DUAL-PATH SELF-ATTENTION RNN FOR REAL-TIME SPEECH ENHANCEMENT

Ashutosh Pandey[1] and DeLiang Wang[1,2]

[1]Department of Computer Science and Engineering, The Ohio State University, USA
[2]Center for Cognitive and Brain Sciences, The Ohio State University, USA
{pandey.99, wang.77}@osu.edu

ABSTRACT

We propose a dual-path self-attention recurrent neural network (DP-SARNN) for time-domain speech enhancement. We improve dual-path RNN (DP-RNN) by augmenting inter-chunk and intra-chunk RNN with a recently proposed efficient attention mechanism. The combination of inter-chunk and intra-chunk attention improves the attention mechanism for long sequences of speech frames. DP-SARNN outperforms a baseline DP-RNN by using a frame shift four times larger than in DP-RNN, which leads to a substantially reduced computation time per utterance. As a result, we develop a real-time DP-SARNN by using long short-term memory (LSTM) RNN and causal attention in inter-chunk SARNN. DP-SARNN significantly outperforms existing approaches to speech enhancement, and on average takes 7.9 ms CPU time to process a signal chunk of 32 ms.

Index Terms— time-domain, real-time, dual-path RNN, self-attention, speaker- and noise-independent

1. INTRODUCTION

Speech enhancement aims at improving the intelligibility and quality of a speech signal corrupted by additive noise. It is used as a preprocessor in many applications, such as automatic speech recognition, telecommunication, and hearing aids, to improve their performance in noisy environments.

Traditional speech enhancement approaches include spectral subtraction, Wiener filtering and statistical model-based methods [1]. In recent years, speech enhancement has been extensively studied as a deep learning problem [2]. In particular, time-domain speech enhancement is becoming increasingly popular due to its capability to jointly enhance both the magnitude and the phase of noisy speech. Further, time domain approaches do not require transformations to and from the frequency domain.

Representative time-domain networks include UNet [3] convolutional neural networks (CNNs) [4, 5], CNNs with temporal and dilated convolutions [6, 7], dense CNN [8], and CNN with self-attention [9].

Dual-path recurrent neural network (DP-RNN) was recently proposed for time-domain speaker separation with state-of-the art performance [10]. In DP-RNN, a sequence of frames is divided into overlapping chunks and processed by a series of intra-chunk and inter-chunk RNNs. The efficacy of DP-RNN can be attributed to a reduced sequence length per RNN for efficient training, and a smaller frame size (0.25 ms) for speech processing.

Motivated by the success of DP-RNN for speaker separation [10] and the attention mechanism for speech enhancement [9], in this work, we propose to augment DP-RNN with attention. We replace inter-chunk and intra-chunk RNN in DP-RNN with a recently proposed self-attention RNN (SARNN) [11]. SARNN was proposed as an efficient technique to augment RNNs with attention, resulting in reduced memory consumption, faster training, and state-of-the-art performance in natural language processing. A similar idea of augmenting DP-RNN with attention using a different approach was recently proposed in [12] for binaural speaker separation.

Dual-path SARNN (DP-SARNN) outperforms DP-RNN by using a frame shift four times larger than in DP-RNN, resulting in considerably reduced computation time per utterance. We use low-latency DP-SARNN to develop a real-time algorithm by using long short-term memory (LSTM) RNN and causal attention in inter-chunk SARNN.

DP-SARNN significantly outperforms existing methods for speech enhancement for both offline and real-time speech enhancement. Real-time DP-SARNN, on average, takes 7.9 ms CPU time to process a signal chunk of 32 ms, which is significantly less than than other comparable time-domain models, DP-RNN (17.4 ms) and dense convolutional network (DCN) (11.0 ms) [9].

The rest of the paper is organized as follows. Section 2 describes DP-SARNN. Experimental settings and results are given in Section 3. Section 4 concludes the paper.

2. MODEL DESCRIPTION

2.1. Self-Attention RNN

SARNN was recently proposed in [11] for efficiently combining attention mechanism [13] with recurrent processing of

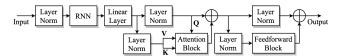RNN. A block diagram of SARNN used in this study is shown in Fig. 1.

Fig. 1: The proposed architecture of SARNN.

SARNN comprises an RNN, an attention block, and a feedforward block . The input to SARNN is a matrix $\boldsymbol{X} \in \mathbb{R}^{T \times N}$, where T is the sequence length and N is the feature dimension. $\boldsymbol{X}$ is layer normalized [14] and fed to an RNN of hidden size H, which is followed by a linear layer to transform the RNN output to the same shape as the input. Next, two different layer normalizations are used to get query ($\boldsymbol{Q}$), key ($\boldsymbol{K}$) and value ($\boldsymbol{V}$) for the following attention block, where $\boldsymbol{K}$ is equal to $\boldsymbol{V}$. $\boldsymbol{Q}$ is added to the output of the attention block to form a residual connection. The output after the attention block is processed using the feedforward block in a residual way as shown in Fig. 1.

A block diagram of the attention block in SARNN is shown in Fig. 2. It comprises three trainable vectors $\{\boldsymbol{Q}', \boldsymbol{K}', \boldsymbol{V}'\} \in \mathbb{R}^{1 \times N}$, and its inputs are $\{\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}\} \in \mathbb{R}^{T \times N}$. $\boldsymbol{Q}, \boldsymbol{K}$, and $\boldsymbol{V}$ are refined using a gating mechanism given in the following equation.

$$\begin{aligned} \boldsymbol{K}_r &= \boldsymbol{K} \odot \text{Sigm}(\boldsymbol{K}') \\ \boldsymbol{Q}_r &= \text{Lin}(\boldsymbol{Q}) \odot \text{Sigm}(\boldsymbol{Q}') \\ \boldsymbol{V}_r &= \boldsymbol{V} \odot [\text{Sigm}(\text{Lin}(\boldsymbol{V}')) \odot \text{Tanh}(\text{Lin}(\boldsymbol{V}'))] \end{aligned} \quad (1)$$

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

The final output of the attention block is computed as

$$\boldsymbol{A} = \text{Softmax}\left(\frac{\boldsymbol{Q}_r \boldsymbol{K}_r^T}{\sqrt{N}}\right) \boldsymbol{V}_r \quad (2)$$

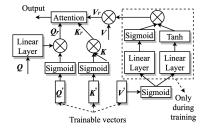The feedforward block in SARNN, shown in Fig. 3, is similar

Fig. 2: Attention block in SARNN.

Fig. 3: Feedforward block in SARNN.

to the ones used in transformer networks [13]. It is a fully connected network with one hidden layer of size 4N with Gaussian error linear unit (GELU) nonlinearity [15] and dropout.

2.2. Dual-Path SARNN

Let $\boldsymbol{X} \in \mathbb{R}^{T \times N}$ be a matrix representing a time series signal $\{\boldsymbol{x}_1 \cdots \boldsymbol{x}_T\}$, where $\boldsymbol{x}_i \in \mathbb{R}^{N \times 1}$. We can divide $\boldsymbol{X}$ into overlapping chunks of length K with a chunk shift of P and concatenate them together to get a 3D tensor $\mathbf{X} \in \mathbb{R}^{J \times K \times N}$, where J is the number of chunks. If required, $\boldsymbol{X}$ is zero padded to the right to get the last chunk of size K. The kth signal in the jth chunk of $\mathbf{X}$ is defined as

$$\mathbf{X}_{j,k} = \boldsymbol{x}_{(j-1)*P+k}, \ 1 \le j \le J, 1 \le k \le K \quad (3)$$

DP-SARNN is modeled by replacing RNNs in a DP-RNN [10] with SARNNs. An illustrative diagram of DP-SARNN is shown in Fig. 4. A DP-SARNN comprises two SARNNs: intra-chunk SARNN and inter-chunk SARNN. It takes $\mathbf{X}$ as input and processes it using intra-chunk SARNN and inter-chunk SARNN in that order. Intra-chunk SARNN considers frames in a chunk as the sequential input, and separately processes all the chunks $[\mathbf{X}_1 \cdots \mathbf{X}_J]$ and concatenates them together to get $\mathbf{X}^1 \in \mathbb{R}^{J \times K \times N}$. Next, $\mathbf{X}^1$ is transposed along the first and second dimension to get $\mathbf{X}^2 \in \mathbb{R}^{K \times J \times N}$. $\mathbf{X}^2$ is fed to inter-chunk SARNN, which considers chunks as the sequential input, and separately processes $[\mathbf{X}^2_1 \cdots \mathbf{X}^2_K]$ and concatenates them together to get $\mathbf{X}^3 \in \mathbb{R}^{K \times J \times N}$. Finally, $\mathbf{X}^3$ is transposed to get the final output $\mathbf{X}^4 \in \mathbb{R}^{J \times K \times N}$.
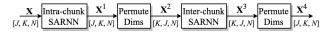

Fig. 4: An illustrative diagram of DP-SARNN.

2.3. Speech enhancement using DP-SARNN

Given a speech signal $\boldsymbol{s}$, a noise signal $\boldsymbol{n}$, the noisy speech signal is modeled as

$$\boldsymbol{x} = \boldsymbol{s} + \boldsymbol{n} \quad (4)$$

where $\{\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{n}\} \in \mathbb{R}^{M \times 1}$, and M represents the number of samples in the signal. The goal of a speech enhancement algorithm is to get a close estimate, $\hat{\boldsymbol{s}}$, of $\boldsymbol{s}$ given $\boldsymbol{x}$.

$\boldsymbol{x}$ is first converted to frames using a frame size of L and frame shift of R to get $\boldsymbol{X} \in \mathbb{R}^{T \times L}$, where T is the number of frames. Next, $\boldsymbol{X}$ is divided into chunks with a chunk size of K and chunk shift of P to get $\mathbf{X} \in \mathbb{R}^{J \times K \times L}$, where J is the number of chunks.
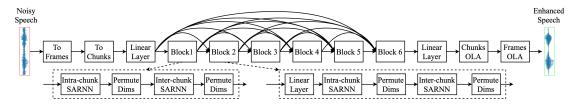
Fig. 5: The proposed speech enhancement network.

X is fed to a DP-SARNN based speech enhancement network shown in Fig. 4. The proposed network consists of a linear layer at the input, six DP-SARNN blocks, and one linear layer at the output. The input linear layer is used to project **X** to a higher dimension of size N. It is then processed by six DP-SARNN blocks. The input to a given DP-SARNN block is a concatenation of the outputs of all the previous blocks and the input layer. If the input size to a block is greater than N, it is projected to size N using a linear layer. Finally, the linear layer at the output transforms the output dimension from N to L (frame size) to get $\widehat{\mathbf{S}} \in \mathbb{R}^{J \times K \times L}$. $\widehat{\mathbf{S}}$ is subjected to overlap-and-add (OLA) along the chunks to get $\widehat{\boldsymbol{S}} \in \mathbb{R}^{T \times L}$. $\widehat{\boldsymbol{S}}$ is subjected to OLA along the frames to get enhanced signal $\hat{\boldsymbol{s}} \in \mathbb{R}^{M \times 1}$.

2.3.1. Real-time speech enhancement using DP-SARNN

A real-time speech enhancement system must satisfy a causality constraint and a latency constraint. The causality constraint requires that the output for a given frame is computed using only the current and the previous frames. For DP-SARNN, we modify the frame-level constraint to chunk-level as given in Eq. 5.

$$\widehat{\mathbf{S}}_j = f_\theta([\mathbf{X}_1, \mathbf{X}_2 \cdots \mathbf{X}_{j-1}, \mathbf{X}_j]) \tag{5}$$

Where f_θ is a DP-SARNN model parametrized by θ.

The latency constraint requires the frame size to be small and the computation time for a frame to be less than the frame shift [16]. Similar to causality constraint, we apply latency constraint to chunks for DP-SARNN.

For non-causal speech enhancement, we use bidirectional long short-term memory (BLSTM) RNN in inter-chunk and intra-chunk SARNN. For causal speech enhancement, the BLSTM in inter-chunk SARNN is replaced with LSTM, and the attention defined in Eq. (2) is replaced with a causal attention defined in Eq. (7).

$$\boldsymbol{W} = \text{Softmax}(\frac{\boldsymbol{Q}_r \boldsymbol{K}_r^T}{\sqrt{N}})$$
$$\boldsymbol{A}_{causal} = \text{Mask}(\boldsymbol{W})\boldsymbol{V}_r \tag{6}$$

where $\boldsymbol{W} \in \mathbb{R}^{T \times T}$, and

$$\text{Mask}(W)(i,j) = \begin{cases} W(i,j), & \text{if } i \leq j \\ -\infty, & \text{otherwise} \end{cases} \tag{7}$$

We use a chunk size of 32 ms and chunk shift of 16 ms, and verify that the computation time for a chunk is less that 16 ms on CPU.

3. EXPERIMENTS

3.1. Datasets

We train speaker- and noise-independent models using a large number of noises and speakers. We use the WSJ0 SI-84 dataset [17] consisting of 83 speakers (42 males and 41 females) in which 76 are used for training and the remaining 6 (3 males and 3 females) are used for evaluation.

320000 training mixtures are generated by adding random noise segments from a sound effect library of 10000 non-speech sounds (www.sound-ideas.com) at random SNRs from {-5 dB, -4 dB, -3 dB, -2 dB, -1 dB, 0 dB}.

Test mixtures are generated by using two noises (babble and cafeteria) from an Auditec CD (http://www.auditec.com) at SNRs of -5 dB, 0 dB and 5 dB. We generate 150 mixtures for all pairs of test noises and test SNRs. For the validation set, we use 6 speakers from the training set (150 utterances) and mix it with factory noise at an SNR of -5 dB.

3.2. Baselines

We compare DP-SARNN with a recently proposed gated convolutional recurrent network (GCRN) [18], auto-encoder CNN (AECNN) [4], speech enhancement generative adversarial network (SEGAN) [5], temporal convolutional neural network (TCNN) [6], and dual-path RNN (DP-RNN) [10]. DP-RNN is adopted for speech enhancement by using one decoder instead of two and replacing masking with mapping as in DP-SARNN. Non-causal DP-RNN and DP-SARNN use BLSTM RNN, whereas causal DP-RNN and DP-SARNN replace BLSTM with LSTM in inter-chunk RNN. In our results (Table 1 and Table 2), causal DP-RNN and DP-SARNN are respectively denoted as DP-LSTM and DP-SALSTM, and non-causal DP-RNN and DP-SARNN are respectively denoted as DP-BLSTM and DP-SABLSTM.

3.3. Experimental settings

All the utterances are resampled to 16 kHz. We use $L = 16$, $R = 8$, $K = 126$ for DP-SABLSTM, $K = 63$ for DP-

Table 1: STOI and PESQ comparisons between DP-SARNN and baseline models of a) complex spectral mapping, and b) time-domain enhancement.

Approach	Causal?	Real-time?	Metric	STOI (%)								PESQ							
			Test Noise	Babble				Cafeteria				Babble				Cafeteria			
			Test SNR	-5 db	0 dB	5 dB	AVG	-5 dB	0 dB	5 dB	AVG	-5 db	0 dB	5 dB	AVG	-5 dB	0 dB	5 dB	AVG
			Mixture	58.4	70.5	81.3	70.1	57.1	69.7	81.0	69.2	1.56	1.82	2.12	1.83	1.46	1.77	2.12	1.78
a)	✓	✓	GCRN [18]	82.4	90.9	94.8	89.4	79.1	89.3	94.0	87.5	2.17	2.70	3.07	2.65	2.10	2.60	2.99	2.56
	✗	✗	NC-GCRN [18]	87.0	93.0	95.6	91.9	84.1	91.7	95.1	90.3	2.53	2.96	3.25	2.91	2.40	2.85	3.17	2.81
b)	✓	✗	SEGAN-T [5]	81.5	90.3	94.1	88.6	79.8	89.5	93.5	87.6	2.11	2.62	2.97	2.57	2.15	2.61	2.94	2.57
	✓	✗	AECNN [4]	82.6	91.5	95.1	89.7	81.1	90.7	94.5	88.8	2.21	2.80	3.17	2.73	2.23	2.76	3.12	2.70
	✓	✓	TCNN [6]	82.8	91.3	94.8	89.6	80.6	89.8	94.0	88.1	2.18	2.70	3.06	2.65	2.14	2.62	2.98	2.58
	✓	✓	DCN [9]	85.1	92.7	95.8	91.2	82.5	91.3	95.1	89.6	2.31	2.91	3.30	2.84	2.29	2.82	3.22	2.78
	✓	✗	DP-LSTM	87.5	93.7	96.2	92.5	83.8	92.1	95.5	90.4	2.50	3.04	3.36	2.97	2.35	2.91	3.29	2.85
	✓	✓	DP-SALSTM	**88.5**	**94.3**	**96.6**	**93.1**	**84.7**	**92.7**	**95.8**	**91.1**	**2.54**	**3.10**	**3.42**	**3.02**	**2.41**	**2.94**	**3.32**	**2.89**
	✗	✗	NC-DCN	89.0	94.3	96.6	93.3	85.6	93.0	95.9	91.5	2.71	3.18	3.48	3.12	2.56	3.07	3.39	3.01
	✗	✗	DP-BLSTM	90.08	94.7	96.7	93.8	86.8	93.5	96.1	92.1	2.82	3.23	3.48	3.18	2.65	3.13	3.43	3.07
	✗	✗	DP-SABLSTM	**91.61**	**95.3**	**97.1**	**94.7**	**87.9**	**94.0**	**96.4**	**92.8**	**2.94**	**3.30**	**3.54**	**3.26**	**2.70**	**3.17**	**3.46**	**3.11**

SALSTM, $N = 128$, and $H = 256$. For DP-SABLSTM and DP-BLSTM, K is chosen in a way so that $K \approx J$. Dropout rate in feedforward block is set to 5%. We use phase constrained magnitude (PCM) loss for training [9] . All the models are trained for 15 epochs on 4 second long utterances with a batch size of 8. Mixed precision training [19] is utilized for faster training. Learning rate is set to 0.0002 for first 5 epochs and exponentially decayed afterwards at every epoch using a rate that results in a learning rate of 0.00002 in the last epoch. Gradient clipping is applied during training with a maximum l^2 norm of 3.

We observe short-time objective intelligibility (STOI) [20] score on the validation set after each epoch of training, and the model with maximum STOI is used for evaluation.

3.4. Experimental results

We compare all the models in terms of STOI whose values typically range between 0 and 1 and perceptual evaluation of speech quality (PESQ) whose values range from -0.5 to 4.5 [21].

First, we compare DP-SARNN with different baselines in Table 1. There are four real-time models in Table 1: GCRN, TCNN, DCN, and DP-SALSTM. The performance trend for these models is GCRN < TCNN < DCN < DP-SALSTM. In particular, on average, DP-SALSTM improves scores over DCN by 1.7% in STOI and 0.15 in PESQ, where best improvements are obtained for the difficult SNR of -5 dB.

A similar behavior is observed for non real-time mod-

Table 2: Model size and processing time comparisons between DCN, DP-LSTM, and DP-SALSTM.

	DCN	DP-LSTM	DP-SALSTM
(K, P)	-	(255, 127)	(63, 31)
(L, R)	(512, 256)	(4, 2)	(16, 8)
Chunk size (ms)	32	32	32
# params (millions)	5.6	3.37	6.49
Avg time per chunk (ms)	11.0	17.4	7.9
Is real-time?	✓	✗	✓

els, in which case the performance trend is SEGAN-T < AECNN < NC-GCRN < NC-DCN < DP-LSTM < DP-BLSTM <DP-SABLSTM. Note that SEGAN, AECNN and DP-LSTM are causal systems but not real-time because AECNN and SEGAN respectively use frame size of 128 ms and 1024 ms, and DP-LSTM does not satisfy the latency constraint (Table 2).

Next, we compare top three causal systems, DP-SALSTM, DP-LSTM, and DCN in terms of number of parameters and average computation time for a signal chunk of 32 ms on a 2.4 GHz quad core machine with Intel Xeon E5-2680 v4 processors and 4GB RAM. The results are given in Table 2. The computation time is in order DP-SALSTM < DCN < DP-LSTM. Even though DP-LSTM is causal, and has fewer parameters, it takes 17 ms CPU time to process a signal chunk of 32 ms, which is greater than the chunk shift (16 ms), and hence it is a non-real-time system. DP-SALSTM is faster than DP-LSTM because it uses a frame shift of 8 instead of 2, and as a result, it needs to process four times less number of frames. DP-LSTM can be converted to a real-time model by increasing the frame size and frame shift, but it will lead to performance degradation as in [10]. Number of parameters in different models are in order DP-LSTM < DCN < DP-SALSTM.

We plan to explore DP-RNN and DP-SARNN for cross-corpus generalization by training on LibriSpeech [22] as in [23, 24].

4. CONCLUSIONS

We have proposed a novel dual-path self-attention RNN for time-domain speech enhancement. The proposed DP-SARNN augments RNNs in DP-RNN with attention. Adding attention in DP-RNN leads to improved enhancement with a four times frame shift, resulting in a low-latency model. As a result, we have developed a real-time version of DP-SARNN that is not only also faster than but also outperforms existing approaches. Future work includes exploring and improving DP-SARNN for cross-corpus generalization.

5. REFERENCES

[1] P. C. Loizou, *Speech Enhancement: Theory and Practice*, CRC Press, Boca Raton, FL, USA, 2nd edition, 2013.

[2] D. L. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, pp. 1702–1726, 2018.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-assisted Intervention*, 2015, pp. 234–241.

[4] A. Pandey and D. L. Wang, "A new framework for CNN-based speech enhancement in the time domain," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 27, pp. 1179–1188, 2019.

[5] S. Pascual, A. Bonafonte, and J. Serrà, "SEGAN: Speech enhancement generative adversarial network," in *INTERSPEECH*, 2017, pp. 3642–3646.

[6] A. Pandey and D. L. Wang, "TCNN: Temporal convolutional neural network for real-time speech enhancement in the time domain," in *ICASSP*, 2019, pp. 6875–6879.

[7] D. Rethage, J. Pons, and X. Serra, "A wavenet for speech denoising," in *ICASSP*, 2018, pp. 5069–5073.

[8] A. Pandey and D. L. Wang, "Densely connected neural network with dilated convolutions for real-time speech enhancement in the time domain," in *ICASSP*, 2020, pp. 6629–6633.

[9] A. Pandey and D. Wang, "Dense CNN with self-attention for time-domain speech enhancement," *arXiv:2009.01941*, 2020.

[10] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path RNN: Efficient long sequence modeling for time-domain single-channel speech separation," in *ICASSP*, 2020, pp. 46–50.

[11] S. Merity, "Single headed attention RNN: Stop thinking with your head," *arXiv:1911.11423*, 2019.

[12] K. Tan, B. Xu, A. Kumar, E. Nachmani, and Y. Adi, "SAGRNN: Self-attentive gated RNN for binaural speaker separation with interaural cue preservation," *arXiv:2009.01381*, 2020.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.

[14] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv:1607.06450*, 2016.

[15] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," *arXiv:1606.08415*, 2016.

[16] C. K. Reddy, H. Dubey, V. Gopal, R. Cutler, S. Braun, H. Gamper, R. Aichner, and S. Srinivasan, "ICASSP 2021 deep noise suppression challenge," *arXiv:2009.06122*, 2020.

[17] D. B. Paul and J. M. Baker, "The design for the wall street journal-based CSR corpus," in *Workshop on Speech and Natural Language*, 1992, pp. 357–362.

[18] K. Tan and D. L. Wang, "Learning complex spectral mapping with gated convolutional recurrent networks for monaural speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 380–390, 2019.

[19] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. Wu, "Mixed precision training," in *ICLR*, 2018.

[20] 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, pp. 2125–2136, 2011.

[21] 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 *ICASSP*, 2001, pp. 749–752.

[22] 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.

[23] A. Pandey and D. L. Wang, "On cross-corpus generalization of deep learning based speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, in press, 2020.

[24] A. Pandey and D. L. Wang, "Learning complex spectral mapping for speech enhancement with improved cross-corpus generalization," in *INTERSPEECH*, 2020, p. in press.