iSEGAN: Improved Speech Enhancement Generative Adversarial Networks

Deepak Baby
Idiap Research Institute, Martigny, Switzerland

***Abstract*—Popular neural network-based speech enhancement systems operate on the magnitude spectrogram and ignore the phase mismatch between the noisy and clean speech signals. Conditional generative adversarial networks (cGANs) show promise in addressing the phase mismatch problem by directly mapping the raw noisy speech waveform to the underlying clean speech signal. However, stabilizing and training cGAN systems is difficult and they still fall short of the performance achieved by the spectral enhancement approaches. This paper investigates whether different normalization strategies and one-sided label smoothing can further stabilize the cGAN-based speech enhancement model. In addition, we propose incorporating a Gammatone-based auditory filtering layer and a trainable pre-emphasis layer to further improve the performance of the cGAN framework. Simulation results show that the proposed approaches improve the speech enhancement performance of cGAN systems in addition to yielding improved stability and reduced computational effort.**

***Index Terms*—speech enhancement, end-to-end models, generative adversarial networks, convolutional neural networks**

I. Introduction

SPEECH enhancement systems aim to improve the quality and intelligibility of acquired speech signals by removing artefacts caused by background noise or other interferences such as room reverberation. Recently, deep neural network (DNN)-based approaches gained much success in speech enhancement due to their powerful modeling capabilities [1]–[5].

DNN-based systems are typically trained to estimate a time-frequency (T-F) mask in the range $[0,\ 1]$, which provides the relative amplitudes of the underlying clean speech and noise signals at every T-F point [1,6]. However, these masks modify only the magnitude spectra of the input signal and ignore the phase mismatch between the noisy and clean speech signals [1,7]. Since speech quality can be significantly improved when the clean phase spectrum is known [8], it is worthwhile exploring speech enhancement techniques which preserve phase information. To remedy this phase mismatch problem, this paper investigates the use of generative neural networks which can directly map the raw noisy speech waveform to the underlying clean speech waveform.

In particular, we use generative adversarial networks (GAN) [9] which consists of a generative model or *generator network* (G) and a *discriminator network* (D) that play a min-max game between each other. D is trained to distinguish the samples generated by G from the real data. G, on the other hand, is trained to fool D into accepting its outputs being real. It was demonstrated that GANs can produce realistic image samples [9]. However, there is no control on the data being generated in such an unconditional generative model. For speech enhancement tasks, we have to control the generated data based on the input noisy data. In order to address this, conditional GANs (cGANs) [10] provide an alternative framework in which the model is conditioned to control the data generation process based on input data.

cGANs have recently been shown to yield promising noise suppression performance [11]–[14]. The technique presented in [12] is based on the pix2pix architecture [15] where the cGAN is trained to generate the spectrogram of clean speech given the noisy speech spectrogram and this technique otherwise ignores the phase mismatch problem. The speech enhancement GAN (SEGAN) system proposed in [11] is a 1D adaptation of the pix2pix architecture that operates on the raw waveform. However, the performance of cGAN-based models is still worse than conventional magnitude spectral enhancement approaches. In addition, training GANs is complex as it requires finding a Nash equilibrium of a non-convex game between G and D [9,16], and these prior works do not provide much insights on how to achieve this equilibrium.

This paper makes use of the SEGAN model [11] as the baseline system and systematically investigates approaches to further stabilize cGAN-based speech enhancement training and improve its performance. The SEGAN setting made use of virtual batch normalization (VBN) [16] in D to stabilize the training which is computationally expensive in memory and training time. Instead, we propose using instance normalization [17] which requires fewer computational resources. In addition, we investigate the use of one-sided label smoothing [16] to further stabilize the GAN training. Lastly, since the ultimate goal of our system is to improve speech intelligibility, we also propose using a trainable auditory filter-bank layer based on Gammatone filter-banks that approximates the cochlear processing in both G and D.

The contributions of this work are threefold. We Present a cGAN-based speech enhancement framework for further research and development, which 1) introduces instance normalization and one-sided label smoothing for training cGAN-based speech enhancement systems, 2) incorporates trainable auditory filtering and pre-emphasis layers to further improve the enhancement quality, and 3) provides an overview of various stabilization methods involved in GAN training.

email: deepak.baby@idiap.ch

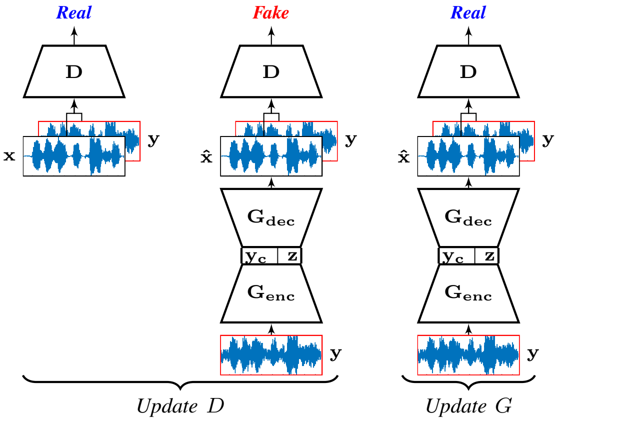

Fig. 1. Training a cGAN-based speech enhancement system. The updates for D and G are alternated over several epochs. $\mathbf{y}$, $\mathbf{x}$ and $\hat{\mathbf{x}}$ are the noisy speech, clean speech and the clean speech estimate generated by G, respectively. $\mathbf{y_c}$ is the encoder output of noisy speech and $\mathbf{z}$ are samples from the prior distribution $\mathcal{Z}$. Due to the adversarial training, G updates its parameters such that it generates samples that are closer to the clean speech manifold.

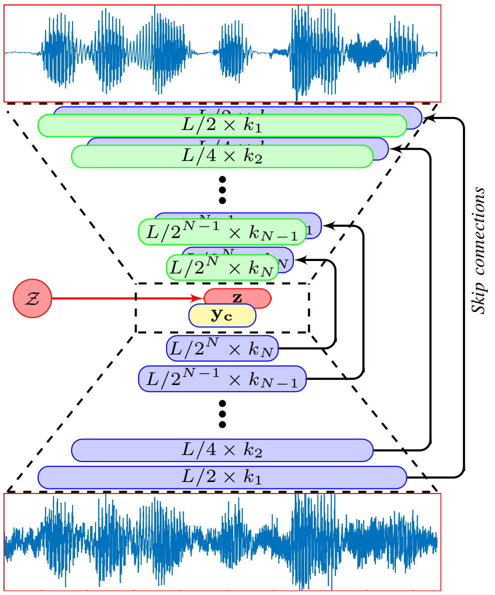

Fig. 2. Generator architecture: Encoder-decoder structure featuring U-shaped skip-connections employed for speech enhancement. Normalization and activation layers are omitted. The arrows denote the connection between the layers. The output shapes of each layer are also provided, where L and k_n are the length of the input signal and the number of feature-maps at the n^{th} layer, respectively. $\mathbf{y_c}$ denotes the encoder output corresponding to the input noisy speech signal and $\mathbf{z}$ are the samples from the prior distribution $\mathcal{Z}$.

II. Speech Enhancement cGAN Framework

The goal of a speech enhancement system is to estimate the clean speech signal $\mathbf{x}$ from the noisy mixture $\mathbf{y} = \mathbf{x} + \mathbf{w}$, where $\mathbf{w}$ is the added background noise.

In the generic GAN model, G acts as a generative model that learns to map samples $\mathbf{z}$ from some prior distribution $\mathcal{Z}$ to samples $\mathbf{x}$ that belong to a data distribution of interest $\mathcal{X}$ (i.e., the distribution of the clean speech samples, in our case). D is a binary classifier that is trained to classify samples from the true data distribution as real and the generated samples from G as fake. Since G is trained to fool D so that D classifies G's output as real, G will in turn learn to generate samples that are closer to the real data manifold. With cGANs, we direct this data generation process based on the input noisy speech $\mathbf{y}$ such that G generates an estimate that is closer to the underlying clean speech signal $\mathbf{x}$ (denoted as $\hat{\mathbf{x}} \triangleq G(\mathbf{y}, \mathbf{z})$).

The training phases of a cGAN-based speech enhancement system are depicted in Fig. 1. Notice that D is conditioned using the noisy speech signal $\mathbf{y}$ and G makes use of an encoder-decoder structure. The encoder (G_{enc}) projects the input noisy signal into a condensed representation $\mathbf{y_c} = G_{\text{enc}}(\mathbf{y})$, which is concatenated with the latent samples $\mathbf{z}$. The decoder (G_{dec}) then reconstructs the signal such that its output $\hat{\mathbf{x}} = G_{\text{dec}}(\mathbf{y_c}, \mathbf{z})$ fools D into classifying it as real. As can be seen from Fig. 1, training a cGAN-based speech enhancement setting is comprised of repeating the following three updates for every mini-batch over several epochs (encoding real as 1 and fake as 0):

1) Update D such that $\mathbf{x}$ and $\mathbf{y}$ pairs are classified as real, i.e., $D(\mathbf{x}, \mathbf{y}) \to 1$
2) Update D such that the generated samples $\hat{\mathbf{x}}$ and $\mathbf{y}$ pairs are classified as fake, i.e., $D(\hat{\mathbf{x}}, \mathbf{y}) \to 0$
3) Freeze D and update G such that D classifies $\hat{\mathbf{x}}$ and $\mathbf{y}$ pairs as real, i.e., $D(\hat{\mathbf{x}}, \mathbf{y}) \to 1$

For updating the G and D-networks, we use least-squares GAN (LSGAN) [18] which substitutes the conventional cross-entropy loss of the binary classifier D by least-squares. It has been shown that LSGANs further stabilize the GAN training and improve the quality of the generated samples in G. In addition, several prior works [11,12,15] use an additional loss term that minimizes the L1 distance between the generated samples $\hat{\mathbf{x}}$ and the clean examples $\mathbf{x}$. This L1 term is controlled by a new hyper-parameter λ. Thus, the loss functions used for updating D and G are,

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

A. G-network

In the cGAN-based speech enhancement framework, G performs the enhancement. The G architecture employed is depicted in Fig. 2. Similar to prior works [11,12], G is designed to be fully convolutional which enforces the network to focus on temporally-close correlations in the input signal. G_{enc} projects and compresses the input noisy signal through several strided convolutional layers followed by a parametric rectified linear unit (PReLU) non-linearity [19]. Strided convolution (stride > 1) is preferred over other pooling approaches as they provide a more stable GAN training [20]. G_{dec} uses an inverted version of G_{enc} by means of fractional-strided deconvolutions, followed by PReLUs.

Also notice that G uses U-shaped skip-connections that bypass the intermediate compression stages (Fig. 2). These skip connections directly pass the fine-grained information such as phase and alignment to the decoder. They also provide a better training behavior as the gradients can flow deeper through the whole network [21].

B. D-network

D-network makes use of the same structure as G_{enc}, but with a few differences: 1) it has two input channels (one for $\mathbf{x}$ or $\hat{\mathbf{x}}$; and one for $\mathbf{y}$), 2) it uses a normalization layer before the non-linearity, 3) it uses LeakyReLU non-linearity instead of PReLU, and 4) there is an additional convolutional layer with one filter of width 1 (1×1 convolution) and its output is fed to a fully-connected layer to perform the binary classification.

C. Proposed cGAN variants

1) Instance normalization: An instance normalization layer [17] applies mean-variance normalization on every channel and input sample. It was successfully used for image stylization [17] and dehazing [22] and requires less computational complexity than VBN. Motivated by this, we propose to use instance normalization in D for training the cGAN model. To our knowledge, instance normalization has not yet been investigated for cGAN-based speech enhancement.

2) One-sided label smoothing: One of the critical scenarios which results in unstable GAN training is when D becomes too confident on the real examples, such that G no longer can fool it. One simple trick to remedy this is to encourage D to estimate soft probabilities on real samples, for e.g., $D(\mathbf{x}, \mathbf{y}) \rightarrow 0.9$ instead of 1. This solution avoids overpowering of D over G and could stabilize GAN training. This approach is called one-sided label smoothing [16] since only the confidence on real samples is modified.

3) Auditory filter-bank layer: The ultimate goal of speech enhancement systems is to improve speech intelligibility. However, existing cGAN-based systems [11,12] which are simple adaptations of the pix2pix architecture do not take this goal into account and provide full freedom to G and D. In this work, we replace the first layer of both G and D with an auditory filter-bank layer that mimics human auditory processing. Similar ideas have been used for speech recognition applications [23,24], but have not yet been used in this context. We make use of a Gammatone-based model for the cochlear filtering and use it to initialize the input layers.

4) Pre-emphasis layer: In most speech processing applications, it is beneficial to boost the high-frequency signal content by means of a pre-emphasis filter. This is typically implemented as a first oder high-pass filter $\tilde{y}[n] = y[n] - \alpha y[n-1]$, with $0.9 \leq \alpha < 1$. Instead of using a fixed α, we propose to optimize it by implementing the pre-emphasis filter in G as a trainable convolutional layer of filter-length 2 and stride 1. The layer is initialized with weights $[-0.95,\ 1]$ and is trained together with the cGAN network.

5) Removing the latent vector $\mathbf{z}$: It is observed in [15] that adding the latent vector $\mathbf{z}$ for image processing applications is sometimes not effective as the generator simply learns to ignore it. Some prior works on cGAN-based speech enhancement [13,14] therefore omitted $\mathbf{z}$ such that cGAN generates deterministic outputs. However, it is to date unclear whether $\mathbf{z}$ is helpful for speech enhancement applications. To investigate this, a comparison of all the proposed algorithms with and without $\mathbf{z}$ is included.

III. Evaluation Setup

A. Database

The experiments were performed on the data set presented in [25]. The database is derived from the voice bank corpus [26] from which recordings from 28 speakers were chosen for the training set and 2 for the test set. The recordings were added with 10 different noise conditions (2 artificial and 8 from the DEMAND database [27]) at signal-to-noise ratios (SNRs) of 0, 5, 10 and 15 dB. Thus the training set simulates 40 different noisy scenarios and is comprised of a total of 11 572 recordings. The test set was created using 5 noise conditions (all from the DEMAND database, but different from training noise conditions) added at SNRs 2.5, 7.5, 12.5 and 17.5 dB. Altogether, the test set contains 824 utterances. The database was downsampled from 48 kHz to 16 kHz for our experiments.

B. cGAN setup

This work used 11 convolutional layers each for G_{enc} and G_{dec} with filter-length 31 and stride $= 2$. Thus, after every layer, the temporal dimension of the features gets halved (Fig. 2). We operated on signals that were sampled at 16 kHz and considered approximately 1 second of speech (16 384 samples) as input to the network. Thus, after 11 convolutional layers in G_{enc} the temporal dimension shrunk to $16\ 384/2^{11} = 8$.

The number of feature-maps (k_i in Fig. 2) used in the convolutional layers were: 16, 32, 32, 64, 64, 128, 128, 256, 256, 512 and 1024. Thus the encoder output was of size 8×2014 which was then concatenated with a latent vector of the same size. G_{dec} followed the reverse procedure together with skip connections that doubled the temporal dimension after every layer resulting in a final output size that was identical to that of the input noisy signal.

As mentioned in Section II, D used the same structure as G_{enc}, but with two input channels of 16 384 samples each. The output of the convolutional layer (of size 8×1024) was fed to another convolutional layer of filter length 1 and stride 1, resulting in a representation of size 8×1. This was fed to a fully-connected layer for classification.

The model was trained using the Adam optimizer [28] (as opposed to RMSProp used in SEGAN [11]) for 80 epochs with a learning rate of 0.0002 using a batch-size of 100. The speech signals were windowed using sliding windows of length 16 384 with 50% overlap. During testing, the enhanced signals were reconstructed by adding the generated signals with the same overlap and dividing the overlapping sections by 2 to compensate for the 50% overlap. We also applied a pre-emphasis filter with $\alpha = 0.95$ to all input samples, except for the trainable pre-emphasis layer setting where the input to G was not pre-emphasized.

Similar to [11], the λ parameter that controls the L1 loss was set to 100. The latent noise input $\mathbf{z}$ of size 8×1024 was drawn from a normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The whole project

TABLE I

Comparison of the different GAN-based speech enhancement systems. Note that a higher value means a better performance for all the measures except CD and LLR. The best results obtained are highlighted in bold font.

Setting	**STOI**	**PESQ**	**CD**	**LLR**	**segSNR**	**STOI**	**PESQ**	**CD**	**LLR**	**segSNR**
Unprocessed	0.921	1.97	4.41	0.46	8.77	0.921	1.97	4.41	0.46	8.77
LSTM-IRM [6]	0.931	2.48	**2.76**	0.33	15.73	0.931	2.48	**2.76**	**0.33**	15.73
	With latent vector					*Without latent vector*				
SEGAN [11]	0.928	2.16	3.35	0.48	15.69	0.925	2.18	3.39	0.44	15.43
IN	0.933	2.49	3.11	0.43	16.66	0.934	2.50	3.11	0.44	16.45
IN + LabSmth	0.938	2.53	3.04	0.43	17.01	0.938	2.54	3.20	0.34	16.68
IN + GT	**0.940**	2.59	2.96	0.38	17.00	0.939	**2.62**	2.96	**0.32**	**17.28**
IN + PreEm	0.939	**2.64**	3.06	0.37	17.13	**0.941**	2.57	3.20	0.35	16.42
IN + LabSmth + GT + PreEm	0.939	2.60	3.04	0.39	**17.31**	*Unstable*				

was developed in Keras [29] with Tensorflow [30] back-end and is made available on github[1].

C. Evaluation metrics

The speech enhancement performance was evaluated using the following measures: the short-term objective intelligibility (STOI) metric [31], perceptual evaluation of speech quality (PESQ) [32] in terms of mean opinion score (MOS), segmental SNR (segSNR), cepstral distance (CD) and log-likelihood ratio (LLR). The CD, LLR and segSNR measures are expressed in dB and were obtained using the implementations provided with the REVERB challenge [33]. Higher values of PESQ, STOI and segSNR, and lower values of CD and LLR indicate better performance.

IV. Results and Discussion

The noise suppression performance obtained for the various speech enhancement systems in terms of various speech quality measures are provided in Table I. To compare cGAN with the conventional magnitude spectral enhancement approach, an LSTM-based speech enhancement system that estimates the ideal-ratio mask (IRM) for enhancing the Gammatone spectrogram of noisy speech is also included. The details of the model are provided in [6].

It can be seen that the LSTM-IRM model outperforms the SEGAN model, even though the former reused the noisy phase for reconstructing the time-domain signal. Using instance normalization (*IN* in Table I) instead of VBN reduced the training time considerably and resulted in a better performance than using VBN. Using one-sided label smoothing (denoted as *LabSmth*) together with instance normalization yielded a more stable GAN training and this approach outperformed the LSTM-IRM baseline model (except for CD and LLR).

Incorporating the trainable auditory filterbank (*GT*) and pre-emphasis (*PreEm*) layers yielded further improvements even without one-sided label smoothing. To our knowledge, this is the first time where such speech processing principles are incorporated in a cGAN-based speech enhancement system. The results show that this approach greatly benefits in stabilizing the model and yields improved state-of-the-art speech enhancement performance.

It can also be seen that using the latent vector $\mathbf{z}$ sometimes yield a better performance (*with latent vector* vs. *without latent vector*), suggesting that G does not always learn to ignore the latent vector. Moreover, removing the latent vector made the last system unstable and the model failed to achieve any equilibrium. The effect of latent vectors in cGAN models requires further investigation, which is beyond the scope of this paper. In our experiments, the best performance was achieved using instance normalization with the trainable pre-emphasis layer and the latent vector. However, combining all the cGAN variants (*InstNorm* + *LabSmooth* + *GTLayer* + *PreEmLayer*) yielded a slightly worse performance as it resulted in a different equilibrium state as compared to the other settings.

V. Conclusions and Future Work

This paper investigated several approaches to improve the performance of end-to-end raw speech waveform enhancement using cGANs. First, we investigated using instance normalization instead of VBN in order to reduce the computational complexity during the model training. The investigated cGAN model that combined one-sided label smoothing with instance normalization was shown to outperform a VBN-based cGAN model and a LSTM-IRM-based speech enhancement system. To our knowledge, this is the first time a cGAN-based speech enhancement system is shown to outperform a popular IRM-based approach. This paper also showed that using trainable Gammatone-based auditory filtering and pre-emphasis layers also can stabilize the model as well as improve its performance.

Since the proposed models are shown to outperform the popular IRM-based models, using the proposed cGAN-based models as a front-end for automatic speech recognition systems is a suggested future work. The project published in github offers more flexibility in terms of combining different auditory model-based initializations and other normalization approaches such as batch renormalization and group normalization. Investigating different combinations and applying them for dereverberation is also a promising research direction.

[1] The proposed cGAN-based speech enhancement framework is available at `https://github.com/deepakbaby/isegan`

References

[1] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1849–1858, Dec 2014.

[2] Y. Xu, J. Du, L. Dai, and C. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech & Language Processing*, vol. 23, no. 1, pp. 7–19, 2015.

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

[4] L. Sun, J. Du, L. Dai, and C. Lee, "Multiple-target deep learning for LSTM-RNN based speech enhancement," in *Hands-free Speech Communications and Microphone Arrays, HSCMA*, Mar 2017, pp. 136–140.

[5] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Deep recurrent networks for separation and recognition of single-channel speech in nonstationary background audio," in *New Era for Robust Speech Recognition, Exploiting Deep Learning.*, 2017, pp. 165–186.

[6] D. Baby and S. Verhulst, "Biophysically-inspired features improve the generalizability of neural network-based speech enhancement systems," in *Proc. INTERSPEECH*. ISCA, Sep 2018.

[7] A. Narayanan and D. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 7092–7096.

[8] K. Paliwal, K. Wójcicki, and B. Shannon, "The importance of phase in speech enhancement," *Speech Communication*, vol. 53, no. 4, pp. 465 – 494, 2011.

[9] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 27, Dec 2014, pp. 2672–2680.

[10] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *CoRR*, vol. abs/1411.1784, 2014.

[11] S. Pascual, A. Bonafonte, and J. Serrà, "SEGAN: speech enhancement generative adversarial network," in *Proc. INTERSPEECH*. ISCA, Aug 2017, pp. 3642–3646.

[12] D. Michelsanti and Z. Tan, "Conditional generative adversarial networks for speech enhancement and noise-robust speaker verification," in *Proc. INTERSPEECH*, Aug 2017, pp. 2008–2012.

[13] C. Donahue, B. Li, and R. Prabhavalkar, "Exploring speech enhancement with generative adversarial networks for robust speech recognition," *CoRR*, vol. abs/1711.05747, 2017.

[14] K. Wang, J. Zhang, S. Sun, Y. Wang, F. Xiang, and L. Xie, "Investigating generative adversarial networks based speech dereverberation for robust speech recognition," *CoRR*, vol. abs/1803.10132, 2018.

[15] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017, pp. 5967–5976.

[16] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 29, Dec 2016, pp. 2226–2234.

[17] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *CoRR*, vol. abs/1607.08022, 2016.

[18] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 2813–2821.

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

[20] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *CoRR*, vol. abs/1511.06434, 2015.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016, pp. 770–778.

[22] Z. Xu, X. Yang, X. Li, and X. Sun, "The effectiveness of instance normalization: A strong baseline for single image dehazing," *CoRR*, vol. abs/1805.03305, 2018.

[23] Y. Hoshen, R. J. Weiss, and K. W. Wilson, "Speech acoustic modeling from raw multichannel waveforms," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr 2015, pp. 4624–4628.

[24] T. N. Sainath, R. J. Weiss, A. W. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform CLDNNs," in *Proc. INTERSPEECH*. ISCA, Sep 2015, pp. 1–5.

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

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

[27] J. Thiemann, N. Ito, and E. Vincent, "The diverse environments multi-channel acoustic noise database (DEMAND): A database of multi-channel environmental noise recordings," *Proceedings of Meetings on Acoustics*, vol. 19, no. 1, p. 035081, 2013.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[29] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[30] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

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

[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 International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2001, pp. 749–752.

[33] K. Kinoshita, M. Delcroix, S. Gannot, E. A. P. Habets, R. Haeb-Umbach, W. Kellermann, V. Leutnant, R. Maas, T. Nakatani, B. Raj, A. Sehr, and T. Yoshioka, "The REVERB challenge: A benchmark task for reverberation-robust ASR techniques," in *New Era for Robust Speech Recognition, Exploiting Deep Learning*. Springer, 2017, pp. 345–354.