Real Time Speech Enhancement in the Waveform Domain

Alexandre Défossez[1,2,3], *Gabriel Synnaeve*[1], *Yossi Adi*[1]

[1]Facebook AI Research [2]INRIA [3]PSL Research University

defossez,gab,adiyoss@fb.com

arXiv:2006.12847v3 [eess.AS] 6 Sep 2020

Abstract

We present a causal speech enhancement model working on the raw waveform that runs in real-time on a laptop CPU. The proposed model is based on an encoder-decoder architecture with skip-connections. It is optimized on both time and frequency domains, using multiple loss functions. Empirical evidence shows that it is capable of removing various kinds of background noise including stationary and non-stationary noises, as well as room reverb. Additionally, we suggest a set of data augmentation techniques applied directly on the raw waveform which further improve model performance and its generalization abilities. We perform evaluations on several standard benchmarks, both using objective metrics and human judgements. The proposed model matches state-of-the-art performance of both causal and non causal methods while working directly on the raw waveform.

Index Terms: Speech enhancement, speech denoising, neural networks, raw waveform

1. Introduction

Speech enhancement is the task of maximizing the perceptual quality of speech signals, in particular by removing background noise. Most recorded conversational speech signal contains some form of noise that hinders intelligibility, such as street noise, dogs barking, keyboard typing, etc. Thus, speech enhancement is a particularly important task in itself for audio and video calls [1], hearing aids [2], and can also help automatic speech recognition (ASR) systems [3]. For many such applications, a key feature of a speech enhancement system is to run in real time and with as little lag as possible (online), on the communication device, preferably on commodity hardware.

Decades of work in speech enhancement showed feasible solutions which estimated the noise model and used it to recover noise-deducted speech [4, 5]. Although those approaches can generalize well across domains they still have trouble dealing with commons noises such as non-stationary noise or a babble noise which is encountered when a crowd of people are simultaneously talking. The presence of such noise types degrades hearing intelligibility of human speech greatly [6]. Recently, deep neural networks (DNN) based models perform significantly better on non-stationary noise and babble noise while generating higher quality speech in objective and subjective evaluations over traditional methods [7, 8]. Additionally, deep learning based methods have also shown to be superior over traditional methods for the related task of a single-channel source separation [9, 10, 11].

Inspired by these recent advances, we proposed a real-time version of the DEMUCS [11] architecture adapted for speech enhancement. It consists of a causal model, based on convolutions and LSTMs, with a frame size of 40ms, a stride of 16ms, and that runs faster than real-time on a single laptop CPU core. For audio quality purposes, our model goes from waveform to waveform, through hierarchical generation (using U-Net [12] like skip-connections). We optimize the model to directly output the "clean" version of the speech signal while minimizing a regression loss function (L1 loss), complemented with a spectrogram domain loss [13, 14]. Moreover, we proposed a set of simple and effective data augmentation techniques: namely frequency band masking and signal reverberation. Although enforcing a vital real-time constraint on model run-time, our model yields comparable performance to state of the art model by objective and subjective measures.

Although, multiple metrics exist to measure speech enhancement systems these have shown to not correlate well with human judgements [1]. Hence, we report results for both objective metrics as well as human evaluation. Additionally we conduct an ablation study over the loss and augmentation functions to better highlight the contribution of each part. Finally, we analyzed the artifacts of the enhancement process using Word Error Rates (WERs) produced by an Automatic Speech Recognition (ASR) model.

Results suggest that the proposed method is comparable to the current state-of-the-art model across all metrics while working directly on the raw waveform. Moreover, the enhanced samples are found to be beneficial for improving an ASR model under noisy conditions.

2. Model

2.1. Notations and problem settings

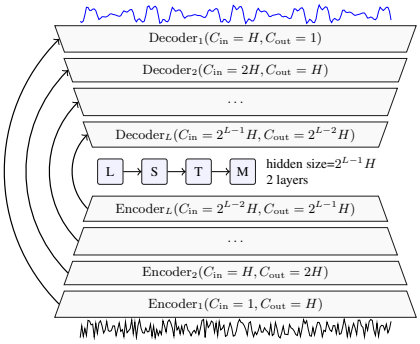
We focus on monaural (single-microphone) speech enhancement that can operate in real-time applications. Specifically, given an audio signal $\boldsymbol{x} \in \mathbb{R}^T$, composed of a clean speech $\boldsymbol{y} \in \mathbb{R}^T$ that is corrupted by an additive background signal $\boldsymbol{n} \in \mathbb{R}^T$ so that $\boldsymbol{x} = \boldsymbol{y} + \boldsymbol{n}$. The length, T, is not a fixed value across samples, since the input utterances can have different durations. Our goal is to find an enhancement function f such that $f(\boldsymbol{x}) \approx \boldsymbol{y}$.

In this study we set f to be the DEMUCS architecture [11], which was initially developed for music source separation, and adapt it to the task of causal speech enhancement, a visual description of the model can be seen in Figure 1a.

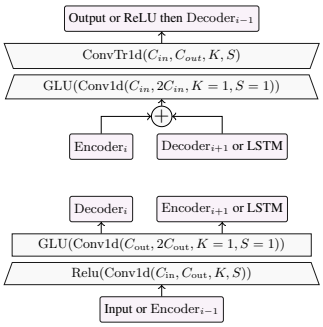
2.2. DEMUCS architecture

DEMUCS consists in a multi-layer convolutional encoder and decoder with U-net [12] skip connections, and a sequence modeling network applied on the encoders' output. It is characterized by its number of layers L, initial number of hidden channels H, layer kernel size K and stride S and resampling factor U. The encoder and decoder layers are numbered from 1 to L (in reverse order for the decoder, so layers at the same scale have the same index). As we focus on monophonic speech enhancement, the input and output of the model has a single channel only.

Formally, the encoder network E gets as input the raw wave form and outputs a latent representation $E(\boldsymbol{x}) = \boldsymbol{z}$. Each layer



(a) *Causal* DEMUCS *with the noisy speech as input on the bottom and the clean speech as output on the top. Arrows represents U-Net skip connections. H controls the number of channels in the model and L its depth.*



(b) *View of each encoder (bottom) and decoder layer (top). Arrows are connections to other parts of the model. C_{in} (resp. C_{out}) is the number of input channels (resp. output), K the kernel size and S the stride.*

Figure 1: *Causal* DEMUCS *architecture on the left, with detailed representation of the encoder and decoder layers on the right. The on the fly resampling of the input/output by a factor of U is not represented.*

consists in a convolution layer with a kernel size of K and stride of S with $2^{i-1}H$ output channels, followed by a ReLU activation, a "1x1" convolution with $2^i H$ output channels and finally a GLU [15] activation that converts back the number of channels to $2^{i-1}H$, see Figure 1b for a visual description.

Next, a sequence modeling R network takes the latent representation $\boldsymbol{z}$ as input and outputs a non-linear transformation of the same size, $R(\boldsymbol{z}) = LSTM(\boldsymbol{z}) + \boldsymbol{z}$, denoted as $\hat{\boldsymbol{z}}$. The LSTM network consists of 2-layers and $2^{L-1}H$ hidden units. For causal prediction, we use an unidirectional LSTM, while for non causal models, we use a bidirectional LSTM, followed by a linear layer to merge the both outputs.

Lastly, a decoder network D, takes as input $\hat{\boldsymbol{z}}$ and outputs an estimation of clean signal $D(\hat{\boldsymbol{z}}) = \hat{\boldsymbol{y}}$. The i-th layer of the decoder takes as input $2^{i-1}H$ channels, and applies a 1x1 convolution with $2^i H$ channels, followed by a GLU activation function that outputs $2^{i-1}H$ channels and finally a transposed convolution with a kernel size of 8, stride of 4, and $2^{i-2}H$ output channels accompanied by a ReLU function. For the last layer the output is a single channel and has no ReLU. A skip connection connects the output of the i-th layer of the encoder and the input of the i-th layer of the decoder, see Figure 1a.

We initialize all model parameters using the scheme proposed by [16]. Finally, we noticed that upsampling the audio by a factor U before feeding it to the encoder improves accuracy. We downsample the output of the model by the same amount. The resampling is done using a sinc interpolation filter [17], as part of the end-to-end training, rather than a pre-processing step.

2.3. Objective

We use the L1 loss over the waveform together with a multi-resolution STFT loss over the spectrogram magnitudes similarly to the one proposed in [13, 14]. Formally, given $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ be the clean signal and the enhanced signal respectively. We define the STFT loss to be the sum of the *spectral convergence (sc)* loss and the *magnitude* loss as follows,

$$
\begin{aligned}
L_{\text{stft}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) &= L_{sc}(\boldsymbol{y}, \hat{\boldsymbol{y}}) + L_{mag}(\boldsymbol{y}, \hat{\boldsymbol{y}}) \\
L_{sc}(\boldsymbol{y}, \hat{\boldsymbol{y}}) &= \frac{|||STFT(\boldsymbol{y})| - |STFT(\hat{\boldsymbol{y}})|||_F}{|||STFT(\boldsymbol{y})|||_F} \\
L_{mag}(\boldsymbol{y}, \hat{\boldsymbol{y}}) &= \frac{1}{T}\| \log |STFT(\boldsymbol{y})| - \log |STFT(\hat{\boldsymbol{y}})|\|_1
\end{aligned}
\tag{1}
$$

where $\| \cdot \|_F$ and $\| \cdot \|_1$ are the Frobenius the L_1 norms respectively. We define the multi-resolution STFT loss to be the sum of all STFT loss functions using different STFT parameters. Overall we wish to minimize the following,

$$
\frac{1}{T}[\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_1 + \sum_{i=1}^{M} L_{\text{stft}}^{(i)}(\boldsymbol{y}, \hat{\boldsymbol{y}})] \tag{2}
$$

where M is the number of STFT losses, and each $L_{\text{stft}}^{(i)}$ applies the STFT loss at different resolution with number of FFT bins $\in \{512, 1024, 2048\}$, hop sizes $\in \{50, 120, 240\}$, and lastly window lengths $\in \{240, 600, 1200\}$.

3. Experiments

We performed several experiments to evaluate the proposed method against several highly competitive models. We report objective and subjective measures on the Valentini et al. [18] and Deep Noise Suppression (DNS) [19] benchmarks. Moreover, we run an ablation study over the augmentation and loss functions. Finally, we assessed the usability of the enhanced samples to improve ASR performance under noisy conditions. Code and samples can be found in the following link: https://github.com/facebookresearch/denoiser.

3.1. Implementation details

Evaluation Methods We evaluate the quality of the enhanced speech using both objective and subjective measures. For the objective measures we use: (i) PESQ: Perceptual evaluation of speech quality, using the wide-band version recommended in ITU-T P.862.2 [24] (from 0.5 to 4.5) (ii) Short-Time Objective Intelligibility (STOI) [25] (from 0 to 100) (iii) CSIG: Mean opinion score (MOS) prediction of the signal distortion attending only to the speech signal [26] (from 1 to 5). (iv) CBAK: MOS prediction of the intrusiveness of background noise [26] (from 1 to 5). (v) COVL: MOS prediction of the overall effect [26] (from 1 to 5).

For the subjective measure, we conducted a MOS study as recommended in ITU-T P.835 [27]. For that, we launched a crowd source evaluation using the CrowdMOS package [28]. We randomly sample 100 utterances and each one was scored

Table 1: *Objective and Subjective measures of the proposed method against SOTA models using the Valentini benchmark [18].*

	PESQ	STOI (%)	pred. CSIG	pred. CBAK	pred. COVL	MOS SIG	MOS BAK	MOS OVL	Causal
Noisy	1.97	91.5	3.35	2.44	2.63	4.08	3.29	3.48	-
SEGAN [7]	2.16	-	3.48	2.94	2.80	-	-	-	No
Wave U-Net [20]	2.40	-	3.52	3.24	2.96	-	-	-	No
SEGAN-D [8]	2.39	-	3.46	3.11	3.50	-	-	-	No
MMSE-GAN [21]	2.53	93	3.80	3.12	3.14	-	-	-	No
MetricGAN [22]	2.86	-	3.99	3.18	3.42	-	-	-	No
DeepMMSE [23]	2.95	94	**4.28**	**3.46**	**3.64**	-	-	-	No
DEMUCS (H=64, S=2 ,U=2)	**3.07**	**95**	**4.31**	3.4	**3.63**	4.02	3.55	3.63	No
Wiener	2.22	93	3.23	2.68	2.67	-	-	-	Yes
DeepMMSE [23]	2.77	93	4.14	**3.32**	**3.46**	4.11	**3.69**	**3.67**	Yes
DEMUCS (H=48,S=4, U=4)	**2.93**	**95**	**4.22**	3.25	**3.52**	4.08	3.59	3.40	Yes
DEMUCS (H=64,S=4, U=4)	2.91	**95**	**4.20**	3.26	**3.51**	4.03	**3.69**	3.39	Yes
DEMUCS (H=64,S=4, U=4) + dry=0.05	2.88	**95**	4.14	3.21	**3.54**	4.10	3.58	**3.72**	Yes
DEMUCS (H=64,S=4, U=4) + dry=0.1	2.81	**95**	4.07	3.10	3.42	**4.18**	3.45	3.60	Yes

by 15 different raters along three axis: level of distortion, intrusiveness of background noise, and overall quality. Averaging results across all annotators and queries gives the final scores.

Training We train the DEMUCS model for 400 epochs on the Valentini [18] dataset and 250 on the DNS [19] dataset. We use the L1 loss between the predicted and ground truth clean speech waveforms, and for the Valentini dataset, also add the STFT loss described in Section 2.3 with a weight of 0.5. We use the Adam optimizer with a step size of $3\mathrm{e}{-4}$, a momentum of $\beta_1 = 0.9$ and a denominator momentum $\beta_2 = 0.999$. For the Valentini dataset, we use the original validation set and keep the best model, for the DNS dataset we train without a validation set and keep the last model. The audio is sampled at 16 kHz.

Model We use three variants of the DEMUCS architecture described in Section 2. For the non causal DEMUCS , we take U=2, S=2, K=8, L=5 and H=64. For the causal DEMUCS , we take U=4, S=4, K=8 and L=5, and either H=48, or H=64. We normalize the input by its standard deviation before feeding it to the model and scale back the output by the same factor. For the evaluation of causal models, we use an online estimate of the standard deviation. With this setup, the causal DEMUCS processes audio has a frame size of 37 ms and a stride of 16 ms.

Data augmentation We always apply a random shift between 0 and S seconds. The *Remix* augmentation shuffles the noises within one batch to form new noisy mixtures. *Band-Mask* is a band-stop filter with a stop band between f_0 and f_1, sampled to remove 20% of the frequencies uniformly in the mel scale. This is equivalent, in the waveform domain, to the SpecAug augmentation [29] used for ASR training. *Revecho*: given an initial gain λ, early delay τ and RT60, it adds to the noisy signal a series of N decaying echos of the clean speech and noise. The n-th echo has a delay of $n\tau + \mathrm{jitter}$ and a gain of $\rho^n\lambda$. N and ρ are chosen so that when the total delay reaches RT60, we have $\rho^N \leq 1\mathrm{e}{-3}$. λ, τ and RT60 are sampled uniformly respectively over $[0, 0.3]$, $[10, 30]$ ms, $[0.3, 1.3]$ sec.

We use the random shift for all datasets, Remix and Banmask for Valentini [18], and Revecho only for DNS [19].

Causal streaming evaluation In order to test our causal model in real conditions we use a specific streaming implementation at test time. Instead of normalizing by the standard deviation of the audio, we use the standard deviation up to the current position (i.e. we use the cumulative standard deviation).

Table 2: *Subjective measures of the proposed method with different treatment of reverb, on the DNS blind test set [19]. Recordings are divided in 3 categories: no reverb, reverb (artificial) and real recordings. We report the OVL MOS. All models are causal. For* DEMUCS *, we take U=4, H=64 and S=4.*

	No Rev.	Reverb	Real Rec.
Noisy	3.1	3.2	2.6
NS-Net [30, 19]	3.2	3.1	2.8
DEMUCS , no reverb	**3.7**	2.7	3.3
DEMUCS , remove reverb	3.6	2.6	3.2
DEMUCS , keep reverb	3.6	3.0	3.2
DEMUCS , keep 10% rev.	3.3	**3.3**	3.1
DEMUCS , keep 10% rev., two sources	3.6	2.8	**3.5**

We keep a small buffer of past input/output to limit the side effect of the sinc resampling filters. For the input upsampling, we also use a 3ms lookahead, which takes the total frame size of the model to 40 ms. When applying the model to a given frame of the signal, the rightmost part of the output is invalid, because future audio is required to compute properly the output of the transposed convolutions. Nonetheless, we noticed that using this invalid part as a padding for the streaming downsampling greatly improves the PESQ. The streaming implementation is pure PyTorch. Due to the overlap between frames, care was taken to cache the output of the different layers as needed.

3.2. Results

Table 1 summarizes the results for Valentini [18] dataset using both causal and non-causal models. Results suggest that DEMUCS matched current SOTA model (DeepMMSE [23]) using both objective and subjective measures, while working directly on the raw waveform and without using extra training data. Additionally, DEMUCS is superior to the other baselines methods, (may they be causal or non-causal), by a significant margin. We also introduce a dry/wet knob, i.e. we output $\mathrm{dry} \cdot x + (1 - \mathrm{dry}) \cdot \hat{y}$, which allows to control the trade-off between noise removal and conservation of the signal. We notice that a small amount of bleeding (5%) improves the overall perceived quality.

We present on Table 2 the overall MOS evaluations on the 3 categories of the DNS [19] blind test set: no reverb (synthetic mixture without reverb), reverb (synthetic mixture, with artificial reverb) and real recordings. We test different strategies for

Table 3: *Ablation study for the causal* DEMUCS *model with* H=64, S=4, U=4 *using the Valentini benchmark [18].*

	PESQ	STOI (%)
Reference	2.91	95
no BandMask (BM)	2.87	95
no BM, no remix	2.86	95
no BM, no remix, no STFT loss	2.68	94
no BM, no remix, no STFT loss , no shift	2.38	93

the reverb-like Revecho augmentation described in Section 3.1. We either ask the model to remove it (dereverberation), keep it, or keep only part of it. Finally, we either add the same reverb to the speech and noise or use different jitters to simulate having two distinct sources. We entered the challenge with the "remove reverb" model, with poor performance on the Reverb category due to dereverberation artifacts[1]. Doing partial dereverberation improves the overall rating, but not for real recordings (which have typically less reverb already). On real recordings, simulating reverb with two sources improves the ratings.

3.3. Ablation

In order to better understand the influence of different components in the proposed model on the overall performance, we conducted an ablation study over the augmentation functions and loss functions. We use the causal DEMUCS version and report PESQ and STOI for each of the methods. Results are presented in Table 3. Results suggest that each of the components contribute to overall performance, with the STFT loss and time shift augmentation producing the biggest increase in performance. Notice, surprisingly the *remix* augmentation function has a minor contribution to the overall performance.

3.4. Real-Time Evaluation

We computed the Real-Time Factor (RTF, e.g. time to enhance a frame divided by the stride) under the streaming setting to better match real-world conditions. We benchmark this implementation on a quad-core Intel i5 CPU (2.0 GHz, up to AVX2 instruction set). The RTF is 1.05 for the H=64 version, while for the H=48 the RTF is 0.6. When restricting execution to a single core, the H=48 model still achieves a RTF of 0.8, making it realistic to use in real conditions, for instance along a video call software. We do not provide RTF results for DeepMMSE [23] since no streaming implementation was provided by the authors, thus making it an unfair comparison.

3.5. The effect on ASR models

Lastly, we evaluated the usability of the enhanced samples to improve ASR performance under noisy conditions. To that end, we synthetically generated noisy data using the LIBRISPEECH dataset [31] together with noises from the test set of the DNS [19] benchmark. We created noisy samples in a controlled setting where we mixed the clean and noise files with SNR levels $\in \{0, 10, 20, 30\}$. For the ASR model we used a Convolutions and Transformer based acoustic model which get states-of-the-art results on LIBRISPEECH, as described in [32]. To get Word Error Rates (WERs) we follow a simple Viterbi (argmax) decoding with neither language model decoding nor beam-search. That way, we can better understand the impact of the enhanced samples on the acoustic model. Results are depicted in Table 4. DEMUCS is able to recover up to 51% of the WER lost to the added noise at SNR 0, and recovering in average 41% of the WER on dev-clean and 31% on dev-other. As the acoustic model was not retrained on denoised data, those results show the direct applicability of speech enhancement to ASR systems as a black-box audio preprocessing step.

[1] MOS for the baselines differ from the challenge as we evaluated on 100 examples per category, and used ITU-T P.835 instead of P.808.

Table 4: *ASR results with a state-of-the-art acoustic model, Word Error Rates without decoding, no language model. Results on the* LIBRISPEECH *validation sets with added noise from the test set of DNS, and enhanced by* DEMUCS .

Viterbi WER on	dev-clean	enhanced	dev-other	enhanced
original (no noise)	2.1	2.2	4.6	4.7
noisy SNR 0	12.0	6.9	21.1	14.7
noisy SNR 10	9.8	6.3	18.4	13.1
noisy SNR 20	5.2	4.0	11.7	9.4
noisy SNR 30	3.3	2.9	7.6	7.2

4. Related Work

Traditionally speech enhancement methods generate either an enhanced version of the magnitude spectrum or produce an estimate of the ideal binary mask (IBM) that is then used to enhance the magnitude spectrum [5, 33].

Over the last years, there has been a growing interest towards DNN based methods for speech enhancement [34, 35, 36, 37, 20, 38, 39, 22, 40, 7, 41, 8, 42, 21, 37]. In [34] a deep feed-forward neural network was used to generate a frequency-domain binary mask using a cost function in the waveform domain. Authors in [43] suggested to use a multi-objective loss function to further improve speech quality. Alternatively authors in [44, 35] use a recursive neural network (RNN) for speech enhancement. In [7] the authors proposed an end-to-end method , namely Speech Enhancement Generative Adversarial Networks (SEGAN) to perform enhancement directly from the raw waveform. The authors in [41, 8, 42, 21] further improve such optimization. In [37] the authors suggest to use a WaveNet [45] model to perform speech denoising by learning a function to map noisy to clean signals.

While considering causal methods, the authors in [46] propose a convolutional recurrent network at the spectral level for real-time speech enhancement, while Xia, Yangyang, et al. [30] suggest to remove the convolutional layers and apply a weighted loss function to further improve results in the real-time setup. Recently, the authors in [23] provide impressive results for both causal and non-causal models using a minimum mean-square error noise power spectral density tracker, which employs a temporal convolutional network (TCN) a priori SNR estimator.

5. Discussion

We have showed how DEMUCS , a state-of-the-art architecture developed for music source separation in the waveform domain, could be turned into a causal speech enhancer, processing audio in real time on consumer level CPU. We tested DEMUCS on the standard Valentini benchmark and achieved state-of-the-art result without using extra training data. We also test our model in real reverberant conditions with the DNS dataset. We empirically demonstrated how augmentation techniques (reverb with two sources, partial dereverberation) can produce a significant improvement in subjective evaluations. Finally, we showed that our model can improve the performance of an ASR model in noisy conditions even without retraining of the model.

6. References

[1] C. K. Reddy *et al.*, "A scalable noisy speech dataset and online subjective test framework," *preprint arXiv:1909.08050*, 2019.

[2] C. K. A. Reddy *et al.*, "An individualized super-gaussian single microphone speech enhancement for hearing aid users with smartphone as an assistive device," *IEEE signal processing letters*, vol. 24, no. 11, pp. 1601–1605, 2017.

[3] C. Zorila, C. Boeddeker, R. Doddipatla, and R. Haeb-Umbach, "An investigation into the effectiveness of enhancement in asr training and test for chime-5 dinner party transcription," *preprint arXiv:1909.12208*, 2019.

[4] J. S. Lim and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech," *Proceedings of the IEEE*, vol. 67, no. 12, pp. 1586–1604, 1979.

[5] 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.

[6] N. Krishnamurthy and J. H. Hansen, "Babble noise: modeling, analysis, and applications," *IEEE transactions on audio, speech, and language processing*, vol. 17, no. 7, pp. 1394–1407, 2009.

[7] S. Pascual, A. Bonafonte, and J. Serra, "Segan: Speech enhancement generative adversarial network," *preprint arXiv:1703.09452*, 2017.

[8] H. Phan *et al.*, "Improving gans for speech enhancement," *preprint arXiv:2001.05532*, 2020.

[9] Y. Luo and N. Mesgarani, "Conv-TASnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[10] E. Nachmani, Y. Adi, and L. Wolf, "Voice separation with an unknown number of multiple speakers," *arXiv:2003.01531*, 2020.

[11] A. Dfossez *et al.*, "Music source separation in the waveform domain," 2019, preprint arXiv:1911.13254.

[12] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015.

[13] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," *preprint arXiv:1910.11480*, 2019.

[14] ——, "Probability density distillation with generative adversarial networks for high-quality parallel waveform generation," *preprint arXiv:1904.04472*, 2019.

[15] Y. N. Dauphin *et al.*, "Language modeling with gated convolutional networks," in *ICML*, 2017.

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

[17] J. Smith and P. Gossett, "A flexible sampling-rate conversion method," in *ICASSP*, vol. 9. IEEE, 1984, pp. 112–115.

[18] C. Valentini-Botinhao, "Noisy speech database for training speech enhancement algorithms and tts models," 2017.

[19] C. K. A. Reddy *et al.*, "The interspeech 2020 deep noise suppression challenge: Datasets, subjective speech quality and testing framework," 2020.

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

[21] 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.

[22] S.-W. Fu *et al.*, "Metricgan: Generative adversarial networks based black-box metric scores optimization for speech enhancement," in *ICML*, 2019.

[23] Q. Zhang *et al.*, "Deepmmse: A deep learning approach to mmse-based noise power spectral density estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020.

[24] I.-T. Recommendation, "Perceptual evaluation of speech quality (pesq): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," *Rec. ITU-T P. 862*, 2001.

[25] C. H. Taal *et al.*, "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, 2011.

[26] Y. Hu and P. C. Loizou, "Evaluation of objective quality measures for speech enhancement," *IEEE Transactions on audio, speech, and language processing*, vol. 16, no. 1, pp. 229–238, 2007.

[27] I. Recommendation, "Subjective test methodology for evaluating speech communication systems that include noise suppression algorithm," *ITU-T recommendation*, p. 835, 2003.

[28] F. Protasio Ribeiro *et al.*, "Crowdmos: An approach for crowdsourcing mean opinion score studies," in *ICASSP*. IEEE, 2011.

[29] D. S. Park *et al.*, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Interspeech*, 2019.

[30] Y. Xia *et al.*, "Weighted speech distortion losses for neural-network-based real-time speech enhancement," *preprint arXiv:2001.10601*, 2020.

[31] V. Panayotov *et al.*, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*. IEEE, 2015, pp. 5206–5210.

[32] G. Synnaeve *et al.*, "End-to-end asr: from supervised to semi-supervised learning with modern architectures," *preprint arXiv:1911.08460*, 2019.

[33] Y. Hu and P. C. Loizou, "Subjective comparison of speech enhancement algorithms," in *ICASSP*, vol. 1. IEEE, 2006, pp. I–I.

[34] Y. Wang and D. Wang, "A deep neural network for time-domain signal reconstruction," in *ICASSP*. IEEE, 2015, pp. 4390–4394.

[35] 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.

[36] Y. Xu *et al.*, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, 2014.

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

[38] A. Nicolson and K. K. Paliwal, "Deep learning for minimum mean-square error approaches to speech enhancement," *Speech Communication*, vol. 111, pp. 44–55, 2019.

[39] F. G. Germain, Q. Chen, and V. Koltun, "Speech denoising with deep feature losses," *preprint arXiv:1806.10522*, 2018.

[40] M. Nikzad, A. Nicolson, Y. Gao, J. Zhou, K. K. Paliwal, and F. Shang, "Deep residual-dense lattice network for speech enhancement," *preprint arXiv:2002.12794*, 2020.

[41] K. Wang, J. Zhang, S. Sun, Y. Wang, F. Xiang, and L. Xie, "Investigating generative adversarial networks based speech dereverberation for robust speech recognition," *arXiv:1803.10132*, 2018.

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

[43] Y. Xu *et al.*, "Multi-objective learning and mask-based post-processing for deep neural network based speech enhancement," *preprint arXiv:1703.07172*, 2017.

[44] F. Weninger *et al.*, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *GlobalSIP*. IEEE, 2014, pp. 577–581.

[45] A. v. d. Oord *et al.*, "Wavenet: A generative model for raw audio," *preprint arXiv:1609.03499*, 2016.

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