SELF-ATTENTION GENERATIVE ADVERSARIAL NETWORK FOR SPEECH ENHANCEMENT

Huy Phan[*1], *Huy Le Nguyen*[2], *Oliver Y. Chén*[3], *Philipp Koch*[4],
Ngoc Q. K. Duong[5], *Ian McLoughlin*[6], *Alfred Mertins*[4]

[1]Queen Mary University of London, UK, [2]HCMC University of Technology, Vietnam
[3]University of Oxford, UK, [4]University of Lübeck, Germany
[5]InterDigital R&D France, France, [6]Singapore Institute of Technology, Singapore
[*]Correspondence email: h.phan@qmul.ac.uk

ABSTRACT

Existing generative adversarial networks (GANs) for speech enhancement solely rely on the convolution operation, which may obscure temporal dependencies across the sequence input. To remedy this issue, we propose a self-attention layer adapted from non-local attention, coupled with the convolutional and deconvolutional layers of a speech enhancement GAN (SEGAN) using raw signal input. Further, we empirically study the effect of placing the self-attention layer at the (de)convolutional layers with varying layer indices as well as at all of them when memory allows. Our experiments show that introducing self-attention to SEGAN leads to consistent improvement across the objective evaluation metrics of enhancement performance. Furthermore, applying at different (de)convolutional layers does not significantly alter performance, suggesting that it can be conveniently applied at the highest-level (de)convolutional layer with the smallest memory overhead[1].

***Index Terms*—** Speech enhancement, self-attention, generative adversarial network, GAN, SEGAN

1. INTRODUCTION

Speech enhancement is useful in many applications, such as speech recognition [1, 2, 3] and hearing aids [4, 5]. Recently, the research community has witnessed a shift in methodology from conventional signal processing methods [6, 7] to data-driven enhancement approaches, particularly those based on deep learning paradigms [8, 9, 3, 10, 11]. Beside discriminative modeling with typical deep network variants, such as deep neural networks (DNNs) [8], convolutional neural networks (CNNs) [9, 10], and recurrent neural networks (RNNs) [11, 3], generative modeling with GANs [12] have been shown to hold promise for speech enhancement [13, 14, 15]. Furthermore, the study in [15] indicates that generative modeling with GANs may result in fewer artefacts than discriminative methods.

Since the seminal work [13], SEGAN has been improved in various ways. Different input types have been exploited, e.g. raw waveform [15, 16] and time-frequency image [9, 14]. Better losses, like Wasserstein loss [14], relativistic loss [16], and metric loss [14], have been tailored to gain stabilization in the training process. SEGANs that learn multi-stage enhancement mappings have also been proposed [15]. However, convolutional layers are still, and will probably remain, the backbone of these SEGAN variants. This reliance on the convolution operator limits SEGAN's capability in capturing long-range dependencies across an input sequence due to the convolution operator's local receptive field. Temporal dependency modeling is, in general, an integral part of a speech modeling system [17, 18] but has mostly remained uncharted in SEGAN systems.

In this work, we aim to address the lack of sequential modeling capacity in SEGAN by integrating it with self-attention. On the one hand, self-attention has been successfully used for sequential modeling in different speech modeling tasks [18, 19, 20]. On the other hand, it is more flexible in modeling both long-range and local dependencies and is more efficient than RNN [14] in terms of computational cost, especially when applied to long sequences. The reason is that RNN is based on temporal iterations which cannot be parallelized whereas self-attention is based on matrix multiplication which is highly parallelizable and easily accelerated. We, therefore, propose a self-attention layer following the principle of non-local attention [21, 22] and couple it with the (de)convolutional layers of a SEGAN to construct a self-attention SEGAN (SASEGAN for short). We further conduct analysis of how the proposed self-attention layer applied at different (de)convolutional layers will affect the enhancement performance. We will show, when equipped with a sequential modeling capability, the proposed SASEGAN leads to better enhancement performance than the SEGAN baseline across all the objective evaluation metrics. Furthermore, the performance gain is consistent regardless of which (de)convolutional layer the self-attention is applied,

[1] Source code is available at http://github.com/pquochuy/sasegan

allowing it to be integrated into a SEGAN with a very small additional memory footprint.

2. SELF-ATTENTION SEGAN

2.1. SEGAN

Given a noise-corrupted raw audio signal $\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{n} \in \mathbb{R}^T$, where $\mathbf{x} \in \mathbb{R}^T$ denotes a clean signal and $\mathbf{n} \in \mathbb{R}^T$ denotes additive background noise, the goal of speech enhancement is to find a mapping $f(\tilde{\mathbf{x}}) : \tilde{\mathbf{x}} \mapsto \mathbf{x}$ to recover the clean signal $\mathbf{x}$ from the noisy signal $\tilde{\mathbf{x}}$. SEGAN methods [13, 14, 15] achieve this goal by designating the generator G as the enhancement mapping, i.e. $\hat{\mathbf{x}} = G(\mathbf{z}, \tilde{\mathbf{x}})$ where $\mathbf{z}$ is a latent variable. The discriminator D is tasked to distinguish the enhanced output $\hat{\mathbf{x}}$ from the real clean signal $\mathbf{x}$. To this end, D learns to classify the pair $(\mathbf{x}, \tilde{\mathbf{x}})$ as real and $(\hat{\mathbf{x}}, \tilde{\mathbf{x}})$ as fake. At the same time, G learns to produce as good an enhanced signal $\tilde{\mathbf{x}}$ as possible to fool D such that D classifies $(\hat{\mathbf{x}}, \tilde{\mathbf{x}})$ as real. SEGAN is trained in this adversarial manner, as illustrated in Fig. 1. Various losses have been proposed to improve adversarial training, such as least-squares loss [13, 23], Wasserstein loss [14], relativistic loss [16], and metric loss, [14]. Here, we employ the least-squares loss as in the seminal work [13]. The least-squares objective functions of D and G are explicitly written as

$$\begin{aligned}\min_D \mathcal{L}_{\text{LS}}(D) =& \frac{1}{2}\mathbb{E}_{\mathbf{x},\tilde{\mathbf{x}}\sim p_{\text{data}}(\mathbf{x},\tilde{\mathbf{x}})}(D(\mathbf{x},\tilde{\mathbf{x}})-1)^2 \\ &+ \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z}),\tilde{\mathbf{x}}\sim p_{\text{data}}(\tilde{\mathbf{x}})} D(G(\mathbf{z},\tilde{\mathbf{x}}),\tilde{\mathbf{x}})^2, \quad (1)\\ \min_G \mathcal{L}_{\text{LS}}(G) =& \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z}),\tilde{\mathbf{x}}\sim p_{\text{data}}(\tilde{\mathbf{x}})}(D(G(\mathbf{z},\tilde{\mathbf{x}}),\tilde{\mathbf{x}})-1)^2 \\ &+ \lambda||G(\mathbf{z},\tilde{\mathbf{x}}) - \mathbf{x}||_1. \quad (2)\end{aligned}$$

2.2. Self-attention SEGAN (SASEGAN)

2.2.1. *Self-attention layer*

The proposed self-attention layer is adapted from the non-local attention [21, 22]. Given the feature map $\mathbf{F} \in \mathbb{R}^{L \times C}$ output by a convolutional layer, where L is the time dimension, C is the number of channels. Note that the feature dimension is one since we are using 1D convolution to deal with raw speech input in this case. The query matrix $\mathbf{Q}$, the key matrix $\mathbf{K}$, and the value matrix $\mathbf{V}$ are obtained via transformations:

$$\mathbf{Q} = \mathbf{F}\mathbf{W}_Q, \ \mathbf{K} = \mathbf{F}\mathbf{W}_K, \ \mathbf{V} = \mathbf{F}\mathbf{W}_V, \quad (3)$$

where $\mathbf{W}_Q \in \mathbb{R}^{C \times \frac{C}{k}}$, $\mathbf{W}_K \in \mathbb{R}^{C \times \frac{C}{k}}$, and $\mathbf{W}_V \in \mathbb{R}^{C \times \frac{C}{k}}$ denote the weight matrices which are implemented by a 1×1 convolution layer of $\frac{C}{k}$ filters. That is, in the new feature spaces, the channel dimension is reduced by the factor k mainly for memory reduction. Furthermore, given the $O(n^2)$ memory complexity, we also reduce the number of keys and values (i.e. the time dimension of $\mathbf{K}$ and $\mathbf{V}$) by a factor of p for memory efficiency. This is accomplished by a max pooling layer with filter width and stride size of p. We use $k = 8$ and $p = 4$ here. The size of the matrices are, therefore, $\mathbf{Q} \in \mathbb{R}^{L \times \frac{C}{k}}$, $\mathbf{K} \in \mathbb{R}^{\frac{L}{p} \times \frac{C}{k}}$, and $\mathbf{V} \in \mathbb{R}^{\frac{L}{p} \times \frac{C}{k}}$. The attention map $\mathbf{A}$ and the attentive output $\mathbf{O}$ are then computed as

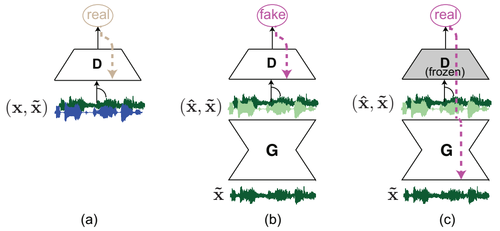
real
fake
real
D
D
D
(frozen)
$(\mathbf{x}, \tilde{\mathbf{x}})$
$(\hat{\mathbf{x}}, \tilde{\mathbf{x}})$
$(\hat{\mathbf{x}}, \tilde{\mathbf{x}})$
G
G
$\tilde{\mathbf{x}}$
$\tilde{\mathbf{x}}$
(a)
(b)
(c)

Fig. 1. Adversarial training of GAN-based speech enhancement methods: D learns to classify the pair $(\mathbf{x}, \tilde{\mathbf{x}})$ as real (a), and $(\hat{\mathbf{x}}, \tilde{\mathbf{x}})$ as fake (b). G learns to fool D so that D classifies $(\hat{\mathbf{x}}, \tilde{\mathbf{x}})$ as real (c). Dashed lines represent the flow of gradient backpropagation.

$$\mathbf{A} = \text{softmax}(\mathbf{Q}\bar{\mathbf{K}}^{\mathsf{T}}), \ \ \mathbf{A} \in \mathbb{R}^{L \times \frac{L}{p}}, \quad (4)$$

$$\mathbf{O} = (\mathbf{A}\mathbf{V})\mathbf{W}_O, \ \ \mathbf{W}_O \in \mathbb{R}^{\frac{C}{k} \times C}. \quad (5)$$

Each element $a_{ij} \in \mathbf{A}$ indicates the extent to which the model attends to the j^{th} column $\mathbf{v}_j$ of $\mathbf{V}$ when producing the i^{th} output $\mathbf{o}_i$ of $\mathbf{O}$. In addition, a transformation with weight $\mathbf{W}_O$ realized by a 1×1 convolution layer of C filters is applied to $\mathbf{AV}$ to restore the shape of $\mathbf{O}$ to the original shape $L \times C$.

Finally, we make use of a shortcut connection to facilitate information propagation, with the final output given as:

$$\tilde{\mathbf{F}} = \beta\mathbf{O} + \mathbf{F}, \quad (6)$$

where β is a learnable parameter. We illustrate the processing steps of a simplified self-attention layer with $L = 6$, $C = 4$, $p = 3$, and $k = 2$ in Fig. 2.

2.2.2. *Network architecture*

Similar to SEGAN, the generator receives a raw-signal input of length $L = 16,384$ samples (approximately one second at 16 kHz) and features an encoder-decoder architecture with fully-convolutional layers [24], as illustrated in Fig. 3 (a). The encoder consists of 11 one-dimensional strided convolutional layers with a common filter width of 31, a stride of 2, and increasing number of filters $\{16, 32, 32, 64, 64, 128, 128, 256, 256, 512, 1024\}$, resulting in feature maps of size 8192×16, 4096×32, 2048×32, 1024×64, 512×64, 256×128, 128×128, 64×256, 32×256, 16×512, 8×1024, respectively. The noise sample $\mathbf{z} \in \mathbb{R}^{8 \times 1024}$ is then stacked on the last feature map and presented to the decoder. The decoder, on the other hand, mirrors the encoder architecture to reverse the encoding process by means of deconvolutions. All the (de)convoltuional layers are followed by parametric rectified linear units (PReLUs) [25]. In order to allow information from the encoding stage to flow into the decoding stage, a skip connection is used to connect each convolutional layer in the encoder to its mirrored deconvolutional layer in the decoder (cf. Fig. 3 (a)).

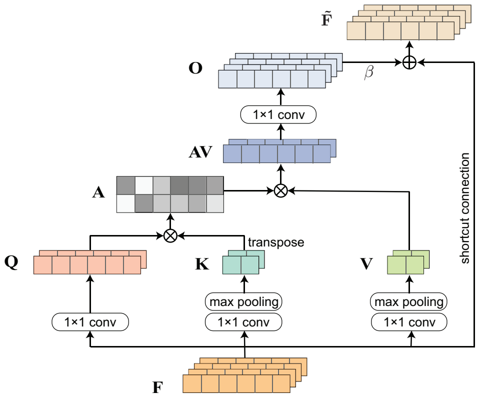

Fig. 2. Illustration of the processing steps of the proposed self-attention layer with $L = 6$, $C = 4$, $p = 3$, and $k = 2$.

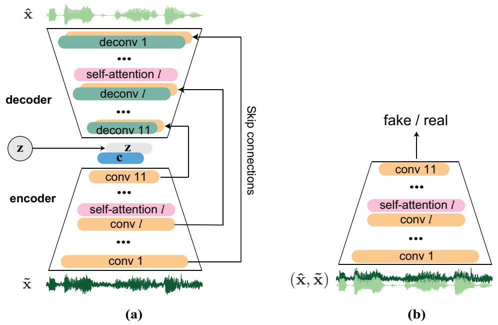

Fig. 3. Illustration of SASEGAN. (a) the generator, (b) the discriminator.

The discriminator architecture, as illustrated in Fig. 3 (b), is similar to the encoder part of the generator. However, it receives a pair of raw audio segments as input. Its convolutional layers are also associated with virtual batch-norm [26] and Leaky ReLU activation [27] with $\alpha = 0.3$. The last convolutional feature map of size 8×1024 is further processed by an additional 1×1 convolutional layer and reduced to 8 features which are used for classification with softmax.

SASEGAN couples the self-attention layer described in Section 2.2.1 with the (de)convolutional layers of both the generator and the discriminator. Fig. 3 (a) and (b) show an example where the self-attention layer is coupled with the l^{th} (de)convolutional layer. In general, the self-attention layer can be used in combination with any number, even all, of the (de)convolutional layers. We will investigate the self-attention layer placement in Section 3. In SASEGAN, spectral normalization [28] is applied to all the (de)convolutional layers of the generator and the discriminator.

3. EXPERIMENTS

3.1. Experimental setup

We set two objectives in the conducted experiments. First, we studied and quantified the effects of using the proposed self-attention layer in speech enhancement. Second, we aimed to analyze the influence of self-attention layer placement in the generator and the discriminator, on enhancement performance. For the former, we used the SEGAN (i.e. without self-attention) [13] as the baseline for comparison. For the latter, we evaluated SASEGAN with different values of the (de)convolutional layer index $l \in \{3, 4, \ldots, 11\}$. Note that, we were unable to experiment with the early (de)convolutional layers (i.e. 1^{st} and 2^{nd}) due to GPU memory limitations given the large time dimension of their feature maps, 8192 and 4096, respectively. We also studied the case when the self-attention layer was combined with all of the l^{th} (de)convolutional layers where $l \in \{3, 4, \ldots, 11\}$.

3.2. Dataset

The experiments were based on the database introduced in [29]. This is also the one used in [13] to evaluate the SEGAN baseline. It consists of data from 30 speakers extracted from the Voice Bank corpus [30]. Ten types of noise were combined at signal-to-noise ratios (SNRs) of 15, 10, 5, and 0 dB, to introduce 40 noisy conditions to the training data. Similarly, 20 noisy conditions were introduced to the test data by combining five types of noise from the Demand database [31] with four SNRs (17.5, 12.5, 7.5, and 2.5 dB). This resulted in 10 and 20 utterances for each noise condition per speaker in the training and test data, respectively. Adhering to prior works [13, 15, 14], data from 28 speakers was used for training and data from two remaining speakers was used for testing. All audio signals were downsampled to 16 kHz.

3.3. Parameters

The implementation was based on the Tensorflow framework [32]. Networks were trained with RMSprop [33] for 100 epochs with a minibatch size of 50. During training, raw speech segments (of length 16,384 samples each) in a batch were sampled from the training utterances with 50% overlap, followed by a high-frequency preemphasis filter with the coefficient of 0.95. The trained network was then applied to the test utterances for enhancement purpose. For each utterance, raw speech segments were extracted without overlap, processed by the trained network, deemphasized, and concatenated to result in the enhanced utterance.

3.4. Experimental results

We used five objective signal-quality metrics: PESQ (in range $[-0.5, 4.5]$), CSIG (in range $[1, 5]$), CBAK (in range $[1, 5]$), COVL (in range $[1, 5]$), and SSNR (in range $[1, \infty]$); and the speech intelligibility measure STOI (%) [34] for evaluation. As in [15], the five latest network snapshots were used and the results were averaged over 824 utterances of the test data.

The results obtained by the proposed SASEGAN alongside the SEGAN baseline and the noisy speech signals (without enhancement) are shown in Table 1. Note that in the table, we denote the SASEGAN with self-attention at the l^{th} (de)convolutional layer as SASEGAN-l, $3 \leq l \leq 11$, and the one with self-attention at all the l^{th} (de)convolutional layers, where $3 \leq l \leq 11$, as SASEGAN-All. Overall, introducing self-attention to the SASEGANs led to consistent improvements over the SEGAN baseline across all the objective metrics. Averaging over the SASEGAN-ls, absolute

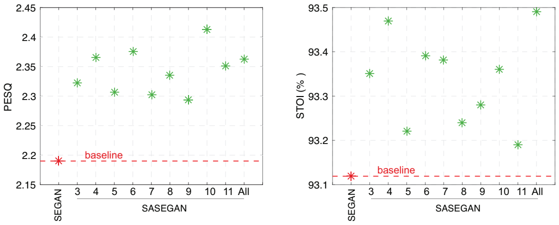

Fig. 4. PESQ and STOI gains obtained by the SASEGAN-ls, $3 \leq l \leq 11$ over the SEGAN baseline.

gains of 0.15, 0.13, 0.14, 0.15, 0.69, and 0.2 were obtained on PESQ, CSIG, CBAK, COVL, SSNR, and STOI over the SEGAN baseline, respectively. The performance was further boosted, although modestly, from the average when multiple self-attention layers were employed in SASEGAN-All, with absolute gains of 0.17, 0.15, 0.18, 0.17, 0.91, and 0.37, respectively. These gains, however, were achieved at the cost of increased computation time and memory requirements.

Furthermore, using self-attention at different layer indices l did not show a clear difference between the performance improvements of SASEGAN-ls over the SEGAN baseline, as depicted in Fig. 4 for PESQ and STOI. This suggests that self-attention applied to the high-level (de)convolutional layer is expected to be as good as when applied in a low-level (de)convolutional layer. More importantly, by doing so, extra memory requirements are exponentially reduced and can be as little as $8^2 = 64$ memory units at $l = 11$ given that the time dimension of the feature map at the l^{th} (de)convolutional layer is given by $\frac{L}{2^l}$.

3.5. Discussion

In order visualize the learned attention weights, taking the case SASEGAN-3 for example, we exhibit in Fig. 5 those attention weights of the generator's encoder corresponding to different temporal locations of the feature map. This suggests that the network leverages complementary features in distant portions of the input rather than local regions of fixed shape to generate the attentive output. Apparently, more attention was put on speech regions (the red distribution in Fig. 5 (c)) to synthesize the new feature at the location with speech (at the red dashed line in Fig. 5 (b)). The opposite is observed for the green distribution in Fig. 5 (c) for the location in background noise region (the green dashed line in Fig. 5 (b)).

Table 1. Results obtained by the studied speech enhancement systems on the objective evaluation metrics. We highlight in bold where the proposed SASEGAN outperforms the baseline SEGAN.

	PESQ	CSIG	CBAK	COVL	SSNR	STOI
Noisy	1.97	3.35	2.44	2.63	1.68	92.10
SEGAN [13]	2.19	3.39	2.90	2.76	7.36	93.12
ISEGAN [15]	2.24	3.23	2.95	2.69	8.17	93.29
DSEGAN [15]	2.35	3.55	3.1	2.93	8.7	93.25
SASEGAN-3	**2.32**	**3.51**	**3.07**	**2.90**	**8.53**	**93.35**
SASEGAN-4	**2.36**	**3.57**	**3.08**	**2.95**	**8.38**	**93.47**
SASEGAN-5	**2.31**	**3.46**	**2.94**	**2.85**	7.20	**93.22**
SASEGAN-6	**2.38**	**3.46**	**3.12**	**2.90**	**8.86**	**93.39**
SASEGAN-7	**2.30**	**3.52**	**2.98**	**2.89**	7.34	**93.38**
SASEGAN-8	**2.34**	**3.55**	**3.03**	**2.92**	**8.03**	**93.24**
SASEGAN-9	**2.29**	**3.45**	**3.05**	**2.85**	**8.48**	**93.28**
SASEGAN-10	**2.41**	**3.62**	**3.06**	**2.99**	**7.87**	**93.36**
SASEGAN-11	**2.35**	**3.57**	**3.03**	**2.94**	**7.76**	**93.19**
Average	**2.34**	**3.52**	**3.04**	**2.91**	**8.05**	**93.32**
SASEGAN-All	**2.36**	**3.54**	**3.08**	**2.93**	**8.27**	**93.49**

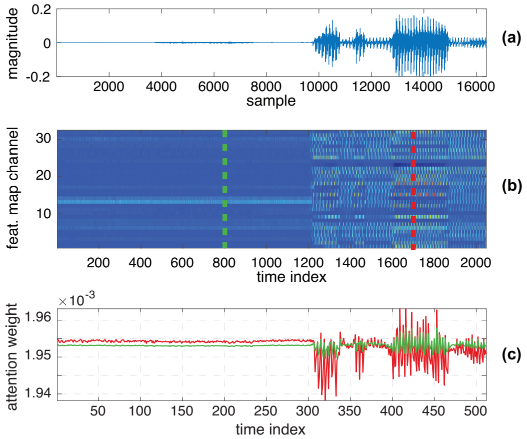

Fig. 5. Visualization of self-attention weights at the 3^{rd} convolutional layer of the generator's encoder. (a) the raw speech input; (b) the feature map of size 2048×32; and (c) the attention weights distributed over 512 time indices of the matrix $\mathbf{V}$ (note that the factor $p = 4$). The green and red distributions in (c) correspond to the two locations specified by the green and red dash lines in (b).

It is also worth mentioning other improved SEGANs [15, 14, 16] that have been built upon the first SEGAN in [13]. While the proposed SASEGAN performs competitively to the existing SEGAN variants, for example ISEGAN and DSEGAN [15] (cf. Table 1), our primary goal in this work is to study the influence of the proposed self-attention layer and its placement on speech enhancement rather than a comprehensive comparison among SEGAN variants. More importantly, the proposed self-attention layer is generic enough that it can be applied to those existing SEGAN variants to further improve their performance. We leave this for a future study.

4. CONCLUSIONS

We proposed and integrated a self-attention layer with SEGAN to improve its temporal dependency modeling for speech enhancement. The proposed self-attention layer can be used at different (de)convolutional layers of the SEGAN's generator and discriminator or even all of them, given sufficient processing memory. Our experiments show that the self-attention SEGAN outperforms the SEGAN baseline over all of the objective evaluation metrics. In addition, consistency in the improvement was seen across the self-attention placement settings. Furthermore, these settings did not result in a significant difference among their performance gains. The results suggest that self-attention can be sufficiently used in a high-level (de)convolutional layer with very small induced memory. Furthermore, it can be easily applied to existing SEGAN variants for potential improvement.

5. REFERENCES

[1] C. Donahue, B. Li, and R. Prabhavalkar, "Exploring speech enhancement with generative adversarial networks for robust speech recognition," in *Proc. ICASSP,*, 2018, pp. 5024–5028.

[2] Y. Xu, C. Weng, L. Hui, J. Liu, M. Yu, D. Su, and D. Yu, "Joint training of complex ratio mask based beamformer and acoustic model for noise robust ASR," in *Proc. ICASSP*, 2019, pp. 6745–6749.

[3] F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. Le Roux, J. R. Hershey, and B. Schuller, "Speech enhancement with LSTM recurrent neural networks and its application to noise-robust asr," *Proc. Intl. Conf. on Latent Variable Analysis and Signal Separation*, pp. 91–99, 2015.

[4] A. Schasse, T. Gerkmann, R. Martin, W. Sörgel, T. Pilgrim, and H. Puder, "Two-stage filter-bank system for improved single-channel noise reduction in hearing aids," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 2, pp. 383–393, 2014.

[5] L.-P. Yang and Q.-J. Fu, "Spectral subtraction-based speech enhancementfor cochlear implant patients in background noise," *Journal of the Acoustical Society of America*, vol. 117, no. 3, pp. 1001–1004, 2005.

[6] T. Gerkmann and R. C. Hendriks, "Unbiased MMSE-based noise power estimation with low complexity and low tracking delay," *IEEE Trans. on Audio, Speech, and Language Processing*, pp. 1383–1393, 2011.

[7] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 33, no. 2, pp. 443–445, 1985.

[8] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Trans. on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 1, pp. 7–19, 2015.

[9] Z. X. Li, L. R. Dai, Y. Song, and I. McLoughlin, "A conditional generative model for speech enhancement," *Circuits, Systems, and Signal Processing*, vol. 37, no. 11, pp. 5005–5022, 2018.

[10] S. R. Park and J. Lee, "A fully convolutional neural network for speech enhancement," in *Proc. Interspeech*, 2017.

[11] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *Proc. ICASSP*, 2015, pp. 708–712.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 2672–2680.

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

[14] Z. Zhang, C. Deng, Y. Shen, D. S. Williamson, Y. Sha, Y. Zhang, H. Song, and X. Li, "On loss functions and recurrency training for gan-based speech enhancement systems," in *Proc. Interspeech*, 2020.

[15] H. Phan, I. V. McLoughlin, L. Pham, O. Y. Chén, P. Koch, M. De Vos, and A. Mertins, "Improving GANs for speech enhancement," *IEEE Signal Processing Letters*, vol. 27, pp. 1700–1704, 2020.

[16] D. Baby and S. Verhulst, "Sergan: Speech enhancement using relativistic generative adversarial networks with gradient penalty," in *Proc. ICASSP*, 2019, pp. 106–110.

[17] A. Defossez, G. Synnaeve, and Y. Adi, "Real time speech enhancement in the waveform domain," in *Proc. Interspeech*, 2020.

[18] N.-Q. Pham, T.-S. Nguyen, J. Niehues, M. Müller, S. Stüker, and A. Waibel, "Very deep self-attention networks for end-to-end speech recognition," in *Proc. Interspeech*, 2019.

[19] M. Sperber, J. Niehues, G. Neubig, S. Stüker, and A. Waibel, "Self-attentional acoustic models," in *Proc. Interspeech*, 2018, pp. 3723–3727.

[20] Z. Tian, J. Yi, J. Tao, Y. Bai, and Z. Wen, "Self-attention transducers for end-to-end speech recognition," in *Proc. Interspeech*, 2019, pp. 4395–4399.

[21] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. CVPR*, 2018.

[22] H. Zhang, I. J. Goodfellow, D. N. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proc. ICML*, 2019, pp. 7354–7363.

[23] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. ICCV*, 2017, pp. 2813–2821.

[24] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. ICLR*, 2016.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. ICCV*, 2015, pp. 1026–1034.

[26] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. NIPS*, 2016, pp. 2226–2234.

[27] A. L. Maas, A. Y. Awni, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, 2013, vol. 30.

[28] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proc. ICLR*, 2018.

[29] C. Valentini-Botinhao, X. Wang, S. Takaki, and J. Yamagishi, "Investigating RNN-based speech enhancement methods for noise-robust text-to-speech," in *Proc. 9th ISCA Speech Synthesis Workshop*, 2016, pp. 146–152.

[30] C. Veaux, J. Yamagishi, and S. King, "The voice bank corpus: design, collection and data analysis of a large regional accent speech database," in *Proc. 2013 International Conference Oriental COCOSDA*, 2013, pp. 1–4.

[31] J. Thiemann, N. Ito, and E. Vincent, "The diverse environments multichannel acoustic noise database: A database of multichannel environmental noise recordings," *The Journal of the AcousticalSociety of America*, vol. 133, no. 5, pp. 3591–3591, 2013.

[32] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[33] T. Tieleman and G. Hinton, "Lecture 6.5 - RMSprop: divide the gradient by a running average of its recent magnitude," *Coursera: Neural Networks for Machine Learning*, 2012.

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