INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Speech Enhancement Using Forked Generative Adversarial Networks with Spectral Subtraction

Ju Lin[1], *Sufeng Niu*[2], *Zice Wei*[1], *Xiang Lan*[1], *Adriaan J. van Wijngaarden*[3], *Melissa C. Smith*[1], *Kuang-Ching Wang*[1]

[1] Dept. of Electrical and Computer Engineering, Clemson University, Clemson, SC
[2] LinkedIn Inc., Mountain View, CA
[3] Nokia Bell Labs, Nokia, Murray Hill, NJ

{jul, sniu, zicew, lan3, smithmc, kwang}@clemson.edu,
adriaan.de_lind_van_wijngaarden@nokia-bell-labs.com

Abstract

Speech enhancement techniques that use a generative adversarial network (GAN) can effectively suppress noise while allowing models to be trained end-to-end. However, such techniques directly operate on time-domain waveforms, which are often highly-dimensional and require extensive computation. This paper proposes a novel GAN-based speech enhancement method, referred to as S-ForkGAN, that operates on log-power spectra rather than on time-domain speech waveforms, and uses a forked GAN structure to extract both speech and noise information. By operating on log-power spectra, one can seamlessly include conventional spectral subtraction techniques, and the parameter space typically has a lower dimension. The performance of S-ForkGAN is assessed for automatic speech recognition (ASR) using the TIMIT data set and a wide range of noise conditions. It is shown that S-ForkGAN outperforms existing GAN-based techniques and that it has a lower complexity.

Index Terms: speech enhancement, generative adversarial network, log-power spectra.

1. Introduction

Speech enhancement aims to improve the intelligibility and perceptual quality of degraded speech signals that are affected by noise. Speech enhancement is an important component in Automatic Speech Recognition (ASR) systems and it is used extensively in many applications, including wireless mobile communication systems and hearing aids. Speech enhancement research [1, 2] typically focuses on improving perception metrics, which are "meta" objects in a speech system. It is our aim to use the phone error rate (PER) at the output of the ASR system to quantify the performance of speech enhancement techniques.

Traditional speech enhancement techniques include Wiener filters [3], spectral subtraction [4], statistical learning methods [5] and non-negative matrix factorization [6]. Recent advances in deep learning have contributed to better speech enhancement algorithms. In particular, algorithms that use a deep neural network (DNN) have been shown to achieve excellent speech enhancement performance due to their powerful non-linear function approximation capabilities [7], which are applied to recover noise-corrupted speech [2, 8, 9]. For example, in [9], a DNN is used as a non-linear regression function, and the objective is to minimize the distance between the clean and enhanced speech signals using a mean square error (MSE) loss metric during the training stage. The performance of DNN-based systems is further improved by applying global variance equalization and noise-aware training strategies. In [2], a DNN-based architecture is proposed that uses a multi-objective learning and ensemble (MOLE) framework, and it is shown that one can improve the performance by combining two compact DNNs via boosting. Such DNN-based methods are commonly referred to as feature-mapping methods. Other speech enhancement methods are based on mask-learning [10, 11, 12], where a DNN is used to estimate the ideal ratio mask or the ideal binary mask based on the noisy input features. The mask is used to filter noisy speech signals and recover clean speech signals.

A generative adversarial network (GAN), introduced in [13], uses a DNN to learn synthetic images without any supervision signals. Similarly, adversarial learning can be used to minimize the discrepancy between the distributions of clean features and enhanced features. In [1], a speech enhancement generative adversarial network (SEGAN) was introduced which trains the model directly upon receiving raw audio data in an end-to-end fashion, and it was shown that significant performance gains can be obtained in terms of perceptual speech quality metrics. In [14], SEGAN's auto-encoder was replaced by a standard DNN, and differences in performance were quantified using L_1 and L_2 norms. In [15], a GAN-based de-reverberation front-end for an ASR system was investigated, and the performance was studied for different DNN architectures.

Currently, most state-of-the-art ASR systems use spectral features as input rather than time-domain waveforms. Common techniques to extract spectral features include the use of filter banks. In [16], a GAN-based algorithm was proposed that directly operates on spectral features instead of waveforms, and as such, post-ASR modules do not need to perform extra feature extraction operations.

In this paper, we propose a novel speech enhancement method that operates on the log-power spectra (LPS) of noisy speech waveforms and use a forked GAN structure to extract both speech and noise information. The proposed method, referred to as S-ForkGAN, simultaneously extracts both speech and noise information to aid the reconstruction of the speech signals. This concept is motivated in part by recent work on a time-domain forked GAN framework for speech and noise extraction, which achieved significant performance gains relative to prior GAN-based methods [17]. It will be shown that spectral processing has several significant advantages relative to time-domain processing. The performance of the proposed S-ForkGAN method will be assessed in the scope of automatic speech recognition (ASR) using the TIMIT data set [18] and a wide range of noise conditions.

http://dx.doi.org/10.21437/Interspeech.2019-2954

2. Basic Concepts and Architecture

A general adversarial network (GAN) [13] consists of two neural networks that compete with each other in a two-player min-max game, offering the implicit loss function to train the generator without any supervision. Speech enhancement can be modeled with a GAN-based framework, where a discriminator eventually cannot distinguish the distributions of the original and the enhanced speech signals. The speech enhancement generative adversarial network (SEGAN) model [1] can be expressed as a combination of two component networks: 1) a generator network $\mathbf{G}$ that tries to learn a distribution $P_g(\boldsymbol{x})$ of a noisy speech signal $\tilde{\boldsymbol{x}}$ and a prior input noise variable $p_z(\boldsymbol{z})$. The goal is to generate the true data distribution $P_t(\boldsymbol{x})$ to fool the discriminator; 2) The discriminator network $\mathbf{D}$ serves as a binary classifier which aims to determine the probability that a given sample comes from the real data set rather than from $\mathbf{G}$. A least-squares GAN (LS-GAN) is used to stabilize training and improve the quality of the generated samples in the generator in the original SEGAN concept. A speech enhancement module is to output a signal $\hat{\boldsymbol{x}}$ that is as close as possible to the clean speech signal $\tilde{\boldsymbol{x}}$. The objective functions are given by

$$\begin{aligned}\mathcal{L}_{\mathbf{D}} = &\frac{1}{2}\mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}\sim p_{\text{data}}(\boldsymbol{x},\tilde{\boldsymbol{x}})}[(D(\tilde{\boldsymbol{x}},\boldsymbol{x})-1)^2] \\ &+\frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z}),\tilde{\boldsymbol{x}}\sim p_{\text{data}}}[D(G(\boldsymbol{z},\boldsymbol{x}),\boldsymbol{x})^2]\end{aligned} \tag{1}$$

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

In this paper, we propose a framework that utilizes noise information to aid speech enhancement. Several previous studies also showed that noise information is beneficial when using a deep neural network for speech recognition (noise-aware training) [19] and for speech enhancement [20]. The basic concept of the proposed method is to extract spectral features from the noisy speech signal $\boldsymbol{x}$ and to use two forked GAN networks to simultaneously extract both speech and noise information to aid the reconstruction of the speech signals. We refer to this framework as a spectral forked GAN-based (S-ForkGAN) framework.

Spectral processing has three major advantages: 1) speech enhancement can seamlessly interface with a post-ASR system, since state-of-the-art ASRs widely use acoustic features in the frequency domain; 2) the input dimensions of the raw time-domain noisy speech signals are typically much higher than for the spectral features; and 3) by performing speech enhancement in the frequency domain, one reinforces ASR robustness.

2.1. Architecture

The proposed S-ForkGAN architecture uses a generator and a discriminator network as shown in Fig. 1. It takes a noisy speech signal $\boldsymbol{x}$ as input and extracts its LPS features using a Fast Fourier transform (FFT). The LPS features are normalized globally over all training speech samples into a zero mean and unit variance. The generator consists of two forked GAN decoders that simultaneously estimate speech and noise patterns conditioned on input latent variables. The resulting GAN networks are trained with adversarial learning and L_1 regression loss. The noise information that is learned by the extra decoder can be integrated into the GAN-based framework via spectral subtraction. Spectral subtraction is as such used to recover the speech signal by subtracting an estimate of the average noise spectrum from the noisy signal spectrum. Thus, a spectral domain speech enhancement model for an input LPS is achieved by introducing two auxiliary loss functions that aid in decoupling the noise from the source signal: a margin-based loss component which pushes the speech and noise signals apart, and a spectral subtraction loss component that combines conventional signal processing spectral subtraction with the neural network predictions.

The denoising operation is now considered in more detail. Let the encoding and decoding operation be denoted by $\boldsymbol{\Phi}(\cdot)$ and $\boldsymbol{\Psi}(\cdot)$, respectively. It should be noted that $\boldsymbol{c} \in \mathbb{R}^d$ is a latent representation of the noisy speech signal $\boldsymbol{x} \in \mathbb{R}^n$, and $\boldsymbol{s} \in \mathbb{R}^m$ represents the LPS features, where $d < m \ll n$.

Algorithm 1 Denoising Algorithm

```
Input: Noisy speech signal data set $\mathcal{X}$
Output: Enhanced signal $\hat{\boldsymbol{s}}_u, \hat{\boldsymbol{v}}_u, \forall u \in \mathcal{X}$
for each $u \in \mathcal{X}$ do
    Step 1. $\boldsymbol{s}_u \leftarrow$ LPS $(\boldsymbol{x}_u)$;
    Step 2. $\boldsymbol{c}_u \leftarrow \boldsymbol{\Phi}(\boldsymbol{s}_u)$;
    Step 3. $(\boldsymbol{c}_u^s, \boldsymbol{c}_u^v) \leftarrow \{\text{CONCAT}(\boldsymbol{w}_s\boldsymbol{c}_u, \boldsymbol{z})|\boldsymbol{z} \sim \mathcal{N}(0, I)\}$,
            $\{\text{CONCAT}(\boldsymbol{w}_v\boldsymbol{c}_u, \boldsymbol{z})|\boldsymbol{z} \sim \mathcal{N}(0, I)\}$;
    Step 4. $(\hat{\boldsymbol{s}}, \hat{\boldsymbol{v}}) \leftarrow \boldsymbol{\Psi}_s(\boldsymbol{c}_u^s), \boldsymbol{\Psi}_v(\boldsymbol{c}_u^v)$ ;
end for
```

The computational procedure in the generator is outlined in Algorithm 1. In Step 1, the LPS features are extracted as the input of encoder using an FFT. Then, in Step 2, the encoder function $\Phi(\cdot)$ extracts a latent vector $\boldsymbol{c}$ from the received noisy speech signal $\tilde{\boldsymbol{s}}$. In Step 3, the latent vector $\boldsymbol{c}$ is decoupled using a linear transformation to extract two latent features $\boldsymbol{c}_u^s$ and $\boldsymbol{c}_u^v$ for the decoder, where $\boldsymbol{c}_u^s$ encodes the clean speech information and $\boldsymbol{c}_u^v$ encodes the noise information. Each input of the decoder splices an encoder latent representation $\boldsymbol{c}$ with a random vector $\boldsymbol{z}$ sampled from a normal distribution $\mathcal{N}(0, I)$. In Step 4, the speech decoder $\boldsymbol{\Psi}_s(\cdot)$ and the noise decoder $\boldsymbol{\Psi}_v(\cdot)$ aim to generate the speech signal and the additive noise signal, respectively. Both decoders have the same architecture. The generator $\mathbf{G}$ is an end-to-end module that performs convolutional operations, and both decoder layers have the inverse structure of the encoder, with the same configurations. Note that each layer input is concatenated with skip connections from the encoder. The resulting outputs are the clean speech prediction $\hat{\boldsymbol{s}} \in \mathbb{R}^m$, and the noise prediction $\hat{\boldsymbol{v}} \in \mathbb{R}^m$. In the training phase, the objective is to minimize the difference between the enhanced signal pair $(\hat{\boldsymbol{s}}, \hat{\boldsymbol{v}})$ and the ground truth signal pair $(\tilde{\boldsymbol{s}}, \tilde{\boldsymbol{v}})$ by optimizing the encoder and decoder functions.

2.2. Loss Functions

Additional training objectives that are based on the characteristics of the proposed architecture are as follows:

Margin-based Loss. For S-ForkGAN to operate well, it is important to maximize the distance between the speech signal and the noise signal. A max-margin-based loss function is proposed to regularize the loss of the model, and to ensure that the distance between the embedding of the clean speech signal and the noise speech signal is larger than some pre-defined margin. As such, the loss function can make the generated speech and noise as dissimilar as possible. A suitable function is to use the Euclidean distance between the two embeddings in Step 3 of Algorithm 1, which is defined as $\mathcal{D} = \frac{1}{d}\Sigma||\boldsymbol{c}_v - \boldsymbol{c}_s||_2$. Note that a normalization factor on the embeddings is added to stabilize

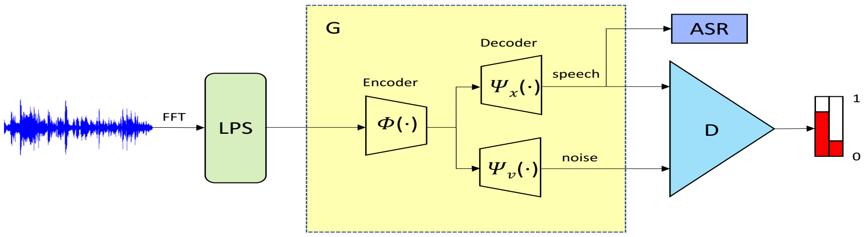

Figure 1: *Proposed S-ForkGAN architecture which consists of forked GAN networks for simultaneous speech enhancement and noise identification.*

training, where $\boldsymbol{c}_v \leftarrow \boldsymbol{c}_v/||\boldsymbol{c}_v||_2$, $\boldsymbol{c}_s \leftarrow \boldsymbol{c}_s/||\boldsymbol{c}_s||_2$. The loss for each pair of clean speech embedding and noise embedding is thus:

$$\mathcal{L}_{\text{margin}} = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \max(0, \Delta - \mathcal{D}(\boldsymbol{s})), \tag{3}$$

where Δ denotes the margin hyper-parameter.

Spectral Subtraction Loss. Spectral subtraction is one of the traditional algorithms for enhancing a single speech channel. Since the noisy signal $x_t = \tilde{x}_t + v_t$ is the addition of the desired signal value $\tilde{x}_t$ and the noise value v_t at time t, the standard spectral subtraction is defined in the frequency domain as:

$$\tilde{X}(j\omega) = X(j\omega) - V(j\omega) \tag{4}$$

where $X(j\omega)$, $\tilde{X}(j\omega)$ and $V(j\omega)$ are Fourier transforms of x_t, $\tilde{x}_t$, v_t, respectively. As shown in (4), the accuracy of spectral subtraction heavily depends on accurate noise spectrum estimation. Unlike conventional noise spectrum estimation, S-ForkGAN uses a neural network as a function estimator to evaluate the noise signal. Since the S-ForkGAN directly operates on the spectral domain features, a standard spectral subtraction loss term is incorporated into the proposed training objectives. The noise reduction term $\mathcal{L}_{\text{reduction}}$ can be derived by subtracting the noise prediction term from the generator, i.e.,

$$\mathcal{L}_{\text{reduction}} = \mathbb{E}_{\boldsymbol{s}\sim p_{\text{data}}}||\boldsymbol{s} - \hat{\boldsymbol{v}} - \tilde{\boldsymbol{s}}||_1 \tag{5}$$

The generator loss $\mathcal{L}_G$ is the weighted sum of the margin-based and spectral-subtraction loss factors, which can be written as

$$\begin{aligned}\mathcal{L}_G = & \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z}),\boldsymbol{s}\sim p_{\text{data}}(\boldsymbol{s})}[(D(G(\boldsymbol{z},\boldsymbol{s}),\boldsymbol{s})-1)^2] \\ & + \lambda\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z}),\tilde{\boldsymbol{s}},\boldsymbol{s}\sim p_{data}}||G(\boldsymbol{z},\boldsymbol{s}) - \tilde{\boldsymbol{s}}||_1 \\ & + \alpha\mathcal{L}_{\text{margin}} + \beta\mathcal{L}_{\text{reduction}}\end{aligned} \tag{6}$$

where α and β denote coefficients that control the strength of each auxiliary loss function.

3. Experimental Setup

The performance of the proposed S-ForkGAN method is evaluated using extensive simulations.

Data Setup. The data set for the experiments is generated from two sources: the DARPA TIMIT corpus [18] is used for clean speech references, whereas the noise is extracted from the NOISEX-92 corpus [21]. The TIMIT corpus includes eight major American-English dialects recorded from 630 speakers, each reading ten phonetically rich sentences, and this corpus is partitioned into test and training subsets. The training set includes 4620 sentences; 192 sentences are selected for the testing set. The NOISEX-92 corpus includes 15 different noise types, ranging from machinery noise to machine gun noise. For the training set, a randomly-selected noise sound from NOISEX-92 is added to every silenced-added segment with a signal-to-noise ratio (SNR) of -5 dB, 0 dB, 5 dB and 10 dB. The test set was generated by adding noise from the NOISEX-92 corpus, using the same settings as the training set.

S-ForkGAN Setup. The proposed technique and architecture can be summarized as follows: the model is trained for 20 epochs with the RMSprop [22] method. It operates directly on spectral domain features, LPS, instead of on raw audio, and it aims to learn a mapping from the LPS feature input to the LPS feature output. The input and target LPS features are normalized by using zero mean and unit variance, respectively. The input feature contains a context window of 11 frames (±5), thus it is a 2827-to-257 mapping relation. Further experiments with ForkGAN use exactly the same settings [17]. In S-ForkGAN, the shared encoder consists of 11 one-dimensional strided convolutional layers of filter width-31 and stride length 2. The number of filters per convolutional layer increases so that the depth increases as the width gets shorter. The resulting dimensions per layer in terms of the number of samples times the number of feature maps is 2827×1, 1414×16, 707×32, 354×32, 177×64, 89×64, 45×128, 23×128, 12×256, 6×256, 3×512 and 2×1024. A flattening operation is used for converting a 2×1024 vector to two length-2048 vectors via fully-connected layers. After that two encoded latent variables are obtained, which are used for speech and noise respectively. Also, the margin-based loss is calculated by these two vectors. The two encoded latent vectors are concatenated with two noise samples, which are from an a prior 2×1024-dimensional normal distribution $\mathcal{N}(0, I)$. The concatenated vectors are the input of each decoder. The network parameters of the decoder are symmetric to the encoder. The discriminator also utilizes a one-dimensional convolution similar to the generator's encoding stage and is adapted to behave as a classification network.

Baseline Setup. Several GAN-based method with different enhancement networks are used as baseline systems, e.g., a DNN and long short-term memory (LSTM). Note that GAN-DNN and GAN-LSTM were originally used for speech dereverberation; they are adjusted here for speech enhancement. The setup is similar to [15]. If the generator is an auto-encoder, the suffix AE is used.

GAN-DNN. The feed-forward DNN includes four hidden layers, each of which contains 1024 ReLU neurons. The input feature consists of a stacked 11-frame LPS feature. The mode is trained for 20 epochs using the learning rate 0.001 with a mini-batch size of length-8. Batch normalization is performed for this model.

GAN-LSTM. Instead of using a plain-vanilla LSTM, an LSTM with recurrent projection layer (LSTMP) [23] was adopted here. The LSTM includes four LSTMP layers followed by a linear output layer. Each LSTMP layer has 760 memory cells and 257 projection units and the input to the LSTM is a single acoustic frame with 257-dimensional LPS features. The learning rate was set to $3.0 \cdot 10^{-4}$ and the model was trained with eight full-length utterances parallel processing.

GAN-AE. The setup for GAN-AE is similar to S-ForkGAN. The only difference is that the generator is an auto-encoder architecture with the same configuration as the proposed method, using one decoder to generate clean speech.

ASR Setup. A Deep Neural Network-Hidden Markov Model (DNN-HMM) acoustic model is developed to evaluate the enhanced LPS features. A Gaussian Mixture Model-Hidden Markov Model(GMM-HMM) is first trained to obtain senones (tied tri-phone states) and the corresponding aligned frames for DNN training. The input feature vectors that are used to train the GMM-HMM contain 257-dimensional LPS and their first and second derivatives. The splices of nine frames (four on each side of the current frame) are projected down to 40-dimensional vectors by linear discriminant analysis (LDA), together with maximum likelihood linear transform (MLLT), and then used to train the GMM-HMM using maximum likelihood estimation.

The LPS features take a context size of 11 frames (± 5), as the input of the DNN. The DNN topology consists of six hidden layers, and each layer contains $1,024$ nodes. Since TIMIT is a small corpus, the DNN acoustic model was first initialized with stacked restricted Boltzmann machines (RBMs) that were pre-trained in a greedy layered fashion [24]. After pre-training, all weights and biases were discriminator-trained by optimizing the cross-entropy between the target probability, corresponding to context-dependent HMM states, and the actual soft-max output with the Back-Propagation (BP) algorithm [25]. The weights are refined using sequence-discriminative training, state-level minimum Bayes risk (sMBR).

4. Performance Results

Using the experimental set-up presented in the previous section, the acoustic model was trained using clean data, and it was determined that the phone error rate (PER) on the TIMIT test set equals 18.0 %. The PER values were determined for several existing GAN-based speech enhancement approaches. It can be observed from Table 1 that all methods reduce the noise and improve the ASR performance. It is shown that GAN-LSTM achieves better results than GAN-DNN for all SNR values. For example, a GAN-LSTM reduces the PER from 32.6 % to 29.4 %. This indicates that LSTMs ability to model long-term contextual information is essential for speech enhancement.

Table 1: *Phone error rate (as a percentage) for S-ForkGAN and prior methods; the best results for each SNR value are boldfaced.*

SNR	-5 dB	0 dB	5 dB	10 dB
LPS w/o SE	87.2	81.3	70.1	56.0
GAN-DNN (LPS) [15]	54.6	48.0	39.4	32.6
GAN-LSTM (LPS) [15]	51.7	42.5	34.8	29.4
GAN-AE (raw audio) [1]	48.6	44.9	35.2	34.4
GAN-AE (LPS) [1]	45.7	38.3	34.1	32.4
ForkGAN [17]	47.1	37.2	33.5	30.4
S-ForkGAN	45.1	37.8	30.5	26.8

The measurements show that GAN-AE with LPS inputs can further improve the performance, especially for high SNR. In contrast to GAN-LSTM, the PER drops from 51.7 % to 45.7 % and from 42.5 % to 38.3 % for an SNR of -5 dB and 0 dB, respectively. This means that the convolution layers in the auto-encoder can also provide additional useful information for speech enhancement. The proposed S-ForkGAN method with LPS achieved the best performance, which shows the effectiveness of the additional decoder and two auxiliary loss functions.

Given that S-ForkGAN and GAN-AE are auto-encoder-based methods, the performance is determined for different input features. The raw audio input is set to be the same as for the original SEGAN [1], where each chunk of waveform was extracted with a sliding window of approximately one second of speech (16,384 samples) every 500 ms. A high-frequency pre-emphasis filter coefficient of 0.95 was applied to all input samples during the training and test stages. From Table 1, one can see that both GAN-AE and S-ForkGAN with LPS features outperform systems with raw audio as input. The results show that directly operating on LPS is more helpful for the ASR tasks. Note that S-ForkGAN outperforms GAN-AE with respect to these two features.

To visualize the performance of the GAN-based methods, a single sentence was selected and mixed with destroyer noise at 0 dB. The spectrograms for each method are shown in Fig. 2. The S-ForkGAN and GAN-AE methods are clearly better than the GAN-DNN and GAN-LSTM methods.

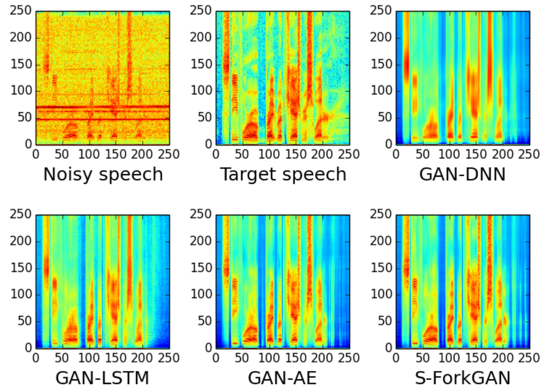

Figure 2: *Spectrograms for GAN-DNN, GAN-LSTM, GAN-AE and the proposed S-ForkGAN method for a sample input mixture with destroyer noise and an SNR of 0 dB.*

5. Conclusions

In this paper, a novel GAN-based speech enhancement module is proposed, which consists of a dual GAN decoder to capture both speech and noise patterns. Speech signal extraction is achieved using a spectral subtraction loss term and a margin-based loss term to further improve the quality of the enhanced speech signals. The experiments show that the proposed S-ForkGAN method outperforms well-known GAN-based speech enhancement techniques, including GAN-DNN, GAN-LSTM and GAN-AE (SEGAN). Additional experiments show that when raw waveforms and LPS were used as inputs, the performance of LPS-based systems is better than waveform-based systems and that fewer computations are required.

6. Acknowledgements

This work is supported by the US Army Medical Research and Materiel Command under Contract No. W81XWH-17-C-0238.

7. References

[1] S. Pascual, A. Bonafonte, and J. Serrà, "SEGAN: Speech enhancement generative adversarial network," *ArXiV:1703.09452*, 2017.

[2] Q. Wang, J. Du, L.-R. Dai, and C.-H. Lee, "A multiobjective learning and ensembling approach to high-performance speech enhancement with compact neural network architectures," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 26, no. 7, pp. 1181–1193, Jul. 2018.

[3] P. Scalart and J. Vieira Filho, "Speech enhancement based on a priori signal to noise estimation," in *Proc. IEEE Int'l Conf. on Acoustics, Speech, and Signal Proc.*, Atlanta, GA, May 1996, pp. 629–632.

[4] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 27, no. 2, pp. 113–120, Apr. 1979.

[5] P. C. Loizou, *Speech Enhancement: Theory and Practice*. CRC Press, 2007.

[6] N. Mohammadiha, P. Smaragdis, and A. Leijon, "Supervised and unsupervised speech enhancement using nonnegative matrix factorization," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 21, no. 10, pp. 2140–2151, Oct. 2013.

[7] D. Yu and L. Deng, *Automatic Speech Recognition: A Deep Learning Approach*. London, United Kingdom: Springer, 2016.

[8] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Process. Lett.*, vol. 21, no. 1, pp. 65–68, Jan. 2014.

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

[10] A. Narayanan and D. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *Proc. IEEE Int'l Conf. on Acoustics, Speech and Signal Proc.*, Vancouver, BC, May 2013, pp. 7092–7096.

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

[12] 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. Int'l Conf. on Latent Variable Analysis and Signal Separation*, Liberec, Czech Republic, Aug. 2015, pp. 91–99.

[13] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial sets," in *Proc. Conf. on Neural Inform. Proc. Sys.*, Montréal, QC, Dec. 2014.

[14] A. Pandey and D. Wang, "On adversarial training and loss functions for speech enhancement," in *Proc. IEEE Int'l Conf. on Acoustics, Speech and Signal Processing*, Calgary, AB, Apr. 2018, pp. 5414–5418.

[15] 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 preprint arXiv:1803.10132*, 2018.

[16] C. Donahue, B. Li, and R. Prabhavalkar, "Exploring speech enhancement with generative adversarial networks for robust speech recognition," in *Proc. IEEE Int'l Conf. on Acoustics, Speech and Signal Proc.*, Calgary, AB, Apr. 2018, pp. 5024–5028.

[17] J. Lin, S. Niu, X. Lan, L. Sun, A. J. van Wijngaarden, M. C. Smith, and K.-C. Wang, "ForkGAN: Forked GAN-based decoders for speech enhancement," submitted for possible presentation at the *2019 Conference on Neural Information Processing Systems*.

[18] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic phonetic continuous speech corpus CD-ROM - NIST speech disc 1-1-1," 1993.

[19] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. IEEE Int'l Conf. on Acoustics, Speech and Signal Proc.*, Vancouver, BC, May 2013, pp. 7398–7402.

[20] A. Kimar and D. Florencio, "Speech enhancement in multiple-noise conditions using deep neural networks," *arXiv preprint arXiv:1605.02427*, 2016.

[21] A. Varga and H. J. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Communication*, vol. 12, no. 3, pp. 247–251, Jul. 1993.

[22] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop, coursera: Neural networks for machine learning," University of Toronto, Techn. Rep., 2012.

[23] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. Annual Conf. Int'l Speech Commun. Assoc.*, Singapore, Singapore, Sep. 2014, pp. 338–342.

[24] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, p. 533, 1986.