LOW-LATENCY SINGLE CHANNEL SPEECH ENHANCEMENT USING U-NET CONVOLUTIONAL NEURAL NETWORKS

Ahmet E. Bulut[1,2], *Kazuhito Koishida*[2]

[1]Center for Robust Speech Systems, University of Texas at Dallas, TX 75080
[2]Microsoft Corporation, One Microsoft Way, Redmond, WA 98052

ahmet.bulut@utdallas.edu, kazukoi@microsoft.com

ABSTRACT

Single-channel speech enhancement (SE) can be described, in its simplest terms, as learning a transformation from single-channel noisy speech to the clean speech. To do this, we propose a simple but effective U-Net convolutional neural network (CNN) based architecture with skip-connections with a focus on real-time applications which require low-latency processing. To that end, we choose to process relatively small temporal windows and apply time-frequency (T-F) featurization on it to achieve magnitude estimation. Two state-of-the-art systems are picked for bench-marking: One operating on spectral-domain [1] and the other on temporal-domain [2]. We evaluate the performance of the systems in terms of perceptual evaluation of speech quality (PESQ), short-time objective intelligibility (STOI). Experimental results show that in terms of PESQ measure the proposed method provides around 27% and 11% relative improvement over the baseline systems respectively and has significantly lower latency compared to them. We further investigate the trade-off between performance and overall latency of the proposed system.

Index Terms— Speech enhancement, low-latency, U-Net, convolutional neural networks

1. INTRODUCTION

Single-channel speech enhancement (SE) is defined as the process of suppressing the background noise with little or no degradation on target speech so that improved sound quality and intelligibility is achieved. The SE algorithms are generally used as a pre-processing for various speech processing applications such as speech recognition [3, 4] and speaker recognition [5, 1]. For many such applications, real-time processing might be required and for this case, it has been recently studied that SE systems should be performed with low computational complexity and low-latency [6]. Traditional methods used in SE were concentrated on signal processing based methods [7, 8, 9] in order to tackle the problem, whereas the most recent studies mainly address various neural networks based methods such as deep neural networks (DNN) [10, 11], deep denoising autoencoders (DAE) [12], convolutional neural networks (CNN) [13], recurrent neural networks (RNN) [14, 3], and generative adversarial networks (GAN) [1, 15, 2] for the task.

Neural network based methods have gained quite an amount of attention in recent years for many applications (text/audio/ image processing etc.) due to its ability to learn complex hierarchical representations from data. Generally, the most recent state-of-the-art methods on SE can be classified as those operating on T-F domain and temporal domain. It is claimed that phase information coming out of short-time Fourier analysis has minor importance on SE [16]. However, further study [17] shows that phase information has significant importance on the reconstruction of the enhanced speech which constitutes the main motivation for the studies operating on the temporal domain. On the other hand, our experiments show that temporal processing requires more processing time at inference as opposed to the spectral. And there are also some studies [18, 19] working in complex numbers domain to take the phase information into consideration for the SE processing. The studies using spectral featurization mainly use direct mapping or masking strategy. For the ones using former strategy the network tries to learn to transform noisy magnitude spectra to their clean equivalents [10, 11] and for the ones that use latter strategy to predict corresponding masks such as ideal binary mask (IBM) [20], ideal ratio mask (IRM) [21], and phase sensitive mask (PSM) [4].

In recent years generative adversarial networks (GAN) [22] based methods have been adopted for various SE applications [1, 15, 2] operating on spectral or temporal domains. However, it is addressed [23] that for the system that Pascual et al. proposed [15] using the L_1 loss alone performs better in terms of objective perceptual speech quality and intelligibility measures. Actually we observe the similar behaviour for the study [1] which uses combined GAN and L_1 losses as well.

In this paper, we propose a simple and effective CNN based U-Net architecture [24] adopted from the generator of Pix2Pix network which is a recent general-purpose GAN framework proposed for image-to-image translation [25]. We operate on T-F domain and try to have relatively low pro-

The work was done when AEB was an intern at Microsoft Corporation.

978-1-5090-6631-5/20/$31.00 ©2020 IEEE

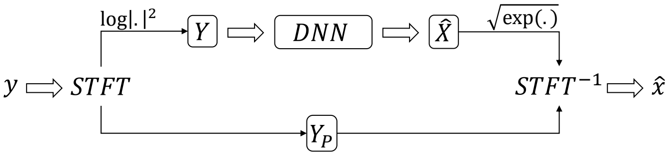

Fig. 1: System overview. y is the input noisy signal and Y is LPS of the signal and $\hat{X}$ is estimated LPS which combined with noisy phase Y_P to get estimated signal $\hat{x}$.

cessing window in the temporal dimension and construct an architecture tailored for the corresponding input. We choose to concentrate on magnitude prediction by disregarding phase information. Our proposed network includes encoder and decoder layers which fulfil the downsampling and upsampling of the input data respectively. We try various loss function such as L_1, L_2 norms and log-spectral distance (LSD) metric out of which LSD gives the best results. We analyse the objective speech quality of the systems and further investigate the processing time of baseline systems with respect to the proposed system at inference time. Moreover we analyse the performance change of the proposed system under various latency conditions.

This paper is organized as follows: We first describe the overview of the system and the details of the proposed network architecture in Section 2. And in Section 3 experimental details are explained as well as the information about the dataset and baseline methods. We present the results and some analysis in Section 4 and Section 5 concludes the paper.

2. PROPOSED SYSTEM

2.1. System Overview

The transformation function from noisy to clean feature space can be learned by means of a DNN which is trained on a dataset of parallel noisy and clean speech files. As an input to the network we use log-power spectrogram (LPS), Y which can be obtained after applying a short-time Fourier transform (STFT) to the input noisy waveform, y to get magnitude, Y_M and phase, Y_P and then calculated as $Y = \log(|Y_M|^2)$.

Within the scope of this study, at inference time, we only forward propagate the Y features through the network and reconstruct the enhanced signal $\hat{x}$ by applying the inverse STFT with estimated magnitude, $\hat{X}_M$ and noisy phase, Y_P as shown in Figure 1.

2.2. Network Architecture

The proposed network architecture is illustrated in Figure 2. To encode the input features we deploy 8 2-dimensional convolution (conv2d) layers named as (e1-e8) and to decode we apply 8 2-dimensional sub-pixel convolution (subconv2d) layers named as (d1-d8). The sub-pixel convolution layers are first introduced in [26] for an image and video super-resolution task and it is proved useful in speech super-resolution task [27] as well. The main idea is to compute more feature channels on the convolution layer and resize them into the target upsample dimension. To each layer we apply leaky rectified linear unit (LReLU) activation function followed by batch normalization. The stride values that applied for downsampling are $(1, 2)$ for first 4 layers and $(2, 2)$ for the other 4 layers, kernel sizes are $(5, 7), (5, 7), (5, 7), (5, 5), (5, 5)$ for first 5 layers and $(3, 3)$ for the rest (e6-e8), and number of output feature channels are $64, 128, 256$ for the first 3 layers and 512 for the rest (e4-e8), respectively. All stride, kernel, and channel values are symmetric for the subconv2d layers. At each downsampling step, the number of channels doubles up to 512 and reduces gradually down to 1 at the end of the upsampling steps. For the decoder layers (d1-d7) we apply skip-connections with corresponding encoder layers in reverse order which are (e7-e1), respectively. Moreover we apply dropout to the first 3 layers (d1-d3) with a probability rate of 0.5.

For the training loss, we experiment three type of functions, namely L_1, L_2 norms and log-spectral distance (LSD), our overall testing shows that LSD yields slightly better results for SE task. In general terms, LSD measures the distance between two spectrograms in decibels, and it is defined as follows:

$$loss_{\text{LSD}} = \frac{1}{T}\sum_{i=1}^{T}\sqrt{\frac{1}{S}\sum_{j=1}^{S}[X(i,j) - \hat{X}(i,j)]^2} \qquad (1)$$

where X and $\hat{X}$ are the clean and estimated LPS, respectively and T is the number of frames and S is the number of spectral bins.

3. EXPERIMENTS

3.1. Dataset

To benchmark our proposed system we resort to the dataset presented by Valentini et al. [28], which is publicly available[1] and also used by various recent studies [2, 15, 29, 30].

The dataset includes clean and noisy audio data at 48 kHz sampling frequency. The clean dataset is composed of 30 speakers (gender-balanced) selected from the voice bank corpus [31], 28 of which reserved for training and 2 intended for the test. The noisy dataset is created with 10 types of noise (2 artificial and 8 real obtained from the Demand database [32]) with varying signal-to-noise ratio (SNR) values of 15, 10, 5, and 0 dB. The total number of conditions for the training set sums up to 40 and total duration of the train set is around 10

[1] https://datashare.is.ed.ac.uk/handle/10283/2791

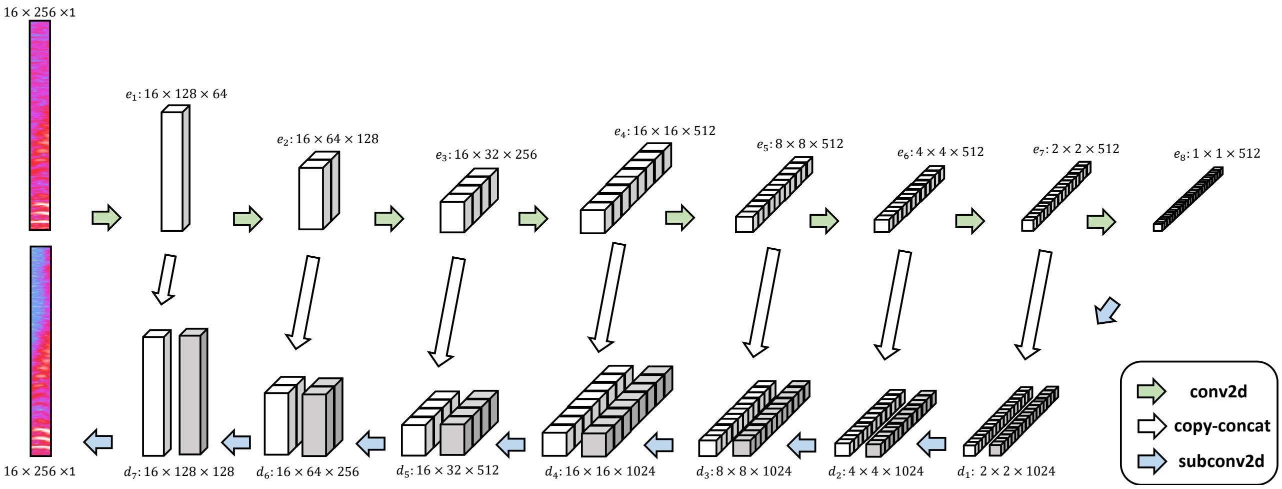

Fig. 2: Proposed network architecture.

hours. For the test set, 5 types of noise are also chosen from the Demand database and mixed with SNR values of 17.5, 12.5, 7.5, and 2.5 dB which gives rise to 20 different conditions and total duration of test is around 30 mins. It has to be stressed out that all the noise types and speakers are mutually exclusive between training and test conditions.

3.2. Preprocessing and Training Setup

We generally adopt the preprocessing from [1] with slight changes. The audio signals are downsampled from 48 kHz to 16 kHz. The spectral representation is obtained by applying 512-point STFT with a Hanning window of size 32 ms and a hop size of 16 ms (256 samples). Only the 257-point STFT magnitudes are considered by removing the symmetric half. We remove the last STFT point as well which covers the highest 31.25 Hz band of the signal and has slight importance. By doing so, a power-of-2 dimensional input is achieved which is instrumental in our hierarchical encoder-decoder network as described in Section 2.2. In order to achieve a fixed dimension for the processing of both train and test sets, we use 16 frames of clips which in turn creates an input dimension of 16 × 256 × 1 processing window. We choose relatively small, 16 frame-window (0.256 sec), on the temporal domain in order to meet our low-latency constraint. Inputs to the network are normalized to have zero mean and unit variance.

The network is trained with the Adam optimizer [33] with a batch size of 64 and learning rate of 0.0001 for 30 epochs. The decay rates of optimizer are $\beta_1 = 0.5$ and $\beta_2 = 0.9$. The weights of the network are initialized from the normal distribution with zero mean and 0.02 standard deviation [25].

3.3. Baseline Methods and Evaluation Metric

We compare the results of our proposed approach with two state-of-the-art methods both using the GAN based networks that we refer one, operating on the spectral domain [1], as SPcGAN and the other as SERGAN which operates on the temporal domain [2]. These two systems are retrained with the dataset described in Section 3.1 and by following the corresponding experimental configuration that they proposed.

The first baseline SPcGAN uses a network adopted from a general-purpose conditional GAN (cGAN) framework. We follow the proposed preprocessing procedure to create clean and noisy input data [1]. Then noisy input is fed to the generator (G) to get an estimate of clean spectra and discriminator (D) gets the combination of estimate/clean and noisy spectra as an input. Here G has U-Net architecture with skip connections [24]. General GAN training procedure and architecture are followed as described in [25] but 5 × 5 filters are used in 2-dimensional convolutional layers and single sigmoid output generated after the last layer of D flatten out. The Adam optimizer is used at the training for 10 epochs with a learning rate of 0.0002 and a batch size of 1. L_1 loss has been added to the GAN loss with a scaling factor of 100 as suggested in [25].

The SERGAN system also uses a cGAN framework which is similar architecture used in SPcGAN system but operates on input raw waveform instead of spectral domain. The model uses 16384-sample processing windows with 50% overlap. Regular cGAN training procedure is followed as described in the SPcGAN system, specifically relativistic standard GAN implementation is adopted and the gradient penalty is applied in the D network [34] to stabilize the training. The system trained with the Adam optimizer for 80 epochs with a learning rate of 0.0002 and batch size of 100. Here L_1 loss is also added to GAN loss with a penalty term of 200 with an additional gradient penalty term of 10.

The performance comparison of the systems is evaluated in terms of PESQ [35] and STOI [36] measures. These objective speech quality measures are generally the most well-known and used metrics by the SE community. The implementations used in this paper come from [37] for PESQ and [38] for STOI. In order to evaluate latency and processing time of the systems, the length of the processing window and real-time factor (RTF) are used respectively. We define the latency (L) as the summation of the shift length of the processing window (W) and the duration of time needed to process it. RTF is the most common speed performance metric for speech processing applications and defined as the ratio of processing time of the input over the actual duration of the input. Any application is considered real-time if its RTF is less than 1. The computer that is used for all latency and RTF calculations has an Intel Xeon Gold 6130 CPU @ 2.10GHz and a GeForce 2080 RTX Ti, 24GB GPU.

4. RESULTS AND DISCUSSION

We first analyze the perceptual speech quality/intelligibility and processing time performance comparison of the baseline methods and proposed method on the test data as shown in Table 1. We also include some results from the state-of-the-art papers that use the same train and test dataset. The SPcGAN system is the least performing system in terms of speech quality/intelligibility but has the best RTF score which makes it a good candidate for the offline systems working on the enhancement of a huge amount of data at a time. Because of the wide processing window (4096 ms) of the SPcGAN has the highest latency which makes it almost impossible to operate in a real-time scenario. The SERGAN system has relatively better PESQ/STOI measure and latency value but on the other hand, has around 4× higher RTF value as opposed to the SPcGAN system, our further experimentation shows that operation on the temporal domain is computationally more costly than the spectral domain. The proposed system has clearly the best PESQ value and has a comparable STOI measure as opposed to the baseline systems. By reducing the processing window considerably as compared to the baseline systems we compromise the RTF value a little bit but we achieve relatively good performance in terms of speech quality and intelligibility with the proposed simple but effective DNN architecture. Moreover, low latency and moderate RTF value make the proposed system a good candidate for the low-latency required systems. We have to note that for the SPcGAN and the proposed system both operate on the spectral domain and the latency calculation includes the sum of the duration of spectral featurization and reconstruction which does not exist in the SERGAN system because of the time-domain operation.

Clearly, it is always an option to further improve the latency by applying some processing tricks at inference time as long as RTF value stays less than 1 (real-time constraint). To achieve this we make some analysis on the speech quality performance trade-off with the latency by applying sliding shift on the processing window. As it is shown in Table 2 the first row includes the performance result of the system that uses 256 ms (16 frames) block-by-block processing and at following rows we keep reducing the shift size and lower the latency concurrently while observing RTF of the systems. Although the proposed network always processes 256 ms of the full input window and produces the corresponding prediction, only the last shift-size portion of the prediction is actually used for the output. Note that, in the initial few frames where there is not enough input data available to reach the input buffer size, it is filled by repeating the very first shift-size portion of the input. It can be observed that our proposed system is able to operate at real-time with a very low latency duration as little as 31 ms and with a tolerable degradation on speech quality.

Table 1: Performance comparison of the systems. **W:** Window Length, **L:** Latency, **RTF:** Real-time Factor

Systems	**PESQ**	**STOI**	**W / L (ms)**	**RTF**
Unprocessed	1.96	0.9211	-	-
SEGAN [15]	2.16	-	1024/-	-
Wave-U-Net [30]	2.40	-	1024/-	-
MMSE-GAN [39]	2.53	-	**70**/-	-
RaLSGAN-GP [2]	2.62	**0.9400**	1024/-	-
MDPhD [40]	2.70	-	1024/-	-
D+M [41]	2.73	-	1024/-	-
SPcGAN	2.28	0.9285	4096/4174	**0.019**
SERGAN	2.59	0.9380	1024/1122	0.096
Proposed*	**2.90**	0.9378	256/**272**	0.059

*__CSIG__, **CBAK**, and **COVL** [37] performance metric values of the proposed system are 4.22, 3.32, and 3.58, respectively for future reference.

Table 2: Latency (L) vs. performance analysis of proposed system with using various shifted processing windows. The first row shows the full window (16 frames) shift case.

Shift (ms)	**PESQ**	**STOI**	**W / L (ms)**	**RTF**
256	**2.90**	**0.9378**	256/272	**0.059**
128	2.88	0.9355	256/144	0.113
64	2.84	0.9323	256/81	0.230
32	2.80	0.9299	256/47	0.459
16	2.75	0.9240	256/**31**	0.915

5. CONCLUSION

In this paper we propose simple but effective U-Net CNN architecture specifically for the SE systems working under low-latency condition. We achieve superior results as opposed to the two GAN based baseline systems operating on spectral and temporal domains. And it has been shown that the proposed system has a real-time operation under extreme low-latency conditions while maintaining performance quality of the system to some extend.

6. REFERENCES

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

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

[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," in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2015, pp. 91–99.

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

[5] O. Plchot, L. Burget, H. Aronowitz, and P. Matejka, "Audio enhancing with dnn autoencoder for speaker recognition," in *ICASSP*. IEEE, 2016, pp. 5090–5094.

[6] K. Tan and D. Wang, "A convolutional recurrent neural network for real-time speech enhancement.," in *Interspeech*, 2018, pp. 3229–3233.

[7] N. Wiener, *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*, MIT Press, 1950.

[8] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 27, no. 2, pp. 113–120, 1979.

[9] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[10] Y. Xu, J. Du, L. Dai, and C. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal processing letters*, vol. 21, no. 1, pp. 65–68, 2013.

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

[12] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder.," *Interspeech*, pp. 436–440, 2013.

[13] S. R. Park and J. W. Lee, "A fully convolutional neural network for speech enhancement," *Interspeech*, pp. 1993–1997, 2017.

[14] F. Weninger, F. Eyben, and B. Schuller, "Single-channel speech separation with memory-enhanced recurrent neural networks," in *ICASSP*. IEEE, 2014, pp. 3709–3713.

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

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

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

[18] D. S. Williamson, Y. Wang, and D. Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM TASLP*, vol. 24, no. 3, pp. 483–492, 2016.

[19] D. S. Williamson and D. Wang, "Speech dereverberation and denoising using complex ratio masks," in *ICASSP*. IEEE, 2017, pp. 5590–5594.

[20] Y. Wang and D. Wang, "Towards scaling up classification-based speech separation," *IEEE TASLP*, vol. 21, no. 7, pp. 1381–1390, 2013.

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

[22] 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*, 2014, pp. 2672–2680.

[23] A. Pandey and D. Wang, "On adversarial training and loss functions for speech enhancement," in *ICASSP*. IEEE, 2018, pp. 5414–5418.

[24] 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*. Springer, 2015, pp. 234–241.

[25] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[26] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE conference on CVPR*, 2016, pp. 1874–1883.

[27] S. E. Eskimez, K. Koishida, and Z. Duan, "Adversarial training for speech super-resolution," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 347–358, 2019.

[28] C. Valentini-Botinhao et al., "Noisy speech database for training speech enhancement algorithms and tts models," *University of Edinburgh. School of Informatics. Centre for Speech Technology Research (CSTR)*, 2017.

[29] 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 SSW*, 2016, pp. 146–152.

[30] C. Macartney and T. Weyde, "Improved speech enhancement with the wave-u-net," *arXiv preprint arXiv:1811.11307*, 2018.

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

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

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[34] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in neural information processing systems*, 2017, pp. 5767–5777.

[35] 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*. IEEE, 2001, vol. 2, pp. 749–752.

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

[37] P. C. Loizou, *Speech enhancement: theory and practice*, CRC press, 2013.

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

[39] M. H. Soni, N. Shah, and H. A. Patil, "Time-frequency masking-based speech enhancement using generative adversarial network," in *ICASSP*. IEEE, 2018, pp. 5039–5043.

[40] J. Kim, J. Yoo, S. Chun, A. Kim, and J. Ha, "Multi-domain processing via hybrid denoising networks for speech enhancement," *arXiv preprint arXiv:1812.08914*, 2018.

[41] Jian Yao and Ahmad Al-Dahle, "Coarse-to-fine optimization for speech enhancement," *Proc. Interspeech 2019*, pp. 2743–2747, 2019.