C is the number of chunks. Next, the frames of size L in \mathbf{C} are projected to D dimensions using the input linear layer, which are then processed by a stack of 4 TPARN blocks. The architecture of a TPARN block is shown in Fig. 2. The TPARN blocks are densely connected. The input to TPARN blocks are 4d tensors of shape $P \times C \times R \times k \cdot D$, which are obtained by concatenating the output from the linear layer encoder and the outputs from preceding TPARN blocks. $k \in \{1, 2, 3, 4\}$ denotes block id.

For $k > 1$, a linear layer is used at the input to project features of size $k \cdot D$ to D . Within a TPARN block, the inputs are processed using a stack of three ARNs: intra-chunk ARN, inter-chunk ARN and inter-channel ARN. The intra-chunk ARN operates independently over all chunks by rearranging its input to shape $P \cdot C \times R \times D$, and using an ARN that treats the first, second and third dimensions as batch, sequence and feature dimensions respectively. Similarly, the inter-chunk ARN combines all chunks together by rearranging its input to shape $P \cdot R \times C \times D$. The inter-channel ARN operates along the channel dimension (spatial dimension) by rearranging its input to shape $R \cdot C \times P \times D$. The sequence length and the batch size for an utterance for different ARNs are given in Table 1.

Fig. 2 also depicts the architecture of an ARN which comprises of a stack of three blocks, RNN block, attention block, and feedforward block. The architecture of the RNN block, the attention block and the feedforward block is shown in Fig. 3. The input to RNN block is normalized using two independent layer normalization layers. We use separate normalization to make sure that the network has a capability to scale a given signal differently at different locations inside the network.

The input to the attention block is layer-normalized using two independent layer-normalization layers. The first layer-normalized input is used as query, \mathbf{Q} , and the second layer-normalized input is

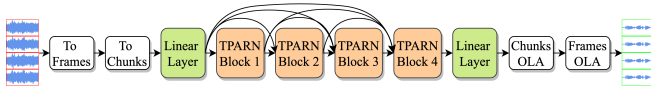


Fig. 1. The proposed TPARN architecture for multichannel speech enhancement.

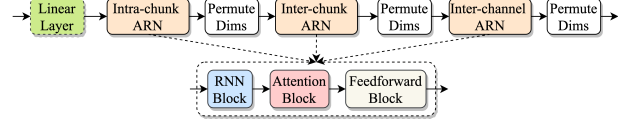


Fig. 2. TPARN block.

Table 1. Input size to different ARNs for an utterance with C chunks.

	Batch Size	Seq. Length	Feature Size
Intra-chunk ARN	$P \cdot C$	R	D
Inter-chunk ARN	$P \cdot R$	C	D
Inter-channel ARN	$R \cdot C$	P	D

used as key \mathbf{K} and value \mathbf{V} for a following attention module. The attention mechanism of the attention module, shown in Fig. 4, is borrowed from [20] where its effectiveness with RNN for natural language processing tasks has been demonstrated. Its effectiveness for speech enhancement has also been established in [19], [21].

The attention module comprises three trainable vectors $\{\mathbf{Q}', \mathbf{K}', \mathbf{V}'\} \in \mathbb{R}^{1 \times D}$, and its inputs are $\{\mathbf{Q}, \mathbf{K}, \mathbf{V}\} \in \mathbb{R}^{B \times U \times D}$, where B is the batch size and U is the sequence length (see Table 1). \mathbf{Q} , \mathbf{K} , and \mathbf{V} are refined using a gating mechanism given in the following equation.

$$\begin{aligned} \mathbf{K}_r &= \mathbf{K} \odot \text{Sigm}(\mathbf{K}') \\ \mathbf{Q}_r &= \text{Lin}(\mathbf{Q}) \odot \text{Sigm}(\mathbf{Q}') \\ \mathbf{V}_r &= \mathbf{V} \odot [\text{Sigm}(\text{Lin}(\mathbf{V}')) \odot \text{Tanh}(\text{Lin}(\mathbf{V}'))] \end{aligned} \quad (2)$$

where $\text{Sigm}()$ is the sigmoidal nonlinearity, $\text{Lin}()$ is a linear layer, and \odot denotes elementwise multiplication. \mathbf{Q}' , \mathbf{K}' , and \mathbf{V}' are broadcast to match the shape of \mathbf{Q} , \mathbf{K} , and \mathbf{V} . Note that $\text{Sigm}(\text{Lin}(\mathbf{V}')) \odot \text{Tanh}(\text{Lin}(\mathbf{V}'))$ is a deterministic vector, and hence this operation is used only during training to better optimize \mathbf{V}' , and its final value is stored as a vector to use during evaluation.

The final output of the attention block is computed as

$$\mathbf{A} = \text{Softmax}\left(\frac{\mathbf{Q}_r \mathbf{K}_r^T}{\sqrt{D}}\right) \mathbf{V}_r \quad (3)$$

The input to the feedforward block in ARN is layer-normalized independently using two different layer normalization layers. The

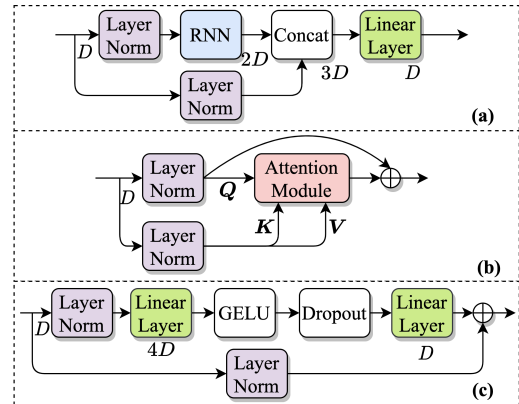


Fig. 3. (a) RNN block, (b) Attention block, (c) Feedforward block.

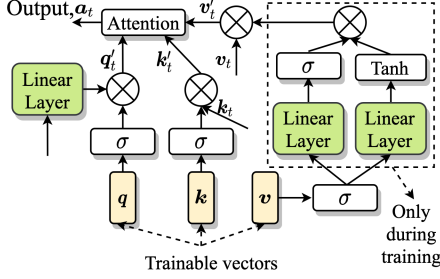


Fig. 4. Self-attention mechanism.

first layer-normalized input is projected to size $4D$ using a linear layer followed by Gaussian error linear unit (GELU) nonlinearity and a dropout with dropout rate D_r . Finally, it is projected back to size D and added to the second layer-normalized input. Note that we use dense connection in the RNN block and residual connections in the attention and the feedforward block.

The output of the final TPARN block is projected to size L using the output linear layer. Next, chunks are combined together using chunk overlap-and-add (OLA) and then frames are combined together using frame OLA to get an enhanced multichannel waveform. Note that TPARN is a MIMO architecture that enhances all input channels simultaneously.

2.3. Loss Function

We use a recently proposed phase constrained magnitude (PCM) loss [24] for training. The PCM loss was proposed to overcome an existing artifact issue with magnitude-based losses for time-domain networks. First, we compute an estimate of overall interference as

$$\hat{U} = X - \hat{D} \quad (4)$$

Then, a magnitude-based loss is used between reference and estimated speech and noise.

$$L_{PCM}(D, \hat{D}) = \frac{1}{2} \cdot L_{SM}(D, \hat{D}) + \frac{1}{2} \cdot L_{SM}(U, \hat{U}) \quad (5)$$

L_{SM} is defined as

$$L_{SM}(D, \hat{D}) = \frac{1}{P \cdot T \cdot F} \sum_{p=0}^P \sum_{t=1}^T \sum_{f=1}^F (|\mathcal{D}_p^r(t, f)| + |\mathcal{D}_p^i(t, f)| - (|\hat{\mathcal{D}}_p^r(t, f)| + |\hat{\mathcal{D}}_p^i(t, f)|)) \quad (6)$$

where \mathcal{D}_p is STFT of d_p and \mathcal{D}_p^r and \mathcal{D}_p^i respectively represent the real and the imaginary component of \mathcal{D}_p . T is the number of frames and F is the number of frequency bins.

3. EXPERIMENTS

3.1. Datasets

We use two datasets for experiments. The first dataset is created using speech from WSJCAM0 corpus [25] and noises from the REVERB challenge [26]. This dataset uses a 4 microphones circular array with a radius of 10 cm, T60 in the range [0.2, 1.2] seconds, and direct speech-to-noise-ratio (SNR) in the range [5, 20] dB. More details about this dataset can be found in [10].

Table 2. Comparison of Loss Functions for TPARN.

	SI-SDR	STOI	PESQ
unproc.	-3.8	70.9	1.63
MSE	10.1	95.7	3.07
PCM	10.4	96.9	3.42

Algorithm 1 DNS dataset spatialization process.

```

for split in {train, test, validation} do
  for speech utterances in split do
    • Draw room length and width from [5,10] m, and height from [3, 4] m;
    • Draw 1 array location and 1 speech source location;
    • Get 4 uniformly placed mic locations on a circle of radius 10 cm centered at array location;
    • Draw  $N_{ns}$ , number of noise sources, from [5, 10]
    • Draw  $N_{ns}$  noise locations in room
    • Generate RIRs corresponding to speech source location and  $N_{ns}$  noise locations for mic locations in circular array
    • Draw  $N_{ns}$  noise utterances from noises in split
    • Propagate speech and noise signals to mics by convolving with corresponding RIRs
    • Draw a value snr from [-10, 10] dB, and add speech and noises at each mic using a scale so that the overall direct speech SNR is snr;
  end for
end for

```

We also create a more challenging dataset using speech and noises from the DNS challenge 2020 corpus¹ [27]. We select all speakers with one chapter in the dataset and randomly select 90% of speakers for training, 5% for validation, and 5% for evaluation. After this, for each utterance a random chunk of a randomly sampled length with an activity threshold (from script in [27]) greater than 0.6 is extracted. The length of utterances are sampled from [3, 6] seconds for training and [3, 10] seconds for test and validation. This results in a total of 53 k utterances for training, 2.6 k for validation, and 3.3 k for test. Next, all the noises from the DNS corpus are randomly divided into training, validation and test noises in a proportion similar to the number of speech utterances.

The algorithm to generate spatialized multichannel data from DNS speech and noises is given in Algorithm 1. All the points inside a room are sampled at least 0.5 m away from walls, and the distance between array center and different sound sources is kept between 0.75 m and 2 m. We use Pyroomacoustics [28] with hybrid approach where the image method with order 6 is used to model early reflections and ray-tracing is used to model the late reverberation. A similar approach to data generation was used in [29].

3.2. Experimental settings

All the utterances are resampled to 16 kHz. We use $P = 4$, $L = 16$, $K = 8$, $R = 126$, $S = 63$, and $D = 128$. For RNN, we use bidirectional long short-term memory networks (BLSTMs) with hidden size D in each direction. Dropout rate D_r in feedforward blocks is set to 5%. We use phase constrained magnitude (PCM) loss in Eq. (6) for training TPARN. All the models are trained for 100 epochs on 4 second long utterances randomly extracted from training

¹<https://github.com/microsoft/DNS-Challenge/blob/master/LICENSE>

samples during training. A batch size of 8 is used. Automatic mixed precision training is utilized for efficient training [30]. Learning rate is initialized with 0.0004 and is dynamically scaled to half if the best validation score does not improve in five consecutive epochs.

We compare TPARN with two recently proposed complex spectrum based end-to-end models; DCRN [10] and channel-attention dense UNet (CA-DUNet)[9]. We also compare it with two time-domain end-to-end models; residual speech denoising fully convolutional network (rSDFCN) [17], and filter-and-sum network with transform average and concatenate module (Fasnet TAC) [18].

All the models are compared using three enhancement objective metrics: short-time objective intelligibility (STOI) [31], perceptual evaluation of speech quality (PESQ) [32], and scale-invariant signal-to-distortion ratio (SI-SDR). STOI scores are reported in percentage.

3.3. Experimental results

We begin by comparing time-domain mean squared error (MSE) loss with spectral magnitude based PCM loss in Eq. (5). Results on WSJ-CAM0 dataset using TPARN are reported in Table 2. We see that even though MSE loss obtains good SI-SDR and STOI scores, it is considerably worse in terms of PESQ. This also suggests that PCM loss obtains bigger improvements for joint denoising and dereverberation compared to denoising in [13]. We use PCM loss for the rest of the experiments with TPARN.

Next, we explore the behavior of spatial (inter-channel) ARNs at different locations inside a TPARN bloc: *Pre* - before intra-chunk and inter-chunk ARN, *Mid* - between intra-chunk and inter-chunk ARN, and *Post* - after inter-chunk and intra-chunk ARN. We also experiment with three different configurations for spatial processing: *Attention* - removing the RNN block from ARN, *RNN* - removing the attention block from ARN, and *ARN*. The results for different models are summarized in Table 3. We see that for the WSJCAM0 dataset, different configurations of spatial ARN in TPARN block obtain similar results. However, for the challenging DNS dataset, *RNN* is considerably worse in comparison of *Attention* and *ARN*. Moreover, best scores are obtained at *Pre* location for *Attention* and at *Post* location for *ARN*. These results indicate that RNNs are good for spatial modeling but may not suffice in extreme conditions, as in the DNS dataset.

Additionally, we perform an ablation experiment (not reported here) on the number of spatial ARNs inside TPARN by removing spatial ARNs from TPARN blocks at different locations. We find that a TPARN with 3 spatial ARNs in the first, second and the fourth TPARN block obtains consistently better results for both datasets and for different learning strategies, such as multiple-input and single-output (MISO) and MIMO.

Finally, we compare best TPARN results with baseline models in Table 4. We report single-input and single-output (SISO), MISO, and MIMO results for DCRN and TPARN, and MISO results for the rest of the models, as in their original studies. TPARN MIMO is converted to MISO by averaging the output of the final TPARN block. We notice a very interesting observation that TPARN-SISO is worse than DCRN-SISO, but TPARN-MIMO and TPARN-MISO are better than DCRN-MISO and DCRN-MIMO. This suggests that TPARN exploits spatial information to a larger extent than in DCRN. Further, DCRN-MIMO is worse than DCRN-MISO, but TPARN-MIMO is slightly better than TPARN-MISO. This indicates that TPARN is capable of MIMO learning without any performance degradation, which provides additional advantage of enhancing all channels simultaneously with spatial cue preservation. Moreover, performance improvements over the baseline models are even better for the dif-

Table 3. Comparisons between different spatial processing modules at different locations in TPARN blocks.

	Test Dataset	WSJCAM0			DNS		
		SI-SDR	STOI	PESQ	SI-SDR	STOI	PESQ
Unproc.	Loc. ↓	-3.8	70.9	1.38	-7.6	63.8	1.38
<i>Attention</i>	<i>Pre</i>	10.3	96.8	3.40	7.6	91.1	2.66
	<i>Mid</i>	10.1	96.8	3.39	7.0	90.2	2.56
	<i>Post</i>	10.2	96.8	3.40	6.9	90.2	2.56
<i>RNN</i>	<i>Pre</i>	10.3	96.8	3.40	6.9	90.0	2.55
	<i>Mid</i>	10.3	96.9	3.44	7.1	90.4	2.59
	<i>Post</i>	10.3	96.9	3.41	7.2	90.3	2.57
<i>ARN</i>	<i>Pre</i>	10.4	96.8	3.40	7.4	90.5	2.59
	<i>Mid</i>	10.4	96.9	3.41	7.1	90.4	2.59
	<i>Post</i>	10.4	96.9	3.42	7.8	91.1	2.65

Table 4. Comparisons with baseline models. WM: waveform mapping (time-domain), CRM: complex ratio masking. CSM: complex spectral mapping.

	Test Dataset	WSJCAM0			DNS		
		SI-SDR	STOI	PESQ	SI-SDR	STOI	PESQ
Approach ↓	Unprocessed	-3.8	70.9	1.38	-7.6	63.8	1.38
WM	rSDFCN-MISO [17]	4.2	83.8	2.00	-2.2	68.5	1.49
CRM	CA-DUNet-MISO [9]	10.7	96.0	2.88	3.5	83.3	1.99
WM	Fasnet TAC-MISO [18]	8.2	94.7	2.93	4.7	86.5	2.26
CSM	DCRN-SISO [10]	6.6	93.6	2.90	3.9	89.8	2.60
CSM	DCRN-MISO [10]	9.4	96.5	3.31	4.6	90.1	2.57
CSM	DCRN-MIMO [10]	8.0	95.9	3.27	3.6	89.4	2.57
WM	TPARN-SISO	5.1	93.6	2.92	3.0	84.1	2.14
WM	TPARN-MISO	10.2	96.8	3.40	8.2	91.6	2.72
WM	TPARN-MIMO	10.4	96.9	3.43	8.4	91.9	2.75

ficult DNS dataset. For example, TPARN is better by 3.8 dB in SI-SDR, 1.9% in STOI and 0.12 in PESQ compared to the second best DCRN-MISO. An exception in baseline models is CA-DUNet that obtains impressive SI-SDR but drastically worse PESQ for WSJ-CAM0.

4. CONCLUSIONS

We have proposed a novel triple-path attentive recurrent network for multichannel speech enhancement in the time-domain. TPARN is designed as a simple extension of a single-channel dual-path network to multichannel network by adding a third-path along the spatial dimension. TPARN is a multiple-input and multiple-output (MIMO) architecture that can simultaneously enhance signals at all microphones. We have shown that TPARN obtains significantly better results than other state-of-the-art models in very noisy and reverberant conditions. Future research includes exploring TPARN for ad-hoc array processing and moving sources.

REFERENCES

- [1] P. C. Loizou, *Speech Enhancement: Theory and Practice*. CRC Press, 2013.
- [2] J. Benesty, J. Chen, and Y. Huang, *Microphone array signal processing*. Springer Science & Business Media, 2008.
- [3] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, "A consolidated perspective on multimicrophone speech enhancement and source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 692–730, 2017.
- [4] D. L. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 1702–1726, 2018.

- [5] H. Erdogan *et al.*, “Improved MVDR beamforming using single-channel mask prediction networks,” in *INTER-SPEECH*, 2016, pp. 1981–1985.
- [6] J. Heymann, L. Drude, and R. Haeb-Umbach, “Neural network based spectral mask estimation for acoustic beamforming,” in *IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, 2016, pp. 196–200.
- [7] Z.-Q. Wang and D. Wang, “Combining spectral and spatial features for deep learning based blind speaker separation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 457–468, 2018.
- [8] Z.-Q. Wang, J. Le Roux, and J. R. Hershey, “Multi-channel deep clustering: Discriminative spectral and spatial embeddings for speaker-independent speech separation,” in *IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, 2018, pp. 1–5.
- [9] B. Tolooshams *et al.*, “Channel-attention dense U-Net for multichannel speech enhancement,” in *IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, 2020, pp. 836–840.
- [10] Z.-Q. Wang and D. Wang, “Multi-microphone complex spectral mapping for speech dereverberation,” in *IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, 2020, pp. 486–490.
- [11] A. Pandey and D. L. Wang, “A new framework for supervised speech enhancement in the time domain,” in *INTERSPEECH*, 2018, pp. 1136–1140. DOI: 10.21437/Interspeech.2018-1223.
- [12] Y. Luo and N. Mesgarani, “Conv-TasNet: Surpassing ideal time-frequency magnitude masking for speech separation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 1256–1266, 2019.
- [13] A. Pandey and D. Wang, “Dense cnn with self-attention for time-domain speech enhancement,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 1270–1279, 2021.
- [14] Y. Luo, Z. Chen, and T. Yoshioka, “Dual-path RNN: Efficient long sequence modeling for time-domain single-channel speech separation,” in *IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, 2020, pp. 46–50.
- [15] Y. Luo and N. Mesgarani, “Real-time single-channel dereverberation and separation with time-domain audio separation network,” in *INTERSPEECH*, 2018, pp. 342–346.
- [16] N. Tawara, T. Kobayashi, and T. Ogawa, “Multi-channel speech enhancement using time-domain convolutional denoising autoencoder,” in *INTERSPEECH*, 2019, pp. 86–90.
- [17] C.-L. Liu *et al.*, “Multichannel speech enhancement by raw waveform-mapping using fully convolutional networks,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 1888–1900, 2020.
- [18] Y. Luo, Z. Chen, N. Mesgarani, and T. Yoshioka, “End-to-end microphone permutation and number invariant multi-channel speech separation,” in *IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, 2020, pp. 6394–6398.
- [19] A. Pandey and D. Wang, “Dual-path self-attention RNN for real-time speech enhancement,” *arXiv preprint arXiv:2010.12713*, 2020.
- [20] S. Merity, “Single headed attention RNN: Stop thinking with your head,” *arXiv:1911.11423*, 2019.
- [21] A. Pandey and D. Wang, “Self-attending RNN for speech enhancement to improve cross-corpus generalization,” *arXiv:2105.12831*, 2021.
- [22] K. Tan *et al.*, “SAGRNN: Self-attentive gated RNN for binaural speaker separation with interaural cue preservation,” *arXiv:2009.01381*, 2020.
- [23] J. Chen, Q. Mao, and D. Liu, “Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation,” *arXiv preprint arXiv:2007.13975*, 2020.
- [24] A. Pandey and D. L. Wang, “Densely connected neural network with dilated convolutions for real-time speech enhancement in the time domain,” in *IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, 2020, pp. 6629–6633.
- [25] T. Robinson *et al.*, “WSJCAMO: a British English speech corpus for large vocabulary continuous speech recognition,” in *1995 International Conference on Acoustics, Speech, and Signal Processing*, IEEE, vol. 1, 1995, pp. 81–84.
- [26] K. Kinoshita *et al.*, “A summary of the REVERB challenge: State-of-the-art and remaining challenges in reverberant speech processing research,” *EURASIP Journal on Advances in Signal Processing*, pp. 1–19, 2016.
- [27] C. K. Reddy *et al.*, “The INTERSPEECH 2020 deep noise suppression challenge: Datasets, subjective testing framework, and challenge results,” *arXiv:2005.13981*, 2020.
- [28] R. Scheibler, E. Bezzam, and I. Dokmanić, “Pyroomacoustics: A python package for audio room simulation and array processing algorithms,” in *IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, IEEE, 2018, pp. 351–355.
- [29] O. Kabeli *et al.*, “Online self-attentive gated rnns for real-time speaker separation,” *arXiv:2106.13493*, 2021.
- [30] P. Micikevicius *et al.*, “Mixed precision training,” *arXiv preprint arXiv:1710.03740*, 2017.
- [31] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, “A short-time objective intelligibility measure for time-frequency weighted noisy speech,” in *IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, 2010, pp. 4214–4217.
- [32] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, “Perceptual evaluation of speech quality (PESQ) - a new method for speech quality assessment of telephone networks and codecs,” in *IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, 2001, pp. 749–752.