Neurocomputing 438 (2021) 63–71

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Generative adversarial networks for single channel separation of convolutive mixed speech signals

Yang Li, Wei-Tao Zhang [*], Shun-Tian Lou

School of Electronic Engineering, Xidian University, Xi'an 710071, China

ARTICLE INFO

Article history:
Received 26 August 2020
Revised 23 November 2020
Accepted 8 January 2021
Available online 18 January 2021
Communicaed by Zidong Wang

Keywords:
Speech separation
Convolutive mixing
Generative adversarial network

ABSTRACT

The suppression of interference for speech recognition is of great significance in noisy situation, especially in single channel receiving mode, the suppression of interference is much more difficult. In this paper, we propose a generative adversarial network (GAN) based method for single channel dereverberation and speech separation. Different from the existing methods, our method considers the influence of strong reverberation on the observed signals. The proposed network involves two parts: reverberation suppression and target speech enhancement. Firstly, we use an improved CyclyGAN to compensate the multi-path effect on both target speech and interference. Secondly, we propose a differentialGAN to extract both target speech and interference while the interference enhancement network can indirectly improve the performance of target speech enhancement network. We use the real and imaginary parts of the complex spectrum as the feature vector, which avoids the phase mismatch during signal recovery. Simulation results show that our method is superior to its competitors in terms of multiple metrics in severe reverberation environment.

1. Introduction

At present, in a relatively quiet and simple acoustic environment, the accuracy of speech recognition has reached the application level. However, in the more complex environment, such as the presence of interference signal, the accuracy of speech recognition will be significantly reduced. Separating target speech from mixed speech can effectively improve the performance of subsequent speech recognition. However, the existing separation methods have certain limitations, such as being sensitive to reverberation, requiring multi-channel data and the phase mismatch of the separated speech. Therefore, the application of speech separation is still far from being satisfactory in the real world.

For multi-channel convolutive mixed speech separation, the current research focuses on CBSS [1–3]. CBSS is based on the statistical independence of different source signals. The mixed signals are usually transformed into frequency domain by Shot-Time Fourier Transform (STFT), and the separation matrix is solved at each frequency bin. However, CBSS usually cannot work in a single channel receiving mode. In addition, since the speech signal is non-stationary, the window length of STFT cannot exceed the maximum short-time stationary time. When the reverberation time exceeds the maximum short-time stationary time, the mixture will no longer satisfy the signal model assumed by CBSS method. In this case, the performance of CBSS will degrade significantly.

In the field of single channel speech separation, the current research focuses on speech enhancement (SE). Classical SE methods include spectral subtraction [4] and Wiener filtering [5]. These traditional signal processing methods usually assume that the signal is stationary or slow-varying. However, as a non-stationary signal, speech signal is usually difficult to satisfy the above assumption. To separate the target signal from the mixture, in recent years, the supervised learning method has achieved great success. The early supervised learning model is nonnegative matrix factorization [6]. With the application of deep learning, Deep Neural Network (DNN) [7] is also used for SE. On this basis, Deep Stack Network (DSN) [8,9] and Deep Recurrent Neural Network (DRNN) [10] use the temporal correlation of speech signals to improve the separation performance. RNN has defects in processing long term memory while Long Short-Term Memory (LSTM) [11,12] has got some improvement and further improved the separation performance. In addition, Convolutional Neural Network (CNN) [13,14] uses the two-dimensional time–frequency spectrum to explore the temporal and spatial structure characteristics of the signal, so as to obtain a good separation performance. However, the above SE methods mainly consider the linear instantaneous mixing mode. In real world, reverberation exists commonly which will lead to a degradation of enhancement performance.

[*] Corresponding author.
E-mail address: zhwt-work@foxmail.com (W.-T. Zhang).

https://doi.org/10.1016/j.neucom.2021.01.052
0925-2312/© 2021 Elsevier B.V. All rights reserved.

Now GAN [15,16] has achieved great success in the field of image processing. Conditional Generative Adversarial Net (CGAN) [17,18] is used to realize image to image translation task. CycleGAN [19] realizes the translation from unpaired images to images. GAN has also found some applications in the field of SE [20–22]. These methods usually use the magnitude spectrum as the signal feature. This means that the network does not have ability to restore the phase of the target speech. Instead, the phase of the target signal is roughly estimated by directly using the phase of the mixture. This phase mismatch usually has no significant effect on SE in high Signal to Noise Ratio (SNR) scene. However, when the SNR is lower, the performance will be degraded. On the other hand, these methods do not consider the multi-path transmission of the speech and interference. The existence of reverberation will further deteriorate the phase mismatch and performance.

In this work, we propose a multi-GAN system for convolutive mixed speech separation. Our major contributions can be summarized as.

- Firstly, we divide the speech separation problem into two sub tasks: reverberation suppression and speech enhancement. We improve CycleGAN by adding conditional supervised loss function, thus achieving the reverberation suppression of both target speech and interference. On the other hand, we propose a differential network for SE. In addition to training a network for enhancing target speech, we also train a network for enhancing interference simultaneously. The interference enhancement network indirectly affects the target speech enhancement network by means of the differential network structure. This training method further improves the separation performance of the mixture.
- Secondly, considering the influence of reverberation on signal, different from existing GAN methods which take the magnitude spectrum of the signal as the training sample, we retain the phase information of the signal's time–frequency spectrum. This improvement can avoid the phase mismatch when the signal is recovered to time-domain.
- Finally, the simulation results and analysis reveal that, our proposed method can deal with the separation under both fixed and random layouts of source signals. Moreover, it has some advantages over the existing methods, especially under the condition of long reverberation time.

The sequel of this paper is organized as follows. In the Section 2, we give a convolutive mixing model, as well as an overview of GAN. Section 3 presents the details of the proposed method, including signal preprocessing, network designing and training. Section 4 presents the experimental settings and analysis of the results. Finally, we conclude the paper in Section 5.

2. Background

2.1. Signal model

Consider that there are target speech $s(k)$ and interference $v(k)$ with one microphone in space. In the real environment, in addition to the direct path, there are also superimposed reflection and refraction paths in the transmission of sound signal. Relative to the direct-path, other paths will produce signal attenuation and time-delay. Therefore, the signal received by the microphone can be expressed as

$$x(k) = \sum_{\tau=0}^{T} a_s(\tau)s(k-\tau) + \sum_{\tau=0}^{T} a_v(\tau)v(k-\tau) \tag{1}$$

where $x(k)$ is the received mixed signal, the FIR filter $a_s(\tau)$ and $a_v(\tau)$ represent the impulse response of all paths from $s(k)$ and $v(k)$ to the microphone. The reverberation time T determines the length of convolutive mixing filter. As non-stationary signal, it is considered that the speech signal is stable in a short time, and the maximum short-time stationary time (approximately10-30 ms) is recorded as $T_{\max}$. STFT is used to convert Eq. (1) into frequency-domain. The length of the window function needs to be shorter than $T_{\max}$. When $T < T_{\max}$, for frequency bin ω, the convolutive mixing model (1) can be written in frequency-domain as

$$X(\omega,\lambda) = A_s(\omega)S(\omega,\lambda) + A_v(\omega)V(\omega,\lambda) \tag{2}$$

where $X(\omega,\lambda), S(\omega,\lambda)$ and $V(\omega,\lambda)$ are the FFTs of $x(k), s(k)$ and $v(k)$ in frame λ. $A_s(\omega)$ and $A_v(\omega)$ are the FFTs of $a_s(\tau)$ and $a_s(\tau)$. If the reverberation time is further increased to make $T > T_{\max}$, Eq. (1) will be rewritten in frequency-domain as

$$X(\omega,\lambda) = \sum_{p=0}^{P-1} A_s(\omega,p)S(\omega,\lambda-p) + \sum_{p=0}^{P-1} A_v(\omega,p)V(\omega,\lambda-p). \tag{3}$$

where the maximum reverberation time determines the size of P. It can be seen that Eq. (2) is a special case of Eq. (3) when $P = 1$. In this paper, we will discuss the speech separation method based on the model of Eq. (3). We assume that the layout of $s(k)$, $v(k)$ and microphone is fixed in one experiment. We set different reverberation time and evaluate their impact on separation performance.

2.2. Generative adversarial network

GAN [15] originated from the theory of two-players zero-sum game, which consists of a generator network G and a discriminator network D. Generator G maps input sample **a** from data space **A** to sample **b** from data space **B**, and packages the output $G(\mathbf{a})$ into a realistic sample to confuse discriminator D. D is a binary classifier, which is used to classify whether the sample matches the probability distribution of real data.

The training of GAN consists of two parts. First, optimize D to enable it to classify real data and generated data. Then, optimize G to make the generated data closer to the real data, so that D cannot classify whether the generated data is real or fake. This adversarial learning process with value function $V(D,G)$ can be expressed as

$$\begin{aligned}\min_G \max_D V(D,G) = {} & E_{\mathbf{b}\sim P(\mathbf{B})}[\log(D(\mathbf{b}))] \\ & + E_{\mathbf{a}\sim P(\mathbf{A})}[\log(1-D(G(\mathbf{a})))]\end{aligned} \tag{4}$$

where **b** is the real data sample from distribution $P(\mathbf{B})$ and $G(\mathbf{a})$ is the generated data sample from distribution $P(\mathbf{A})$.

3. Related work

In this section, we first give the user-defined signal vector space, and then give the network structure based on this definition. Finally, we give a detailed objective function for the network optimization.

3.1. Proposed framework

By Eq. (3), with N frequency bins, the frame vectors of the target speech and interference are denoted as $\mathbf{s}_\lambda = [S(\omega_1,\lambda), S(\omega_2,\lambda),\ldots,S(\omega_N,\lambda)]^T$ and $\mathbf{v}_\lambda = [V(\omega_1,\lambda), V(\omega_2,\lambda),\ldots,V(\omega_N,\lambda)]^T$. Then define define speech vector space and interference vector space as $\mathbf{S}: \{\mathbf{s}_\lambda, \lambda = 1,2,\ldots\}$ and $\mathbf{V}: \{\mathbf{v}_\lambda, \lambda = 1,2,\ldots\}$. From space **S** and space **V**, according to Eq. (3), we can generate spaces **X** and **Y** expressed as

$$\begin{aligned}&\mathbf{X}:\{\mathbf{a}_s*\mathbf{s}_\lambda+\mathbf{a}_v*\mathbf{v}_\lambda|\mathbf{s}_\lambda\in\mathbf{S},\mathbf{v}_\lambda\in\mathbf{V}\}\\&\mathbf{Y}:\{\mathbf{s}_\lambda+\mathbf{v}_\lambda|\mathbf{s}_\lambda\in\mathbf{S},\mathbf{v}_\lambda\in\mathbf{V}\}\end{aligned}\tag{5}$$

where $\mathbf{a}_s=[A_s(\omega_1),A_s(\omega_2),\ldots,A_s(\omega_N)]^T$, $\mathbf{a}_v=[A_v(\omega_1),A_v(\omega_2),\ldots,A_v(\omega_N)]^T$, $*$ represents convolution product. of corresponding elements. **X** represents the convolutive mixed signal space. **Y** represents the additive space of corresponding signal and interference. Space **Y** is equivalent to the suppression of the convolutive mixing filter $A_s(\omega)$ and $A_v(\omega)$ from space **X**.

We divide the interference suppression of convolutive mixed speech signal into two parts: reverberation suppression and speech enhancement. Reverberation suppression corresponds to the mapping from space **X** to **Y**, while speech enhancement corresponds to the mapping from space **Y** to **S**. These two mappings can be implemented by two CGANs respectively, but the result is usually not optimal.

Traditional GAN is unidirectional, and CycleGAN is a circular loop composed of two mirror GANs. The cyclic consistent loss makes the two GANs constrained with each other during optimization. Another advantage of CycleGAN is that the data can be unpaired. However, for speech signals, it is easier to obtain paired data than images. We can also use paired data from Space **X** and **Y** for training. Compared with the unidirectional GAN, the performance will still be improved. The following simulation results also verify this point.

The existing speech enhancement methods usually only use the information of the target speech. We propose a method to construct a differential cross loop using the extra information of interference signal. The direct mapping from space **Y** to **S** does not guarantee a completely clean target speech. We will get a differential signal by subtracting it with space **Y** data. It can be expected that most of the components in this differential signal will be the corresponding space **V** data, while only a small part belongs to space **S.** This is equivalent to a preliminary interference enhancement of data **Y**. At this time, by using this differential data, we try to recover the interference data and we can certainly get a better result than that of the space **Y** data. In the same way, if we do the difference calculation on the mapping data from space **Y** to **V**, the result is equivalent to the preliminary target speech enhancement. Next, we use the same enhancement network to enhance the differential signal again, and we can get the estimated speech closer to the target speech.

Fig. 1 shows the framework of the proposed method in this paper. We use four generators ($G_{XY}, G_{YX}, G_{YS}$and G_{YV}) and four discriminators (D_Y, D_X, D_V and D_S) in this system. G_{XY} represents the mapping from space **X** to space **Y**, while G_{YX} on the contrary. G_{YS} represents the signal separation from space **Y** to **S** and G_{YV} represents the interference separation from space **Y** to **V**.

The mapping between space **X** and space **Y** is bidirectional. We use a CycleGAN to transform the convolutive mixing model to the additive mixing model. Compared with the direct mapping from

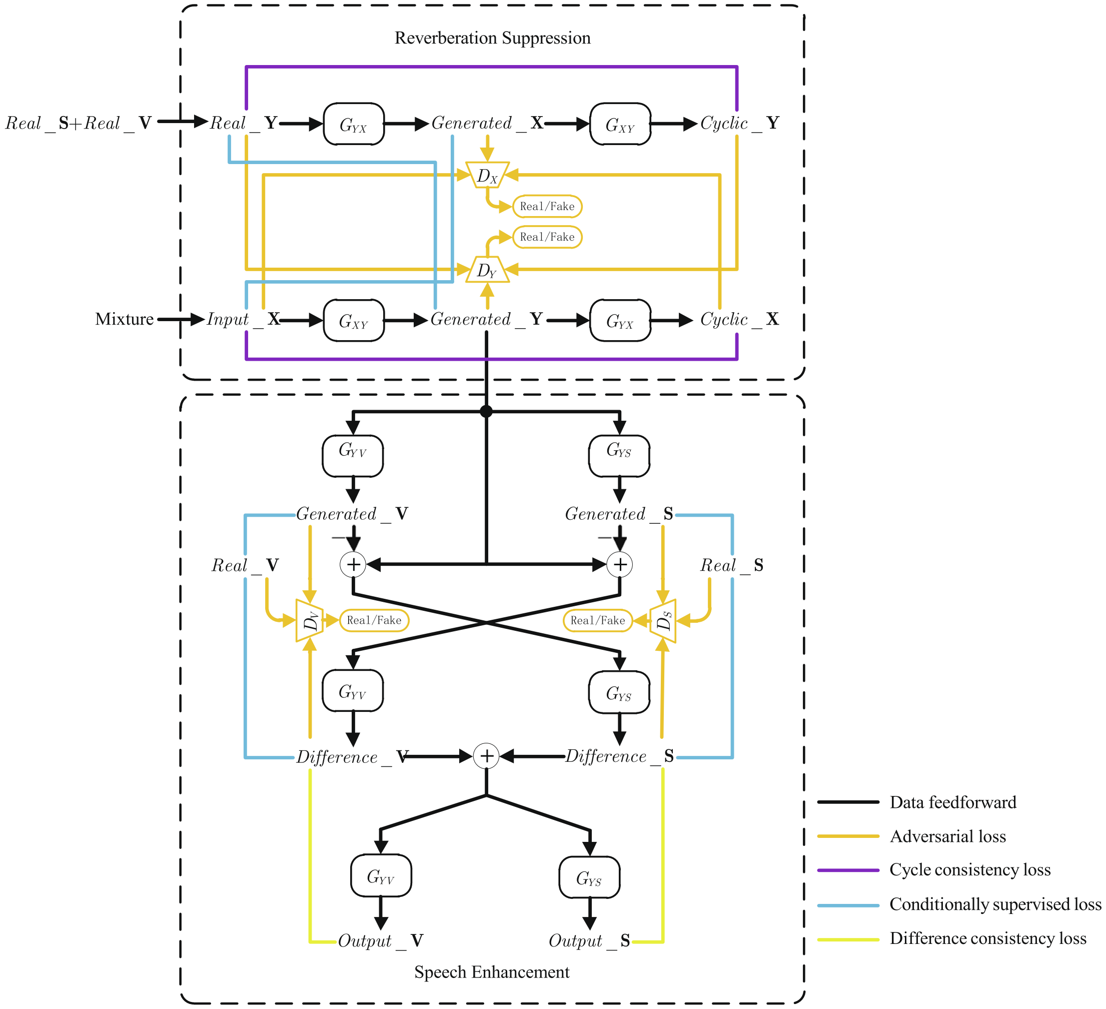

Fig. 1. Framework of proposed method.

space **Y** to space **S**, our method takes advantage of the information of interference space **V**. Two paired networks G_{YS} and G_{YV} are established to separate the target speech and interference from space **Y** respectively. Then differential the output of the G_{YS} and G_{YV} with **Y**, and cross-input the difference results to further separate the signal again.

3.2. Training sample

The training data consists of both real and imaginary parts of the frame vector. In space **X**, for the λth frame vector $\mathbf{x}_\lambda = \left[X(\omega_1,\lambda), X(\omega_2,\lambda), \ldots, X(\omega_{nfft},\lambda)\right]^T$, the training sample vector can be expressed as

$$\widehat{\mathbf{x}}_\lambda = \left[real(\mathbf{x}_\lambda^T), imag(\mathbf{x}_\lambda^T)\right]^T \tag{6}$$

where $real(\mathbf{x}_\lambda^T)$ and $imag(\mathbf{x}_\lambda^T)$ represent the real and imaginary parts of $\mathbf{x}_\lambda^T$. Similarly, we can get $\widehat{y}_\lambda = \left[real(\mathbf{y}_\lambda^T), imag(\mathbf{y}_\lambda^T)\right]^T$, $\mathbf{s}_\lambda = \left[real(\mathbf{s}_\lambda^T), imag(\mathbf{s}_\lambda^T)\right]^T$ and $\widehat{v}_\lambda = \left[real(\mathbf{v}_\lambda^T), imag(\mathbf{v}_\lambda^T)\right]^T$ where $\mathbf{y}_\lambda = \left[Y(\omega_1,\lambda), Y(\omega_2,\lambda), \ldots, Y(\omega_{nfft},\lambda)\right]^T$, $\mathbf{s}_\lambda = [S(\omega_1,\lambda), S(\omega_2,\lambda), \ldots, S(\omega_{nfft},\lambda)]^T$, $\mathbf{v}_\lambda = [V(\omega_1,\lambda), V(\omega_2,\lambda), \ldots, V(\omega_{nfft},\lambda)]^T$.

Considering that the human auditory system is not sensitive to the phase information of speech. The existing SE methods use magnitude spectrum as training samples. However, the presence of reverberation will further deteriorate the phase mismatch. Our improvement of training samples indirectly retains the phase information of the signal, and the network can learn the mapping of phase information between different vector spaces

3.3. Objective function

The training of network consists of two steps. We first train the mapping network between space **X** and space **Y** in step 1. Then, in step 2, we fix the weight of G_{XY}, and then train the mapping network from space **X** to space **S** and space **V**.

Step 1: We use a CycleGAN to realize the conversion between convolutive mixing and additive mixing. The mapping from space **X** to the generated **Y** sample $G_{XY}(\widehat{\mathbf{x}}_\lambda)$ through G_{XY}, and the discriminator D_Y plays a discriminative role in fake sample $G_{XY}(\widehat{\mathbf{x}}_\lambda)$ and the real sample $\widehat{\mathbf{y}}_\lambda$. Then, optimize G_{XY} by confusing D_Y to make the generated sample closer to the real sample. The value function of adversarial loss can be expressed as

$$\begin{aligned} V_{Gen1}(D_Y, G_{XY}) = {} & E_{\widehat{\mathbf{y}}_\lambda \sim P(\mathbf{Y})}[\log(D_Y(\widehat{\mathbf{y}}_\lambda))] \\ & + E_{\widehat{\mathbf{x}}_\lambda \sim P(\mathbf{X})}\left[\log(1 - D_Y(G_{XY}(\widehat{\mathbf{x}}_\lambda))\right] \end{aligned} \tag{7}$$

In addition, in order to control the outputs of G_{XY} mapping to expected results, we introduce $\widehat{\mathbf{y}}_\lambda$ as conditional variable. And we can get the conditional supervised loss function as

$$L_{XY} = \mu_1 E_{\widehat{\mathbf{x}}_\lambda, \widehat{\mathbf{y}}_\lambda \sim P(\mathbf{X},\mathbf{Y})}\left[\|G_{XY}(\widehat{\mathbf{x}}_\lambda) - \widehat{\mathbf{y}}_\lambda\|_{\mathrm{F}}\right] \tag{8}$$

where μ_1 balances the effect of conditional supervised loss on network training. Then G_{YX} maps $G_{XY}(\widehat{\mathbf{x}}_\lambda)$ back to cyclic **X** sample, and the corresponding value function of adversarial loss can be expressed as

$$\begin{aligned} V_{Cyc1}(D_X, G_{YX}) = {} & E_{\widehat{\mathbf{x}}_\lambda \sim P(\mathbf{X})}\left[log(D_X(\widehat{\mathbf{x}}_\lambda))\right] \\ & + E_{\widehat{\mathbf{x}}_\lambda \sim P(\mathbf{X})}\left[\log(1 - D_X(G_{YX}(G_{XY}(\widehat{\mathbf{x}}_\lambda)))\right] \end{aligned} \tag{9}$$

To reduce the discrepancy between cyclic sample $G_{YX}(G_{XY}(\widehat{\mathbf{x}}_\lambda))$ and real sample $\widehat{\mathbf{x}}_\lambda$, a cycle consistency loss function is introduced as

$$L_X = \eta_1 E_{\widehat{\mathbf{x}}_\lambda \sim P(\mathbf{X})}\left[\|G_{YX}(G_{XY}(\widehat{\mathbf{x}}_\lambda)) - \widehat{\mathbf{x}}_\lambda\|_{\mathrm{F}}\right] \tag{10}$$

where η_1 balances the effect of value function cycle consistency loss on network training.

Similarly, through the mapping process from real sample $\widehat{\mathbf{y}}_\lambda$ to generated **X** sample $G_{YX}(\widehat{\mathbf{y}}_\lambda)$ to cyclic sample $G_{XY}(G_{YX}(\widehat{\mathbf{y}}_\lambda))$, we can get the following loss functions

$$\begin{aligned} V_{Gen2}(D_X, G_{YX}) = {} & E_{\widehat{\mathbf{x}}_\lambda \sim P(\mathbf{X})}[\log(D_X(\widehat{\mathbf{x}}_\lambda))] \\ & + E_{\widehat{\mathbf{y}}_\lambda \sim P(\mathbf{Y})}\left[\log(1 - D_X(G_{YX}(\widehat{\mathbf{y}}_\lambda))\right] \end{aligned} \tag{11}$$

$$L_{YX} = \mu_1 E_{\widehat{\mathbf{x}}_\lambda, \widehat{\mathbf{y}}_\lambda \sim P(\mathbf{X},\mathbf{Y})}\left[\|G_{YX}(\widehat{\mathbf{y}}_\lambda) - \widehat{\mathbf{x}}_\lambda\|_{\mathrm{F}}\right] \tag{12}$$

$$\begin{aligned} V_{Cyc2}(D_Y, G_{XY}) = {} & E_{\widehat{\mathbf{y}}_\lambda \sim P(\mathbf{Y})}[\log(D_Y(\widehat{\mathbf{y}}_\lambda))] \\ & + E_{\widehat{\mathbf{y}}_\lambda \sim P(\mathbf{Y})}\left[\log(1 - D_Y(G_{XY}(G_{YX}(\widehat{\mathbf{y}}_\lambda)))\right] \end{aligned} \tag{13}$$

$$L_Y = \eta_1 E_{\widehat{\mathbf{y}}_\lambda \sim P(\mathbf{Y})}\left[\|G_{XY}(G_{YX}(\widehat{\mathbf{y}}_\lambda)) - \widehat{\mathbf{y}}_\lambda\|_{\mathrm{F}}\right] \tag{14}$$

From the discussion above, we can get the training process in step 1 as

$$\begin{aligned} & \min_{G_{XY},G_{YX}} \max_{D_Y,D_X} \{V_{Gen1} + V_{Cyc1} + V_{Gen2} + V_{Cyc2} + L_{XY} + L_{YX} + L_X \\ & \quad + L_Y\} \end{aligned} \tag{15}$$

Step 2: We take a differential GAN method to get the mapping from space **Y** to space **S**. Sample $G_{XY}(\widehat{\mathbf{x}}_\lambda)$ in Step 1 generates the space **V** sample $G_{YV}(G_{XY}(\widehat{\mathbf{x}}_\lambda))$ and space **S** sample $G_{YS}(G_{XY}(\widehat{\mathbf{x}}_\lambda))$ through G_{YV} and G_{YS}. Discriminator D_Y plays a discriminative role in fake samples $G_{YV}(G_{XY}(\widehat{\mathbf{x}}_\lambda))$ and real sample $\widehat{\mathbf{v}}_\lambda$. While discriminator D_S plays a discriminative role in fake samples $G_{YS}(G_{XY}(\widehat{\mathbf{x}}_\lambda))$ and real sample $\widehat{\mathbf{s}}_\lambda$. Then, G_{YV} and G_{YS} are optimized by confusing D_Y and D_S to make the generated samples closer to the real samples. The respective value functions of adversarial loss can be expressed as

$$\begin{aligned} V_{Gen3}(D_V, G_{YV}) = {} & E_{\widehat{\mathbf{v}}_\lambda \sim P(\mathbf{V})}[\log(D_V(\widehat{\mathbf{v}}_\lambda))] \\ & + E_{\widehat{\mathbf{x}}_\lambda \sim P(\mathbf{X})}\left[\log(1 - D_V(G_{YV}(G_{XY}(\widehat{\mathbf{x}}_\lambda)))\right] \end{aligned} \tag{16}$$

$$\begin{aligned} V_{Gen4}(D_S, G_{YS}) = {} & E_{\widehat{\mathbf{s}}_\lambda \sim P(\mathbf{S})}[\log(D_S(\widehat{\mathbf{s}}_\lambda))] \\ & + E_{\widehat{\mathbf{x}}_\lambda \sim P(\mathbf{X})}\left[\log(1 - D_S(G_{YS}(G_{XY}(\widehat{\mathbf{x}}_\lambda)))\right] \end{aligned} \tag{17}$$

We also introduce $\widehat{\mathbf{s}}_\lambda$ and $\widehat{\mathbf{v}}_\lambda$ as conditional variables, and the corresponding conditional supervised loss function can be expressed as

$$L_{GenYV} = \mu_2 E_{\widehat{\mathbf{x}}_\lambda, \widehat{\mathbf{v}}_\lambda \sim P(\mathbf{X},\mathbf{V})}\left[\|G_{YV}(G_{XY}(\widehat{\mathbf{x}}_\lambda)) - \widehat{\mathbf{v}}_\lambda\|_{\mathrm{F}}\right] \tag{18}$$

$$L_{GenYS} = \mu_2 E_{\widehat{\mathbf{x}}_\lambda, \widehat{\mathbf{s}}_\lambda \sim P(\mathbf{X},\mathbf{S})}\left[\|G_{YS}(G_{XY}(\widehat{\mathbf{x}}_\lambda)) - \widehat{\mathbf{s}}_\lambda\|_{\mathrm{F}}\right] \tag{19}$$

where μ_2 balances the effect of conditional supervised loss function on network training. Then $G_{YV}(G_{XY}(\widehat{\mathbf{x}}_\lambda))$ and $G_{YS}(G_{XY}(\widehat{\mathbf{x}}_\lambda))$ are differentiated with $G_{XY}(\widehat{\mathbf{x}}_\lambda)$, and we obtain $G_{XY}(\widehat{\mathbf{x}}_\lambda) - G_{YV}(G_{XY}(\widehat{\mathbf{x}}_\lambda))$ and $G_{XY}(\widehat{\mathbf{x}}_\lambda) - G_{YS}(G_{XY}(\widehat{\mathbf{x}}_\lambda))$. Cross-inputting the results to network G_{YS} and G_{YV}, this process generates the space **S** sample $G_{YS}(G_{XY}(\widehat{\mathbf{x}}_\lambda) - G_{YV}(G_{XY}(\widehat{\mathbf{x}}_\lambda)))$ and space **V** sample $G_{YV}(G_{XY}(\widehat{\mathbf{x}}_\lambda) - G_{YS}(G_{XY}(\widehat{\mathbf{x}}_\lambda)))$. We can find that the discriminator D_Y plays a discriminative role in fake sample $G_{YV}(G_{XY}(\widehat{\mathbf{x}}_\lambda) - G_{YS}(G_{XY}(\widehat{\mathbf{x}}_\lambda)))$ and the real sample $\widehat{\mathbf{v}}_\lambda$. The discriminator D_S plays a discriminative role in fake sample $G_{YS}(G_{XY}(\widehat{\mathbf{x}}_\lambda) - G_{YV}(G_{XY}(\widehat{\mathbf{x}}_\lambda)))$ and the real sample $\widehat{\mathbf{s}}_\lambda$. Then, G_{YV} and G_{YS} are optimized by confusing D_Y and D_S to make the generated samples closer to the real samples. The respective value functions can be expressed as

$$V_{Dif1}(D_V, G_{YV}) = E_{\widehat{\mathbf{v}}_\lambda \sim P(\mathbf{V})}[\log(D_V(\widehat{\mathbf{v}}_\lambda))] + E_{\widehat{\mathbf{x}}_\lambda \sim P(\mathbf{X})}[\log(1 - D_V(G_{YV}(G_{XY}(\widehat{\mathbf{x}}_\lambda) - G_{YS}(G_{XY}(\widehat{\mathbf{x}}_\lambda)))))] \tag{20}$$

$$V_{Dif2}(D_S, G_{YS}) = E_{\widehat{\mathbf{s}}_\lambda \sim P(\mathbf{S})}[\log(D_S(\widehat{\mathbf{s}}_\lambda))] + E_{\widehat{\mathbf{x}}_\lambda \sim P(\mathbf{X})}[\log(1 - D_S(G_{YS}(G_{XY}(\widehat{\mathbf{x}}_\lambda) - G_{YV}(G_{XY}(\widehat{\mathbf{x}}_\lambda)))))] \tag{21}$$

We introduce $\widehat{\mathbf{s}}_\lambda$ and $\widehat{\mathbf{v}}_\lambda$ as conditional variables, and the corresponding conditional supervised loss function can be expressed as

$$L_{DiffYV} = \mu_2 E_{\widehat{\mathbf{x}}_\lambda, \widehat{\mathbf{v}}_\lambda \sim P(\mathbf{X},\mathbf{V})}\left[\|G_{YV}(G_{XY}(\widehat{\mathbf{x}}_\lambda) - G_{YS}(G_{XY}(\widehat{\mathbf{x}}_\lambda)) - \widehat{\mathbf{v}}_\lambda\|_{\mathrm{F}}\right] \tag{22}$$

$$L_{DiffYS} = \mu_2 E_{\widehat{\mathbf{x}}_\lambda, \widehat{\mathbf{s}}_\lambda \sim P(\mathbf{X},\mathbf{S})}\left[\|G_{YS}(G_{XY}(\widehat{\mathbf{x}}_\lambda) - G_{YV}(G_{XY}(\widehat{\mathbf{x}}_\lambda)) - \widehat{\mathbf{s}}_\lambda\|_{\mathrm{F}}\right] \tag{23}$$

where μ_2 balances the effect of conditional supervised loss function on network training.

We sum the differential network output data as

$$Sum_{Diff} = G_{YV}(G_{XY}(\widehat{\mathbf{x}}_\lambda) - G_{YS}(G_{XY}(\widehat{\mathbf{x}}_\lambda)) + G_{YS}(G_{XY}(\widehat{\mathbf{x}}_\lambda) - G_{YV}(G_{XY}(\widehat{\mathbf{x}}_\lambda)) \tag{24}$$

which can be regarded as an update of $G_{XY}(\widehat{\mathbf{x}}_\lambda)$. We map the updated $G_{XY}(\widehat{\mathbf{x}}_\lambda)$ to spaces **S** and **V** again. G_{YV} and G_{YS} are further optimized by defining the following differential consistency loss function

$$L_V = \eta_2 E_{\widehat{\mathbf{x}}_\lambda \sim P(\mathbf{X})}\left[\|G_{YV}(Sum_{Diff}) - G_{YV}(G_{XY}(\widehat{\mathbf{x}}_\lambda) - G_{YS}(G_{XY}(\widehat{\mathbf{x}}_\lambda))\|_{\mathrm{F}}\right] \tag{25}$$

$$L_S = \eta_2 E_{\widehat{\mathbf{x}}_\lambda \sim P(\mathbf{X})}\left[\|G_{YS}(Sum_{Diff}) - G_{YS}(G_{XY}(\widehat{\mathbf{x}}_\lambda) - G_{YV}(G_{XY}(\widehat{\mathbf{x}}_\lambda))\|_{\mathrm{F}}\right] \tag{26}$$

where η_2 balances the effect of difference consistency loss function on network training.

In conclusion, the overall optimization process of Step 2 can be express as

$$\min_{G_{YV},G_{YS}} \max_{D_V,D_S} \{V_{Gen3} + V_{Diff1} + V_{Gen4} + V_{Diff2} + L_{GenXY} + L_{GenYX} + L_{DiffYV} + L_{DiffYS} + L_V + L_S\} \tag{27}$$

4. Simulation results

4.1. Dataset and setting

The proposed framework is evaluated using the TIMIT and NoiseX-92 nonspeech dataset. The TIMIT dataset contains 6300 speech data (630 people, each of whom record 10 different sentences) with a sample rate of 16 kHz. The NoiseX-92 nonspeech dataset contains 100 background interference data. In order to keep same sampling rate of these two databases, all the data are down sampled to 8 kHz.

We randomly select 100 people from the TIMIT dataset with each person randomly selected one sentence and 20 kinds of interference in the Noisex-92 nonspeech dataset. Let us adjust the amplitude of the interference so that the signal to interference ratio ($SIR = 10log_{10}(\sum_{k=0}^{T}|s(k)|^2/\sum_{k=0}^{T}|v(k)|^2)$) changes from -5 dB to 5 dB.

We randomly select another 100 people from the TIMIT dataset with each person selected two sentences. One sentence contains exactly the same content, we use for speech recognition test. The other sentence randomly selected from different speech content used for calculating separation performance. Each sentence is randomly matched with one of the remaining Noisex-92 nonspeech datasets as the interference. Speech mixing is similar to that of the training data.

We divide the simulation into two cases: the fixed and randomly changed layout of target speech and interference. Fig. 2 shows the fixed layout of signals and microphones. We place the target speech and interference on different radius circles centered on the microphone 1. We set the microphones omni-directional and set the target speech at two positions. In Layout 1, we set the angle between target source speech and interference as acute angle, while in layout 2, that is obtuse. We test the separation performance under these two layouts respectively. Furthermore, our method can adapt to the small range changes of the coordinates of speech and interference. Under this condition, we test all methods again. In this test, Microphone 1 is still selected as the center of the circle. The location of speech and interference varies from 1.0 m to 1.5 m in radius and 0 to 360 degrees in normal angle.

The convolutive mixing filters are obtained by Roomsim [23]. We can get the impulse responses under different layouts and different reverberation time. Fig. 3 shows an impulse response in one experiment. STFT is used to transform data from time-domain to frequency-domain. The length of window function is 16 ms (128 sample points) with 8ms overlap.

In order to prove the validity of proposed Cycle + Differential GAN (CDGAN) method, we compare it with the existing GAN based SE method (SEGAN) [21], Cycle + CGAN method (one CycleGAN

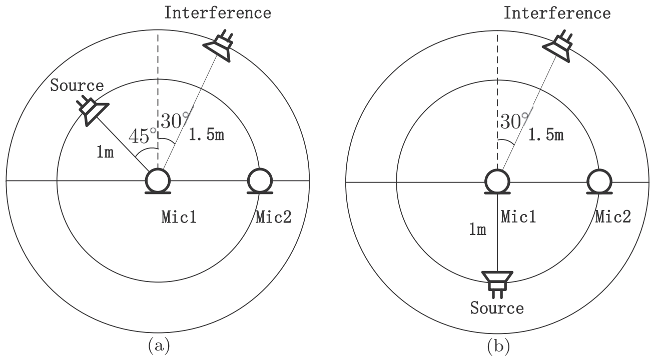

Fig. 2. Layout of target speech and microphones by Roomsim: (a) Layout 1; (b) Layout 2. Microphone 1 used for GAN methods, Microphone 1 + Microphone 2 used for BSS method.

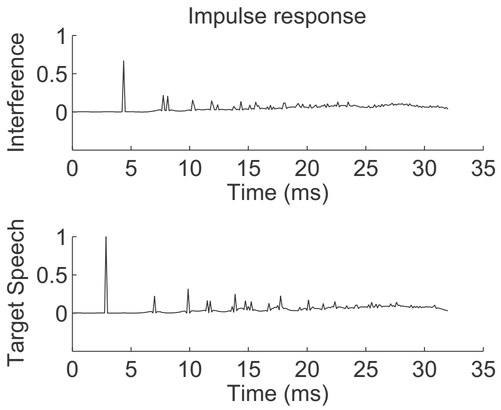

Fig. 3. Impulse responses with 32 ms reverberation time while the target speech is at the position of Source 1.

Table 1
ASIR performance for different methods with several reverberation time under fixed layout. The average SIR of the mixture is −1.02 dB.

	ASIR (dB)							
Reverberation	8 ms		16 ms		32 ms		64 ms	
Layout	1	2	1	2	1	2	1	2
SEGAN	3.88	3.76	3.13	3.15	2.89	2.97	2.66	2.59
Cycle + CGAN	4.06	4.15	3.47	3.43	3.28	3.38	3.03	3.05
CBSS	5.33	5.45	3.03	2.91	−0.91	−1.05	−1.04	−1.12
CDGAN	**5.15**	**5.28**	**4.96**	**4.89**	**4.62**	**4.52**	**4.21**	**4.13**

mapping between space **X** and space **Y**, and another CGAN mapping from space **Y** to space **S**), and traditional CBSS method [1]. We calculate the average performance of each algorithm under two kinds of location distribution. The separation performance can be calculated by the average signal to interference ratio (ASIR), which can be expressed as

$$ASIR = 10\log_{10}\left[\frac{1}{K}\sum_{k=1}^{K}\frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\|\bar{S}_k(\omega_i,\lambda_j)\|_F^2}{\sum_{i=1}^{N}\sum_{j=1}^{M}\|\bar{\mathbf{Y}}_k(\omega_i,\lambda_j)-\bar{\mathbf{S}}_k(\omega_i,\lambda_j)\|_F^2}\right] \tag{28}$$

where $\bar{\mathbf{S}}_k = \mathbf{S}_k/\|\mathbf{S}_k\|_F$ and $\bar{\mathbf{Y}}_k = \mathbf{Y}_k/\|\mathbf{Y}_k\|_F$ are the normalized kth target and estimated speech. N is the maximum number of frequency bins, M is the STFT frame number of one mixed speech, and K is the total number of test mixed speech. In this paper, we setup $N = 256$, $M = 150$ and $K = 100$.

4.2. Result and discussion

Tables 1 and 2 show the ASIR performance of the proposed CDGAN method and its competitors SEGAN and CycleGAN + CGAN with 8 ms, 16 ms, 32 ms and 64 ms reverberation time. It can be seen that the three kinds of GAN structure methods do have a ability to separate the convolutive mixed speech. SEGAN operates on the amplitude spectrum, which makes the noise and reverberation filter bring inaccurate phase information when the signal is recovered, and therefore has the worst overall separation performance. The performance of Cycle + CGAN method is slightly better than that of SEGAN method. This is because SEGAN directly uses speech enhancement network to separate convolutive mixtures while the presence of reverberation make the separation performance degrade. CycleGAN + CGAN uses CycleGAN to suppress the reverberation and also uses the training sample, such as Eq. (5), to avoid the inaccurate phase information in signal recovery. Therefore, compared with SEGAN, CycleGAN + CGAN improves the separation performance. CDGAN and CycleGAN + CGAN have the same dereverberation network. We can find that the ASIR of our CDGAN method is better than that of CycleGAN + CGAN, which is because the proposed DifferentialGAN further optimizes the mapping from space **Y** to space **S**. The results in Tables 3 and 4 also show that our method get higher scores of PESQ and STOI than that of the two competitors. Figs. 4 and 6 show the target speeches and mixed signals under fixed and random layout respectively. Figure 5 and 7 show the corresponding test results. The reverberation time is 32 ms and the dif-

Table 2
ASIR performance for different methods with several reverberation time under random layout. The average SIR of the mixture is −0.92 dB.

	ASIR (dB)			
Reverberation	8 ms	16 ms	32 ms	64 ms
SEGAN	2.97	2.62	2.11	1.98
Cycle + CGAN	3.25	2.81	2.52	2.32
CBSS	5.31	2.92	−0.88	−1.15
CDGAN	**3.93**	**3.47**	**3.37**	**3.09**

Table 4
PESQ and STOI performance for different methods with several reverberation time under random layout.

Reverberation	8 ms	16 ms	32 ms	64 ms
	PESQ			
Mixture	1.69	1.77	1.76	1.69
SEGAN	2.47	2.41	2.30	2.15
Cycle + CGAN	2.55	2.51	2.39	2.22
CBSS	3.11	2.67	1.88	1.85
CDGAN	**2.63**	**2.57**	**2.49**	**2.29**
	STOI			
Mixture	67.39	66.97	67.26	67.29
SEGAN	71.93	70.25	68.91	67.94
Cycle + CGAN	76.66	74.81	73.54	71.95
CBSS	90.06	81.92	63.84	58.88
CDGAN	**80.93**	**78.66**	**77.26**	**76.73**

Table 3
PESQ and STOI performance for different methods with several reverberation time under fixed layout.

Reverberation	8 ms		16 ms		32 ms		64 ms	
Layout	1	2	1	2	1	2	1	2
	PESQ							
Mixture	1.72		1.62		1.71		1.77	
SEGAN	2.58	2.56	2.44	2.45	2.37	2.33	2.26	2.22
Cycle + CGAN	2.76	2.75	2.67	2.64	2.48	2.52	2.43	2.45
CBSS	3.13	3.10	2.63	2.61	1.96	1.91	1.82	1.84
CDGAN	**2.85**	**2.79**	**2.76**	**2.74**	**2.62**	**2.65**	**2.51**	**2.53**
	STOI							
Mixture	68.77		68.62		69.31		68.93	
SEGAN	73.77	73.86	71.89	71.99	70.87	70.79	69.87	69.78
Cycle + CGAN	81.44	81.43	80.59	80.53	78.54	78.43	76.09	76.14
CBSS	90.34	89.97	82.03	82.21	63.90	63.65	58.77	58.53
CDGAN	**84.43**	**84.31**	**82.68**	**82.69**	**81.14**	**81.22**	**80.17**	**80.11**

ferent kinds of interference are randomly selected in the test set. It can be seen that, the target signal is recovered successfully by the three different methods. However, in the recovery of signal details, our method is better than the other two competitors.

Since the CBSS method does not need the layout information of the source signal, the performance of CBSS in Tables 1 and 2 is roughly the same. When the reverberation time is 8 ms, the performance of our method is close to that of CBSS method under fixed layout, but lower than that of CBSS under random layout. However, as the reverberation time increases to 16 ms, CBSS method has begun to show significant performance degradation. Finally, when the reverberation time further increases to 32 ms, which exceeds the maximum short-time stationary time $T_{\max}$, the CBSS method fail to keep consistance. However, compared with CBSS method,

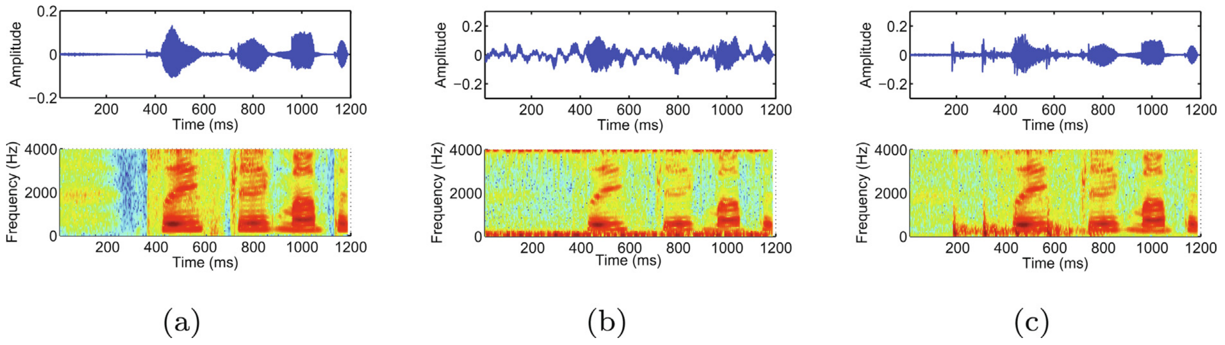

Fig. 4. Time-domain waveforms and corresponding time–frequency spectrograms under fixed layout. (a) Target speech; (b) Mixture with interference 1, SIR = −1.03 dB; (c) Mixture with interference 2, SIR = 2.80 dB.

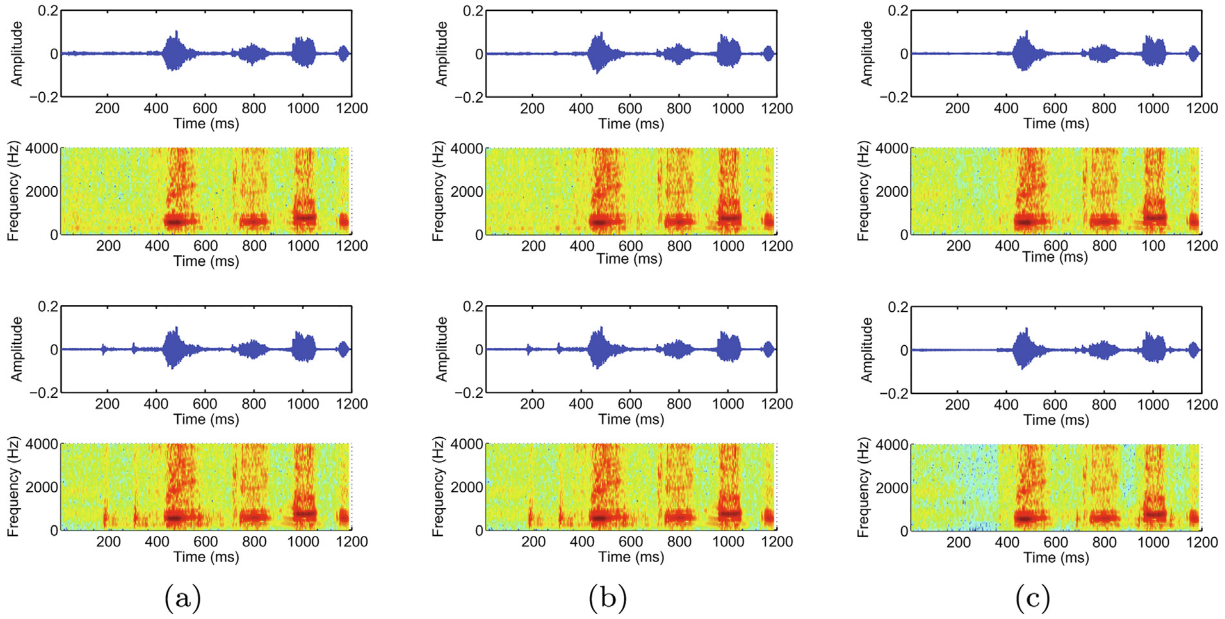

Fig. 5. Time-domain waveforms and corresponding time–frequency spectrograms under fixed layout. The upper row is the separation result with interference 1, while the bottom row is the separation result with interference 2. (a) SEGAN; (b) Cycle + CGAN; (c) CDGAN.

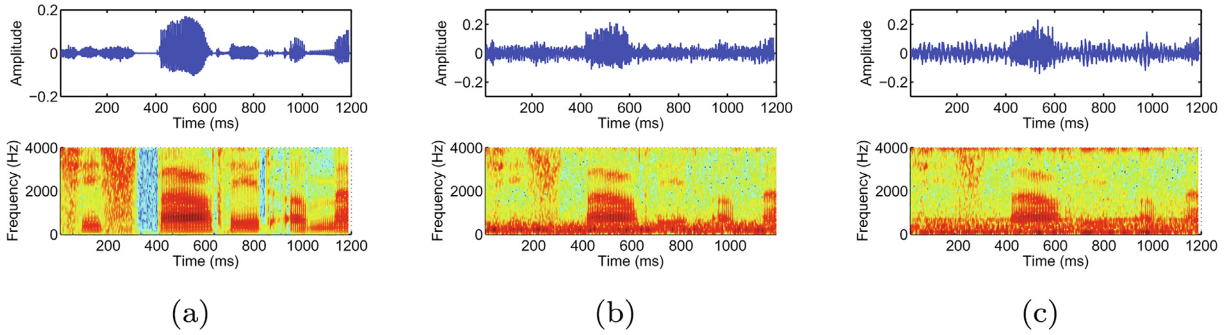

Fig. 6. Time-domain waveforms and corresponding time–frequency spectrograms under random layout. (a) Target speech; (b) Mixture with interference 3, SIR = −1.29 dB; (c) Mixture with interference 4, SIR = −1.20 dB.

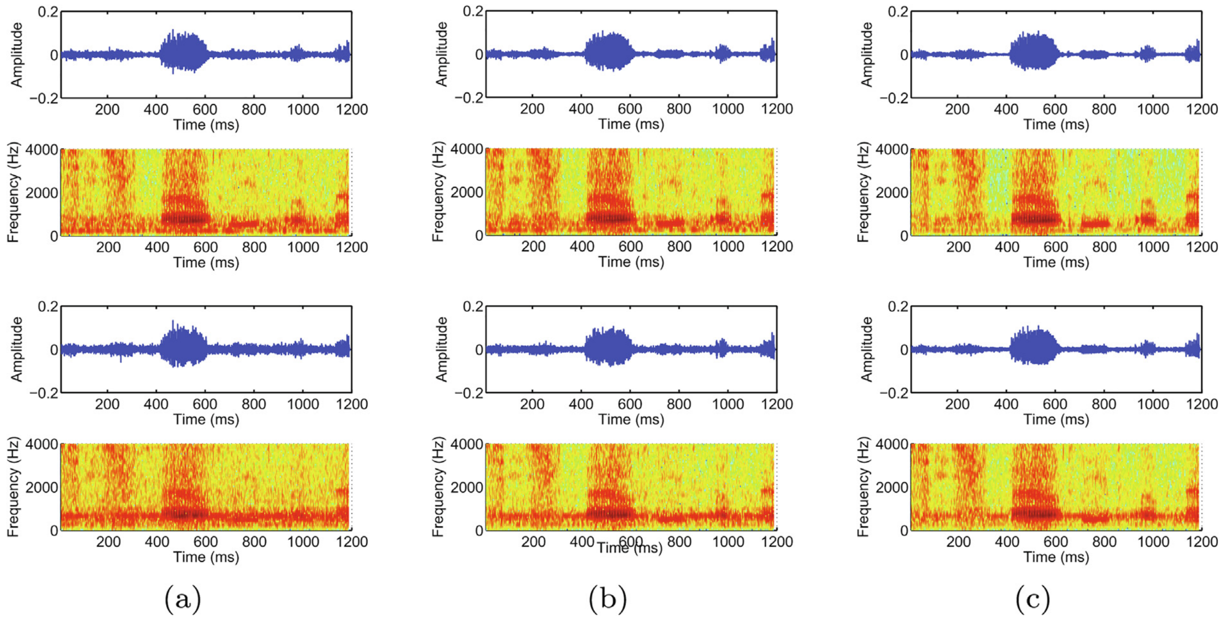

Fig. 7. Time-domain waveforms and corresponding time–frequency spectrograms under random layout. The upper row is the separation result with interference 3, while the bottom row is the separation result with interference 4. (a) SEGAN; (b) Cycle + CGAN; (c) CDGAN.

Table 5
Speech recognition rate for CDGAN and CBSS under different reverberation time. The average SIR of the mixture is 1.12 dB.

		Recognition Rate (%)							
Method	Clean	Mixed				Separated			
		8 ms	16 ms	32 ms	64 ms	8 ms	16 ms	32 ms	64 ms
CBSS	92.8	61.8	58.3	57.5	57.0	89.3	79.3	61.5	61.3
CDGAN(fixed)						**88.8**	**86.5**	**84.3**	**83.8**
CDGAN(random)						**84.5**	**83.0**	**82.3**	**82.0**

our method has better ability to resist the increasing of reverberation time.

Table 5 shows the comparison of average speech recognition rate of CDGAN and CBSS methods. Speech recognition is based on HTK [24]. The test data contains 400 words (100 sentences, 4 words per sentence). We give the average recognition rate of clean target signal and convolutive mixed signal as reference. It can be seen that our method has better recognition rate under the condition of long reverberation time.

5. Conclusion

In this paper, we propose a new method for single channel convolutive mixed speech separation based on GAN. We divide the separation process into two steps: dereverberation and separation of speech and interference. We modified the time–frequency spectrum frame vector, this help the network to learn the mapping of phase information. We use improved CycleGAN to realize the mapping from convolutive mixing to additive mixing. On this basis, we propose a differential GAN to separate target speech and interference. Simulation results show that our method has better separation performance and recognition rate under long reverberation time.

CRediT authorship contribution statement

Yang Li: Methodology, Data curation, Software, Writing - original draft. **Wei-Tao Zhang:** Conceptualization, Investigation, Writing - review & editing. **Shun-Tian Lou:** Supervision, Validation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is supported by a grant from the National Natural Science Foundation of China (No. 62071350)

References

[1] K. Rahbar, J.P. Reilly, A frequency domain method for blind source separation of convolutive audio mixtures, IEEE Trans. Speech Audio Process. 13 (5) (2005) 832–844, https://doi.org/10.1109/TSA.2005.851925.

[2] W.T. Zhang, J.L. Sun, Nonunitary joint diagonalization for overdetermined convolutive blind signal separation, 26th European Signal Processing Conference (EUSIPCO), 2018, pp. 1232–1236, https://doi.org/10.23919/EUSIPCO.2018.8553132.

[3] W. Cheng, Z. Jia, X. Chen, L. Gao, Convolutive blind source separation in frequency domain with kurtosis maximization by modified conjugate gradient, Mech. Syst. Signal Process. 134 (2019), https://doi.org/10.1016/j.ymssp.2019.106331 106331.

[4] S. Boll, Suppression of acoustic noise in speech using spectral subtraction, in: IEEE International Conference on Acoustics Speech and Signal Processing, vol. 4, 1979, pp. 200–203. doi:10.1109/ICASSP.1979.1170696.

[5] J. Chen, J. Benesty, Y. Huang, S. Doclo, New insights into the noise reduction wiener filter, IEEE Trans. Audio Speech Lang. Process. 14 (4) (2006) 1218–1234, https://doi.org/10.1109/TSA.2005.860851.

[6] N. Mohammadiha, P. Smaragdis, A. Leijon, Supervised and unsupervised speech enhancement using nonnegative matrix factorization, IEEE Trans. Audio Speech Lang. Process. 21 (10) (2013) 2140–2151, https://doi.org/10.1109/TASL.2013.2270369.

[7] Y. Xu, J. Du, L. Dai, C. Lee, An experimental study on speech enhancement based on deep neural networks, IEEE Signal Process. Lett. 21 (1) (2014) 65–68, https://doi.org/10.1109/LSP.2013.2291240.

[8] S. Nie, H. Zhang, X. Zhang, W. Liu, Deep stacking networks with time series for speech separation, in: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing – Proceedings, 2014, pp. 6667–6671. doi:10.1109/ICASSP.2014.6854890..

[9] H. Zhang, X. Zhang, S. Nie, G. Gao, W. Liu, A pairwise algorithm for pitch estimation and speech separation using deep stacking network, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) , 2015, pp. 246–250, https://doi.org/10.1109/ICASSP.2015.7177969.

[10] P. Huang, M. Kim, M. Hasegawa-Johnson, P. Smaragdis, Joint optimization of masks and deep recurrent neural networks for monaural source separation, IEEE/ACM Trans. Audio Speech Lang. Process. 23 (12) (2015) 2136–2147, https://doi.org/10.1109/TASLP.2015.2468583.

[11] L. Sun, J. Du, L. Dai, C. Lee, Multiple-target deep learning for lstm-rnn based speech enhancement, Hands-free Speech Communications and Microphone Arrays (HSCMA), 2017, pp. 136–140, https://doi.org/10.1109/HSCMA.2017.7895577.

[12] T. Gao, J. Du, L. Dai, C. Lee, Densely connected progressive learning for lstm-based speech enhancement, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 5054–5058, https://doi.org/10.1109/ICASSP.2018.8461861.

[13] P. Sprechmann, J. Bruna, Y. Lecun, Audio source separation with discriminative scattering networks, in: International Conference on Latent Variable Analysis and Signal Separation, vol. 9237, 2015, pp. 259–267. doi:10.1007/978-3-319-22482-4_30..

[14] A. Simpson, G. Roma, M. Plumbley, Deep karaoke: extracting vocals from musical mixtures using a convolutional deep neural network, in: International Conference on Latent Variable Analysis and Signal Separation, vol. 9237, 2015, pp. 429–436. doi:10.1007/978-3-319-22482-4_50..

[15] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Adv. Neural Inf. Process. Syst. 3 (2014) 2672–2680.

[16] J.F. Zeng, X. Ma, K. Zhou, Photo-realistic face age progression/regression using a single generative adversarial network, Neurocomputing 366 (2019) 295–304, https://doi.org/10.1016/j.neucom.2019.07.085.

[17] P. Isola, J. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5967–5976, https://doi.org/10.1109/CVPR.2017.632.

[18] M. Mirza, S. Osindero, Conditional generative adversarial nets, arXiv preprint arXiv 1411.1784..

[19] J. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2242–2251, https://doi.org/10.1109/ICCV.2017.244.

[20] G. Liu, J. Shi, X. Chen, J. Xu, B. Xu, Improving speech separation with adversarial network and reinforcement learning, International Joint Conference on Neural Networks (IJCNN), 2018, pp. 1–7, https://doi.org/10.1109/IJCNN.2018.8489444.

[21] Z. Fan, Y. Lai, J.R. Jang, Svsgan, Singing voice separation via generative adversarial network, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 726–730, https://doi.org/10.1109/ICASSP.2018.8462091.

[22] D. Michelsanti, Z.H. Tan, Conditional generative adversarial networks for speech enhancement and noise-robust speaker verification, in: Proc. Interspeech 2017, 2017, pp. 1–5. doi:10.21437/Interspeech.2017-1620..

[23] D.R. Campbell, J.P. Kalle, G. Brown, Roomsim, a matlab simulation of shoebox room acoustics for use in teaching and research, Comput. Inf. Syst..

[24] Htk, http://htk.eng.cam.ac.uk/..

Yang Li received the B.Sc. degree in electronic information engineering and M.Sc. degree in circuit and system from Xidian University, Xi'an, China. He has been a full-time Ph.D. student in Xidian University since 2015. His research interests include blind signal processing and speech separation.

Wei-Tao Zhang received the Ph.D. degree in control science and engineering from Xidian University, Xi'an, China, in 2011.

He is currently an Associate Professor at the School of Electronic Engineering, Xidian University. His research interests lie in the area of blind signal processing and machine learning.

Shun-Tian Lou was born in Zhejiang, China, in 1962. He received the B.Sc. degree in automatic control and M.Sc. degree in electronic engineering from Xidian University, Xi'an, China, and the Ph.D. degree in navigation guidance and control from Northwest Polytechnical University, Xi'an, China, in 1985, 1988, and 1999, respectively.

From 1999 to 2002, he was a Postdoctoral Fellow at the Institute of Electronic Engineering, Xidian University. Currently, he is a Professor at the School of Electronic Engineering, Xidian University. His research interests are in the area of signal processing, pattern recognition, and intelligent control using neural network and fuzzy systems.