SEGAN: Speech Enhancement Generative Adversarial Network

Santiago Pascual[1], Antonio Bonafonte[1], Joan Serrà[2]

[1]Universitat Politècnica de Catalunya, Barcelona, Spain
[2]Telefónica Research, Barcelona, Spain

`santi.pascual@upc.edu, antonio.bonafonte@upc.edu joan.serra@telefonica.com`

Abstract

Current speech enhancement techniques operate on the spectral domain and/or exploit some higher-level feature. The majority of them tackle a limited number of noise conditions and rely on first-order statistics. To circumvent these issues, deep networks are being increasingly used, thanks to their ability to learn complex functions from large example sets. In this work, we propose the use of generative adversarial networks for speech enhancement. In contrast to current techniques, we operate at the waveform level, training the model end-to-end, and incorporate 28 speakers and 40 different noise conditions into the same model, such that model parameters are shared across them. We evaluate the proposed model using an independent, unseen test set with two speakers and 20 alternative noise conditions. The enhanced samples confirm the viability of the proposed model, and both objective and subjective evaluations confirm the effectiveness of it. With that, we open the exploration of generative architectures for speech enhancement, which may progressively incorporate further speech-centric design choices to improve their performance.

Index Terms: speech enhancement, deep learning, generative adversarial networks, convolutional neural networks.

1. Introduction

Speech enhancement tries to improve the intelligibility and quality of speech contaminated by additive noise [1]. Its main applications are related to improving the quality of mobile communications in noisy environments. However, we also find important applications related to hearing aids and cochlear implants, where enhancing the signal before amplification can significantly reduce discomfort and increase intelligibility [2]. Speech enhancement has also been successfully applied as a preprocessing stage in speech recognition and speaker identification systems [3, 4, 5].

Classic speech enhancement methods are spectral subtraction [6], Wiener filtering [7], statistical model-based methods [8], and subspace algorithms [9, 10]. Neural networks have been also applied to speech enhancement since the 80s [11, 12]. Recently, the denoising auto-encoder architecture [13] has been widely adopted. However, recurrent neural networks (RNNs) are also used. For instance, the recurrent denoising auto-encoder has shown significant performance exploiting the temporal context information in embedded signals. Most recent approaches apply long short-term memory networks to the denoising task [4, 14]. In [15] and [16], noise features are estimated and included in the input features of deep neural networks. The use of dropout, post-filtering, and perceptually motivated metrics are shown to be effective.

Most of the current systems are based on the short-time Fourier analysis/synthesis framework [1]. They only modify the spectrum magnitude, as it is often claimed that short-time phase is not important for speech enhancement [17]. However, further studies [18] show that significant improvements of speech quality are possible, especially when a clean phase spectrum is known. In 1988, Tamura et al. [11] proposed a deep network that worked directly on the raw audio waveform, but they used feed-forward layers that worked frame-by-frame (60 samples) on a speaker-dependent and isolated-word database.

A recent breakthrough in the deep learning generative modeling field are generative adversarial networks (GANs) [19]. GANs have achieved a good level of success in the computer vision field to generate realistic images and generalize well to pixel-wise, complex (high-dimensional) distributions [20, 21, 22]. As far as we are concerned, GANs have not yet been applied to any speech generation nor enhancement task, so this is the first approach to use the adversarial framework to generate speech signals.

The main advantages of the proposed speech enhancement GAN (SEGAN) are:

- It provides a quick enhancement process. No causality is required and, hence, there is no recursive operation like in RNNs.
- It works end-to-end, with the raw audio. Therefore, no hand-crafted features are extracted and, with that, no explicit assumptions about the raw data are done.
- It learns from different speakers and noise types, and incorporates them together into the same shared parametrization. This makes the system simple and generalizable in those dimensions.

In the following, we give an overview of GANs (Sec. 2). Next, we describe the proposed model (Sec. 3) and its experimental setup (Sec. 4). We finally report the results (Sec. 5) and discuss some conclusions (Sec. 6).

2. Generative Adversarial Networks

GANs [19] are generative models that learn to map samples $\mathbf{z}$ from some prior distribution $\mathcal{Z}$ to samples $\mathbf{x}$ from another distribution $\mathcal{X}$, which is the one of the training examples (e.g., images, audio, etc.). The component within the GAN structure that performs the mapping is called the generator (G), and its main task is to learn an effective mapping that can imitate the real data distribution to generate novel samples related to those of the training set. Importantly, G does so not by memorizing input-output pairs, but by mapping the data distribution characteristics to the manifold defined in our prior $\mathcal{Z}$.

The way in which G learns to do the mapping is by means of an adversarial training, where we have another component, called the discriminator (D). D is typically a binary classifier, and its inputs are either real samples, coming from the dataset that G is imitating, or fake samples, made up by G. The adversarial characteristic comes from the fact that D has to classify the samples coming from $\mathcal{X}$ as real, whereas the samples
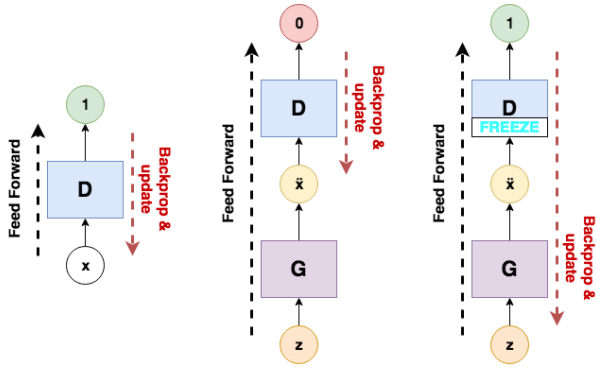
Figure 1: *GAN training process. First, D back-props a batch of real examples. Then, D back-props a batch of fake examples that come from G, and classifies them as fake. Finally, D's parameters are frozen and G back-props to make D misclassify.*

coming from G, $\hat{\mathcal{X}}$, have to be classified as fake. This leads to G trying to fool D, and the way to do so is that G adapts its parameters such that D classifies G's output as real. During back-propagation, D gets better at finding realistic features in its input and, in turn, G corrects its parameters to move towards the real data manifold described by the training data (Fig. 1). This adversarial learning process is formulated as a minimax game between G and D, with the objective

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} \left[\log D(\mathbf{x})\right] + \\ + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} \left[\log\left(1 - D\left(G\left(\mathbf{z}\right)\right)\right)\right]. \quad (1)$$

We can also work with a conditioned version of GANs, where we have some extra information in G and D to perform mapping and classification (see [20] and references therein). In that case, we may add some extra input $\mathbf{x}_c$, with which we change the objective function to

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x},\mathbf{x}_c \sim p_{\text{data}}(\mathbf{x},\mathbf{x}_c)} \left[\log D(\mathbf{x},\mathbf{x}_c)\right] + \\ + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z}),\mathbf{x}_c \sim p_{\text{data}}(\mathbf{x}_c)} \left[\log\left(1 - D\left(G\left(\mathbf{z},\mathbf{x}_c\right),\mathbf{x}_c\right)\right)\right]. \quad (2)$$

There have been recent improvements in the GAN methodology to stabilize training and increase the quality of the generated samples in G. For instance, the classic approach suffered from vanishing gradients due to the sigmoid cross-entropy loss used for training. To solve this, the least-squares GAN (LSGAN) approach [21] substitutes the cross-entropy loss by the least-squares function with binary coding (1 for real, 0 for fake). With this, the formulation in Eq. 2 changes to

$$\min_D V_{\text{LSGAN}}(D) = \frac{1}{2} \mathbb{E}_{\mathbf{x},\mathbf{x}_c \sim p_{\text{data}}(\mathbf{x},\mathbf{x}_c)}[(D(\mathbf{x},\mathbf{x}_c) - 1)^2] + \\ + \frac{1}{2} \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z}),\mathbf{x}_c \sim p_{\text{data}}(\mathbf{x}_c)}[D(G(\mathbf{z},\mathbf{x}_c),\mathbf{x}_c)^2] \quad (3)$$

$$\min_G V_{\text{LSGAN}}(G) = \frac{1}{2} \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z}),\mathbf{x}_c \sim p_{\text{data}}(\mathbf{x}_c)}[(D(G(\mathbf{z},\mathbf{x}_c),\mathbf{x}_c) - 1)^2]. \quad (4)$$

3. Speech Enhancement GAN

The enhancement problem is defined so that we have an input noisy signal $\tilde{\mathbf{x}}$ and we want to clean it to obtain the enhanced signal $\hat{\mathbf{x}}$. We propose to do so with a speech enhancement GAN
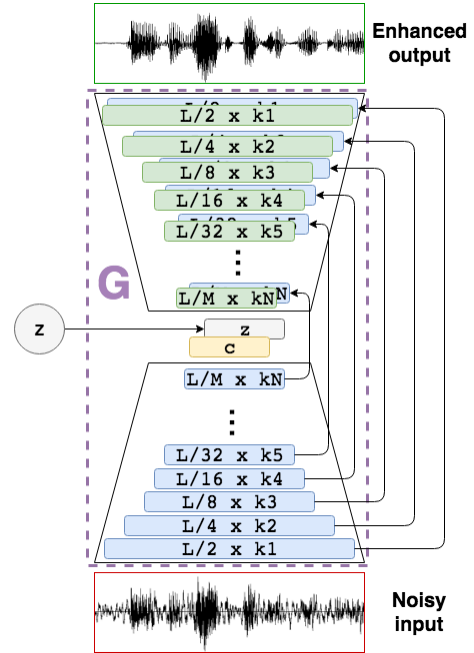

Figure 2: *Encoder-decoder architecture for speech enhancement (G network). The arrows between encoder and decoder blocks denote skip connections.*

(SEGAN). In our case, the G network performs the enhancement. Its inputs are the noisy speech signal $\tilde{\mathbf{x}}$ together with the latent representation $\mathbf{z}$, and its output is the enhanced version $\hat{\mathbf{x}} = G(\tilde{\mathbf{x}})$. We design G to be fully convolutional, so that there are no dense layers at all. This enforces the network to focus on temporally-close correlations in the input signal and throughout the whole layering process. Furthermore, it reduces the number of training parameters and hence training time.

The G network is structured similarly to an auto-encoder (Fig. 2). In the encoding stage, the input signal is projected and compressed through a number of strided convolutional layers followed by parametric rectified linear units (PReLUs) [23], getting a convolution result out of every N steps of the filter. We choose strided convolutions as they were shown to be more stable for GAN training than other pooling approaches [22]. Decimation is done until we get a condensed representation, called the thought vector $\mathbf{c}$, which gets concatenated with the latent vector $\mathbf{z}$. The encoding process is reversed in the decoding stage by means of fractional-strided transposed convolutions (sometimes called deconvolutions), followed again by PReLUs.

The G network also features skip connections, connecting each encoding layer to its homologous decoding layer, and bypassing the compression performed in the middle of the model (Fig. 2). This is done because the input and output of the model share the same underlying structure, which is that of natural speech. Therefore, many low level details could be lost to reconstruct the speech waveform properly if we force all information to flow through the compression bottleneck. Skip connections directly pass the fine-grained information of the waveform to the decoding stage (e.g., phase, alignment). In addition, they offer a better training behavior, as the gradients can flow deeper through the whole structure [24].

An important feature of G is its end-to-end structure, so that it processes raw speech sampled at 16 kHz, getting rid of any

intermediate transformations to extract acoustic features (contrasting to many common pipelines). In this type of model, we have to be careful with typical regression losses like mean absolute error or mean squared error, as noted in the raw speech generative model WaveNet [25]. These losses work under strong assumptions on how our output distribution is shaped and, therefore, impose important modeling limitations (like not allowing multi-modal distributions and biasing the predictions towards an average of all the possible predictions). Our solution to overcome these limitations is to use the generative adversarial setting. This way, D is in charge of transmitting information to G of what is real and what is fake, such that G can slightly correct its output waveform towards the realistic distribution, getting rid of the noisy signals as those are signaled to be fake. In this sense, D can be understood as learning some sort of loss for G's output to look real.

In preliminary experiments, we found it convenient to add a secondary component to the loss of G in order to minimize the distance between its generations and the clean examples. To measure such distance, we chose the L_1 norm, as it has been proven to be effective in the image manipulation domain [20, 26]. This way, we let the adversarial component to add more fine-grained and realistic results. The magnitude of the L_1 norm is controlled by a new hyper-parameter λ. Therefore, the G loss, which we choose to be the one of LSGAN (Eq. 4), becomes

$$\min_G V_{\text{LSGAN}}(G) = \frac{1}{2} \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z}), \tilde{\mathbf{x}} \sim p_{\text{data}}(\tilde{\mathbf{x}})}[(D(G(\mathbf{z}, \tilde{\mathbf{x}}), \tilde{\mathbf{x}}) - 1)^2] +$$
$$+ \lambda \|G(\mathbf{z}, \tilde{\mathbf{x}}) - \mathbf{x}\|_1. \tag{5}$$

4. Experimental Setup

4.1. Data Set

To evaluate the effectiveness of the SEGAN, we resort to the data set by Valentini et al. [27]. We choose it because it is open and available[1], and because the amount and type of data fits our purposes for this work: generalizing on many types of noise for many different speakers. The data set is a selection of 30 speakers from the Voice Bank corpus [28]: 28 are included in the train set and 2 in the test set.

To make the noisy training set, a total of 40 different conditions are considered [27]: 10 types of noise (2 artificial and 8 from the Demand database [29]) with 4 signal-to-noise ratio (SNR) each (15, 10, 5, and 0 dB). There are around 10 different sentences in each condition per training speaker. To make the test set, a total of 20 different conditions are considered [27]: 5 types of noise (all from the Demand database) with 4 SNR each (17.5, 12.5, 7.5, and 2.5 dB). There are around 20 different sentences in each condition per test speaker. Importantly, the test set is totally unseen by (and different from) the training set, using different speakers and conditions.

4.2. SEGAN Setup

The model is trained for 86 epochs with RMSprop [30] and a learning rate of 0.0002, using an effective batch size of 400. We structure the training examples in two pairs (Fig. 3): the real pair, composed of a noisy signal and a clean signal ($\tilde{\mathbf{x}}$ and $\mathbf{x}$), and the fake pair, composed of a noisy signal and an enhanced signal ($\tilde{\mathbf{x}}$ and $\hat{\mathbf{x}}$). To adequate the data set files to our waveform generation purposes, we down-sample the original utterances
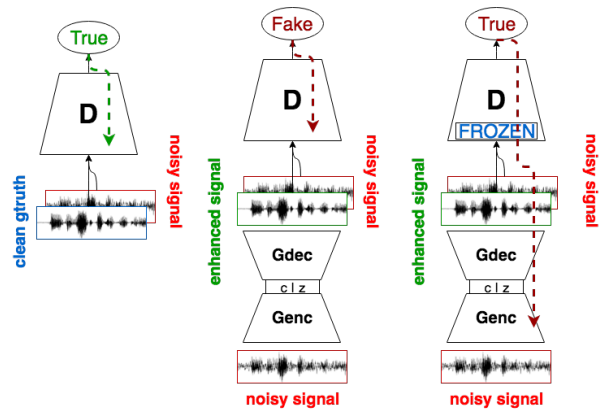
Figure 3: *Adversarial training for speech enhancement. Dashed lines represent gradient backprop.*

from 48 kHz to 16 kHz. During train, we extract chunks of waveforms with a sliding window of approximately one second of speech (16384 samples) every 500 ms (50% overlap). During test, we basically slide the window with no overlap through the whole duration of our test utterance and concatenate the results at the end of the stream. In both train and test, we apply a high-frequecy preemphasis filter of coefficient 0.95 to all input samples (during test, output is correspondingly deemphasized).

Regarding the λ weight of our L_1 regularization, after some experimentation, we set it to 100 for the whole training. We initially set it to 1, but we observed that the G loss was two orders of magnitude under the adversarial one, so the L_1 had no practical effect on the learning. Once we set it to 100, we saw a minimization behavior in the L_1 and an equilibrium behavior in the adversarial one. As the L_1 got lower, the quality of the output samples increased, which we hypothesize helped G being more effective in terms of realistic generation.

Regarding the architecture, G is composed of 22 one-dimensional strided convolutional layers of filter width 31 and strides of $N = 2$. The amount of filters per layer increases so that the depth gets larger as the width (duration of signal in time) gets narrower. The resulting dimensions per layer, being it samples × feature maps, is 16384×1, 8192×16, 4096×32, 2048×32, 1024×64, 512×64, 256×128, 128×128, 64×256, 32×256, 16×512, and 8×1024. There, we sample the noise samples $\mathbf{z}$ from our prior 8×1024-dimensional normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. As mentioned, the decoder stage of G is a mirroring of the encoder with the same filter widths and the same amount of filters per layer. However, skip connections and the addition of the latent vector make the number of feature maps in every layer to be doubled.

The network D follows the same one-dimensional convolutional structure as G's encoder stage, and it fits to the conventional topology of a convolutional classification network. The differences are that (1) it gets two input channels of 16384 samples, (2) it uses virtual batch-norm [31] before LeakyReLU nonlinearities with $\alpha = 0.3$, and (3) in the last activation layer there is a one-dimensional convolution layer with one filter of width one that does not downsample the hidden activations (1×1 convolution). The latter (3) reduces the amount of parameters required for the final classification neuron, which is fully connected to all hidden activations with a linear behavior. This means that we reduce the amount of required parameters in that fully-connected component from $8 \times 1024 = 8192$ to 8, and

Table 1: *Objective evaluation results comparing the noisy signal and the Wiener- and SEGAN-enhanced signals.*

Metric	Noisy	Wiener	SEGAN
PESQ	1.97	2.22	2.16
CSIG	3.35	3.23	3.48
CBAK	2.44	2.68	2.94
COVL	2.63	2.67	2.80
SSNR	1.68	5.07	7.73

the way in which the 1024 channels are merged is learnable in the parameters of the convolution.

All the project is developed with TensorFlow [32], and the code is available at `https://github.com/santi-pdp/segan`. We refer to this resource for further details of our implementation. A sample of the enhanced speech audios is provided at `http://veu.talp.cat/segan`.

5. Results

5.1. Objective Evaluation

To evaluate the quality of the enhanced speech, we compute the following objective measures (the higher the better). All metrics compare the enhanced signal with the clean reference of the 824 test set files. They have been computed using the implementation included in [1], and available at the publisher website[2].

- PESQ: Perceptual evaluation of speech quality, using the wide-band version recommended in ITU-T P.862.2 [33] (from –0.5 to 4.5).
- CSIG: Mean opinion score (MOS) prediction of the signal distortion attending only to the speech signal [34] (from 1 to 5).
- CBAK: MOS prediction of the intrusiveness of background noise [34] (from 1 to 5).
- COVL: MOS prediction of the overall effect [34] (from 1 to 5).
- SSNR: Segmental SNR [35, p. 41] (from 0 to ∞).

Table 1 shows the results of these metrics. To have a comparative reference, it also shows the results of these metrics when applied directly to the noisy signals and to signals filtered using the Wiener method based on a priori SNR estimation [36], as provided in [1]. It can be observed how SEGAN gets slightly worse PESQ. However, in all the other metrics, which better correlate with speech/noise distortion, SEGAN outperforms the Wiener method. It produces less speech distortion (CSIG) and removes noise more effectively (CBAK and SSNR). Therefore, it achieves a better tradeoff between the two factors (COVL).

5.2. Subjective Evaluation

A perceptual test has also been carried out to compare SEGAN with the noisy signal and the Wiener baseline. For that, 20 sentences were selected from the test set. As the database does not indicate the amount and type of noise for each file, the selection was done by listening to some of the provided noisy files, trying to balance different noise types. Most of the files have low SNR, but a few with high SNR were also included.

A total of 16 listeners were presented with the 20 sentences

Table 2: *Subjective evaluation results comparing the noisy signal and the Wiener- and SEGAN-enhanced signals.*

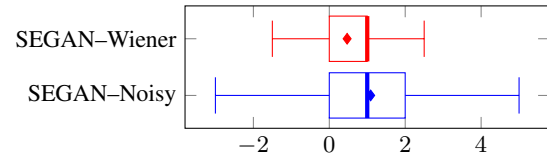| Metric	Noisy	Wiener	SEGAN
MOS	2.09	2.70	3.18

Figure 4: *CMOS box plot (the median line in the SEGAN–Wiener comparison is located at 1). Positive values mean that SEGAN is preferred.*

in a randomized order. For each sentence, the following three versions were presented, also in random order: noisy signal, Wiener-enhanced signal, and SEGAN-enhanced signal. For each signal, the listener rated the overall quality, using a scale from 1 to 5. In the description of the 5 categories, they were instructed to pay attention to both the signal distortion and the noise intrusiveness (e.g., 5=*excellent: very natural speech with no degradation and not noticeable noise*). Listeners could listen to each signal as many times as they wanted, and were asked to pay attention to the comparative rate of the three signals.

In Table 2, it can be observed how SEGAN is preferred over both the noisy signal and the Wiener baseline. However, as there is a large variation in the SNR of the noisy signal, the MOS range is very large, and the difference between Wiener and SEGAN is not significant. However, as the listeners compared all the systems at same time, it is possible to compute the comparative MOS (CMOS) by subtracting the MOS of the two systems being compared. Fig. 4 depicts this relative comparison. We can see how the signals generated by SEGAN are preferred. More specifically, SEGAN is preferred over the original (noisy) signal in 67% of the cases, while the noisy signal is preferred in 8% of the cases (no preference in 25% of the cases). With respect to the Wiener system, SEGAN is preferred in 53% of cases and Wiener is preferred in 23% of the cases (no preference in 24% of the cases).

6. Conclusions

In this work, an end-to-end speech enhancement method has been implemented within the generative adversarial framework. The model works as an encoder-decoder fully-convolutional structure, which makes it fast to operate for denoising waveform chunks. The results show that, not only the method is viable, but it can also represent an effective alternative to current approaches. Possible future work involves the exploration of better convolutional structures and the inclusion of perceptual weightings in the adversarial training, so that we reduce possible high frequency artifacts that might be introduced by the current model. Further experiments need to be done to compare SEGAN with other competitive approaches.

7. Acknowledgements

8. References

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

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

[3] D. Yu, L. Deng, J. Droppo, J. Wu, Y. Gong, and A. Acero, "A minimum-mean-square-error noise reduction algorithm on mel-frequency cepstra for robust speech recognition," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2008, pp. 4041–4044.

[4] A. L. Maas, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen, and A. Y. Ng, "Recurrent neural networks for noise reduction in robust asr." in *Proc. of INTERSPEECH*, 2012, pp. 22–25.

[5] J. Ortega-Garcia and J. Gonzalez-Rodriguez, "Overview of speech enhancement techniques for automatic speaker recognition," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, vol. 2, Oct 1996, pp. 929–932 vol.2.

[6] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," in *Proc. of the Int. Conf. on Acoustics, Speech, and Signal Processing (ICASS)*, vol. 4, Apr 1979, pp. 208–211.

[7] J. Lim and A. Oppenheim, "All-pole modeling of degraded speech," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 26, no. 3, pp. 197–210, Jun 1978.

[8] Y. Ephraim, "Statistical-model-based speech enhancement systems," *Proceedings of the IEEE*, vol. 80, no. 10, pp. 1526–1555, Oct 1992.

[9] M. Dendrinos, S. Bakamidis, and G. Carayannis, "Speech enhancement from noise: A regenerative approach," *Speech Communication*, vol. 10, no. 1, pp. 45–57, 1991.

[10] Y. Ephraim and H. L. Van Trees, "A signal subspace approach for speech enhancement," *IEEE Trans. on speech and audio processing*, vol. 3, no. 4, pp. 251–266, 1995.

[11] S. Tamura and A. Waibel, "Noise reduction using connectionist models," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 1988, pp. 553–556.

[12] S. Parveen and P. Green, "Speech enhancement with missing data techniques using recurrent neural networks," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2004, pp. 733–736.

[13] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder." in *Proc. of INTERSPEECH*, 2013, pp. 436–440.

[14] 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," in *Proc. of the Int. Conf. on Latent Variable Analysis and Signal Separation*, 2015, pp. 91–99.

[15] 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*, vol. 23, no. 1, pp. 7–19, 2015.

[16] A. Kumar and D. Florencio, "Speech enhancement in multiple-noise conditions using deep neural networks," in *Proc. of the Int. Speech Communication Association Conf. (INTERSPEECH)*, 2016, pp. 3738–3742.

[17] D. Wang and J. Lim, "The unimportance of phase in speech enhancement," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 30, no. 4, pp. 679–681, Aug 1982.

[18] K. Paliwal, K. Wójcicki, and B. Shannon, "The importance of phase in speech enhancement," *Speech Communication*, vol. 53, no. 4, pp. 465 – 494, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167639310002086

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

[20] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," ArXiv: 1611.07004, 2016.

[21] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, and Z. Wang, "Least squares generative adversarial networks," ArXiv: 1611.04076, 2016.

[22] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV)*, 2015, pp. 1026–1034.

[24] ——, "Deep residual learning for image recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[25] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR abs/1609.03499*, 2016.

[26] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2536–2544.

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

[28] C. Veaux, J. Yamagishi, and S. King, "The voice bank corpus: Design, collection and data analysis of a large regional accent speech database," in *Int. Conf. Oriental COCOSDA, held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*. IEEE, 2013, pp. 1–4.

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

[30] 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 4, 2, 2012.

[31] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems (NIPS)*, 2016, pp. 2226–2234.

[32] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[33] *P.862.2: Wideband extension to Recommendation P.862 for the assessment of wideband telephone networks and speech codecs*, ITU-T Std. P.862.2, 2007.

[34] Y. Hu and P. C. Loizou, "Evaluation of objective quality measures for speech enhancement," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 229–238, Jan 2008.

[35] S. R. Quackenbush, T. P. Barnwell, and M. A. Clements, *Objective Measures of Speech Quality*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[36] P. Scalart and J. V. Filho, "Speech enhancement based on a priori signal to noise estimation," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, May 1996, pp. 629–632 vol. 2.