IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Learning to Generate Steganographic Cover for Audio Steganography using GAN

LANG CHEN, RANGDING WANG, (Member, IEEE), DIQUN YAN, (Member, IEEE), AND JIE WANG, (Student Member, IEEE)
Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China

Corresponding author: Rangding Wang (wangrangding@nbu.edu.cn).

This work was supported in part by the National Natural Science Foundation of China under Grant U1736215, Grant 61672302, and Grant 61901237, in part by the Natural Science Foundation of Zhejiang Province under Grant LY20F020010 and Grant LY17F020010, in part by the K.C. Wong Magna Fund, Ningbo University, and in part by the Natural Science Foundation of Ningbo under Grant 2019A610103.

ABSTRACT Audio steganography aims to exploit the human auditory redundancy to embed the secret message into cover audio, without raising suspicion when hearing it. However, recent studies have shown that the existing audio steganography can be easily exposed with the deep learning based steganalyzers by extracting high-dimensional features of stego audio for classification. The existing audio steganography schemes still have room for improvements. In this work, we propose an audio steganography framework that could automatically learn to generate superior steganographic cover audio for message embedding. Specifically, the training framework of the proposed framework consists of three components, namely, generator, discriminator and trained deep learning based steganalyzer. Then the traditional message embedding algorithm LSBM, is employed to embed the secret message into the steganographic cover audio to obtain stego audio, which is delivered to the trained steganalyzer for misclassifying as cover audio. Once the adversarial training is completed among these three parties, one can obtain a well-trained generator, which could generate steganographic cover audio for subsequent message embedding. In the practice of our proposed method, the stego audio is produced by embedding the secret message into the steganographic cover audio using a traditional steganography method. Experimental results demonstrate that our proposed audio steganography can yield steganographic cover audio that preserves a quite high perception quality for message embedding. We have compared the detection accuracies with the existing audio steganography schemes as presented in our experiment, the proposed method exhibits lower detection accuracies against the state-of-the-art deep learning based steganalyzers, under various embedding rates. Codes are publicly available at https://github.com/Chenlang2018/Audio-Steganography-using-GAN.

INDEX TERMS Audio steganography, deep learning based steganalysis, generative adversarial network (GAN).

I. INTRODUCTION

Steganography is a technique that utilizes the human perception redundancy to embed secret message into a cover such as video, image and audio. The cover with embedded data, i.e., the stego, could bypass adversary monitoring and realize the covert communication. Steganography has been applied to many multimedia security scenarios, e.g., privacy protection [1].

According to different embedding strategies, steganography can be classified into non-adaptive steganography and adaptive steganography. In general, non-adaptive steganography methods often modify all elements of cover in an indiscriminate manner. The representative works along this line are LSB [2] and LSB Matching (LSBM) [3]. Instead, adaptive steganography methods selectively embed the secret message in areas that are unlikely to be exposed. The renowned adaptive steganography algorithms include WOW [4], HUGO [5], HILL [6], S-UNIWARD [7]. These methods are all based on Syndrome-Trellis Codes (STC) [8]. In the adaptive steganography framework, a distortion cost function of each embedding position in the cover is defined to characterize the degree of distortion for stego. The total distortion is then minimized under the assumption that all embedded operations are independent of each other. Note that, the adaptive steganography can be easily detected by Spatial Rich Model (SRM) [9] steganalysis method. Such methods use multiple high-pass filters to preprocess the stego to magnify steganographic signals.

Recently, deep learning has achieved great breakthroughs in many fields, e.g., computer vision, natural language processing and speech recognition, and it has been transforming the information hiding research in the last few years. As the fact that deep learning based steganalysis methods [10], [11] have significantly exceeded the traditional steganalysis methods [12], [13] in detecting conventional steganography algorithms, which may bring challenges to the development of steganography. So researchers have begun to propose deep learning based image steganography methods. Volkhonskiy *et al.* [14] first proposed Steganographic GAN (SNGAN) to implement cover modified steganography. Unlike SGAN, Hayes *et al.* [15] proposed to use the secret message and cover image to generate stego image. Zhang *et al.* [16] proposed a novel data-driven information hiding scheme called "generative steganography by sampling" (GSS) that the stego image was directly sampled by a generator without using covers. Tang *et al.* [17] combined GAN with adaptive steganography to propose automatic steganographic distortion learning framework (ASDL-GAN), in which the GAN component was supposed to learn the embedding change probability map.

In addition to the GAN-based steganography approaches, researchers also proposed steganographic methods inspired from the adversarial examples. Zhang *et al.* [18] employed Fast Gradient Sign Method (FGSM) [19] to devise a steganography model, where the core idea was to add random noise for simulating embedding operation on the cover image to generate "stego image" with noise, and then performed adversarial attack on deep learning based steganalysis network to acquire perturbation. Finally, the adaptive steganography algorithm was used for embedding secret messages. The reported results showed that the proposed method achieved superior performance in resisting deep learning based steganalysis methods. While Tang *et al.* [20] thoroughly investigated adversarial examples from the perspective of steganography. They suggested that adversarial example can be used to adjust steganographic distortion cost effectively.

The existing conventional audio steganography cannot resist deep learning based steganalysis methods. Lin-Net [21] and Chen-Net [22] are two state-of-the-art deep learning based audio steganalysis methods, which have achieved excellent classification performance for detecting traditional audio steganography algorithms. Moreover, the emerging steganography algorithms based on deep learning are mainly focused in the image domain, while the deep learning based audio steganography algorithms have drawn less attention, which still have room for improvements. Therefore, this paper is devoted to exploring steganography in the audio domain. The proposed training framework consists of three components: generator, discriminator and trained deep learning based steganalyzer. The original cover audio is taken as the input to the generator for generating undistinguishable steganographic cover audio. Then the traditional message embedding algorithm LSBM, is employed to embed the secret message into the generated steganographic cover audio to obtain stego audio, which is delivered to the trained steganalyzer for being misclassified as cover audio. This is trying to fool the trained steganalyzer for outputting wrong prediction probabilities. When misclassification occurs, the error corresponding to prediction loss will be back-propagated to the generator for updating the weight parameters. Once the adversarial training among these three parties is completed, one can obtain a well-trained generator to generate steganographic cover audio for subsequent message embedding, which ensures that the data distribution of generated steganographic cover audio matches well with that of messages. It should be remarked that the steganographic cover audio refers to the cover audio that's suitable for message embedding, not the stego audio. The stego audio is produced by embedding the secret message into the steganographic cover audio using a traditional steganography method. Experimental results demonstrate that our proposed audio steganography can yield cover audio that preserves a quite high perception quality for message embedding, and the proposed method exhibits superior undetectability against the state-of-the-art deep learning based steganalyzers, when comparing with the existing audio steganography methods. The main contributions of this paper are summarized as follows:

- We carefully design the network architecture of the generator and discriminator of GAN framework, which ensures the generator learn to generate steganographic cover audio with high perception quality.
- We not only use L_1 norm to measure the similarities between cover audio and original audio, but also between stego audio and original audio, which further enhances the undetectability of proposed audio steganography method.
- Extensive experiments are conducted to demonstrate the effectiveness and superiority of the proposed audio steganography method, compared with the conventional audio steganography methods.

The rest of this paper is organized as follows. Section II reviews the related work, Section III describes the proposed framework, including the network architecture of generator and discriminator, the loss function and the training strategy. The experiment results are demonstrated in Section IV, with perception quality of steganographic cover audio, comparison with existing methods and ablation experiment. Finally, conclusions are drawn in Section V.

II. RELATED WORK

In this section, we first briefly review the generative adversarial networks, and then state the recent advances in GAN-based steganography approaches.

A. GENERATIVE ADVERSARIAL NETWORK

The generative adversarial network (GAN) was first proposed by Goodfellow *et al*. [23]. The basic purpose of GAN is to utilize the real samples for establishing a generator, which could generate samples that obey the same data distribution as the real samples. The generator can be considered as a transformation that transforms a random noise into the space of real samples. In order to obtain such a generator, a discriminator is introduced to distinguish the generated samples from the real data, which is aimed at enhancing the performance of generator. In brief, through the continuous combat game between generator and discriminator, an equilibrium point in the training process is finally reached. So this makes it possible that the discriminator could not distinguish the generated samples from the real ones.

However, the training process of naive GAN is not stable, and it is also possible to emerge vanishing gradient problem. So researchers have proposed some improved GANs for optimizing the training of GAN. Arjovsky *et al*. [24] proposed WGAN (Wasserstein GAN), which used Earth-Mover distance instead of JS divergence to measure the distance between the distribution of real samples and that of generated samples. Qi *et al*. [25] proposed Loss-Sensitive GAN (LS-GAN) to limit the loss function to satisfy the Lipschitz constraint. Mirza *et al*. [26] proposed Conditional GAN (CGAN) that adds extra conditional information for both discriminator and generator to guide the training of GAN.

B. GAN-BASED STEGANOGRAPHY APPROACHES

In recent years, researchers have applied GAN into the information hiding domain, and most of the GAN-based steganography approaches are focused on the image domain. Volkhonskiy *et al*. [14] first proposed a steganographic model termed as Steganographic GAN (SGAN), which took random noise as input for generating a cover image that was visually indistinguishable from the original one. Then the corresponding stego image was generated by LSBM. Finally, the generator and steganalyzer were involved into an adversarial game. The goal of such a game was to enforce the steganalyzer to classify the stego image as an authentic cover. Shi *et al*. [27] proposed Secure Steganography GAN (SSGAN) on the basis of SGAN, which employed WGAN to replace the GAN framework of SGAN. This could speed up the training of SSGAN and enhance the perceptual quality of generated images. Hayes *et al*. [15] proposed another GAN-based steganography model (HayesGAN) that took the cover image and secret message as the input of GAN to synthesize the stego image. The discriminator was used for extracting secret messages and evaluating their extraction accuracy. The stegnalyzer evaluated the undetectable ability of synthesized stego image. However, it was difficult to ensure that the embedded secret message could be extracted completely because of the existence of errors. Tang *et al*. [17] combined GAN and adaptive steganography to devise ASDL-GAN for steganographic distortion cost. According to the reported ASDL-GAN, the goal of generator was to generate the modified probability map, and the discriminator (namely steganalyzer) aimed at distinguishing the stego image from the cover image. After several rounds of adversarial training between generator and discriminator, the generator could yield a relatively optimal modified probability map for computing steganographic distortion cost. Finally, STC was employed to embed secret messages based on the steganographic distortion cost. While Yang *et al*. [28] have made several improvements to ASDL-GAN, and proposed to employe tanh-simulator as an activation function to replace TES (Ternary embedding simulator) in ASDL-GAN for solving the problem that TES was difficult to perform gradient back-propagation. The selected channel was also considered in the design of discriminator, so that the learned distortion cost could resist the selected channel based steganalysis methods. In addition, Ye. *et al* [29] proposed a GAN-based audio steganography method which the embedding and extraction of secret audio were accomplished by GAN. Yang *et al*. [30] employed GAN for learning the embedding cost to approach optimal embedding for audio steganography in the temporal domain.

III. GENERATING STEGANOGRAPHIC COVER AUDIO USING GAN

In this section, we first describe the proposed training framework, including the network architecture of generator and discriminator, the loss function and the training strategy.

A. OVERALL FRAMEWORK

Figure 1 demonstrates the training framework of the proposed steganography method, which is consisted of three principal parts: generator, discriminator and trained steganalyzer. It should be pointed out that the trained steganalyzer is implemented on Lin-Net which has been trained to convergence in advance. Specifically, the original cover audio is taken as the input to the generator for generating undistinguishable steganographic cover audio. That is to say, the generated steganographic cover audio shall be as resemble as possible to the original one. Then traditional message embedding algorithm LSBM, is employed to embed the secret message into the steganographic cover audio to obtain stego audio, which is delivered to the trained steganalyzer for being misclassified as cover audio. Once misclassification occurs, the error corresponding to prediction loss will be back-propagated to the generator for updating the weight parameters. It is worth noting that we encourage the steganalyzer misbehavior because our goal is to fool the deep learning based steganalyzer. After adequate adversarial training, the well-trained generator will be obtained. In the ultimate steganography, we use the well-trained generator to generate steganographic cover audio, then traditional steganography algorithm LSBM, will be used to embed secret message on the steganographic cover audio to yield undetectable stego audio. The workflow of the ultimate steganography model is illustrated in Figure 2.

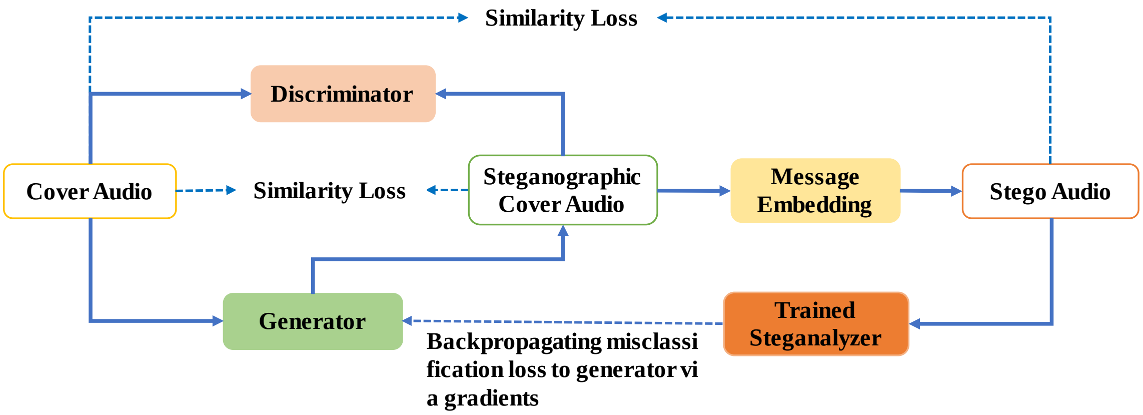


FIGURE 1. The training framework of the proposed method.

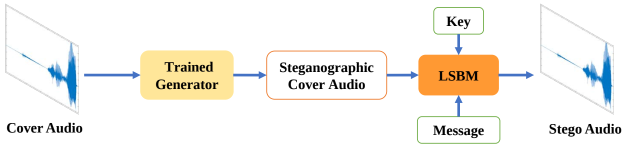


FIGURE 2. The workflow of the ultimate audio steganography model.

B. GENERATOR ARCHITECTURE

Inspired by the fact that U-Net [31] architecture can deal with image-to-image translation tasks in an excellent performance, we elaborately design the architecture of the generator in a U-Net fashion. The goal of generator is to automatically learn to generate steganographic cover audio for message embedding, then the steganalyzer could misclassify corresponding stego audio as cover audio. The architecture of generator as shown in Figure 3 contains 8 convolution layers and 8 deconvolution layers, and Table 1 illustrates the detailed parameter configuration of generator in the proposed method. Specifically, the kernel size of all convolution layers and deconvolution layers are 1×32, with stride 2 and padding 15, cascaded with batch normalization. As the fact that the network is deeper, less content information will be reserved after convolution operations, which may lead the steganographic cover audio to enjoy poor perceptual quality. Hence we employ skip connection as a shortcut to concatenate the feature map with the same size between convolution layers and corresponding deconvolution layers. Skip connection can render the deconvolution layers to share the features extracted by convolution layers, which benefits the perceptual quality of steganographic cover audio. We concatenate the feature map from Group i to Group $L - i$, here L is 16. We apply parametric rectified linear units (PReLU) [32] from Group 1 to Group 15. The tangent activation function in Group 17 is applied to guarantee the sampling values of steganographic cover audio range from -1 to 1. It should be pointed out that all convolution and deconvolution layers of the generator are initialized with Xavier [33] method.

C. DISCRIMINATOR ARCHITECTURE

The architecture of discriminator in the proposed method is shown in Figure 4. The discriminator aims to distinguish original audio from the steganographic cover audio, which could motivate the generator to yield cover audio that approximates the data distribution presented in the original audio. In our proposed method, we employ the spectral normalization technique which is proposed in the work Spectral Normalization GAN (SNGAN)[1] [34]. Compared with other GANs, one of the highlights of SNGAN is that the weight parameters of all convolution layers and fully-connected layers are normalized by spectral norm. The reason why we employ spectral normalization is that this could stabilize the training process of GAN and finally motivate the generator in the proposed method to yield steganographic cover audio with better perceptual quality. In more detail, the discriminator in our proposed method contains 9 convolution layers and 1 fully-connected layer. It should be pointed out that we redesign a novel convolution kernel in the convolution layer by normalizing its weight parameters using spectral norm. This redesigned convolution kernel may be named as “SNConv”. In the redesigned fully-connected layer named “SNLinear”, of which the weight parameters are also normalized by the spectral norm. Each convolution block is cascaded by LeakyReLU [35] with slope setting to 0.01, the sigmoid function is placed in the back of fully-connected layer. The kernel size of all convolution layers is set to 1×32, with stride 1 and padding 15.

D. LOSS FUNCTION

On behalf of forcing the generator to yield steganographic cover audio with excellent perceptual quality, the loss function is a significant criterion to guide the training process of generator and discriminator. For the purpose of ensuring the discriminator possess a relatively better discriminant

[1] The implementation of SNGAN is that the weight parameters of all convolution layers and fully-connected layers are normalized by the spectral norm

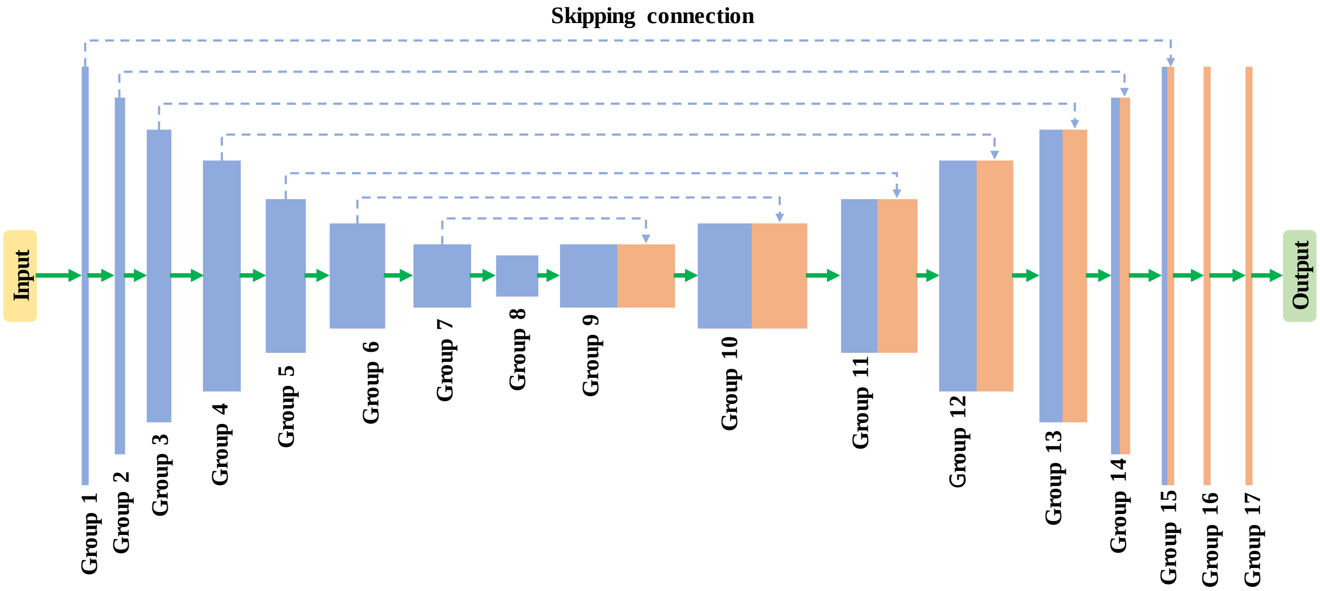


FIGURE 3. The architecture of generator in the proposed method.

TABLE 1. The parameter configuration of generator in the proposed method.

Group	Operation	Convolution/Deconvolution kernels	Output Size
Input Audio	/	/	1 × 16384
Group 1	Conv1d+Batch Normalization+PReLU	16 × (1 × 32)	16 × 8192
Group 2	Conv1d+Batch Normalization+PReLU	32 × (1 × 32)	32 × 4096
Group 3	Conv1d+Batch Normalization+PReLU	32 × (1 × 32)	32 × 2048
Group 4	Conv1d+Batch Normalization+PReLU	64 × (1 × 32)	64 × 1024
Group 5	Conv1d+Batch Normalization+PReLU	64 × (1 × 32)	64 × 512
Group 6	Conv1d+Batch Normalization+PReLU	128 × (1 × 32)	128 × 256
Group 7	Conv1d+Batch Normalization+PReLU	128 × (1 × 32)	128 × 128
Group 8	Conv1d+Batch Normalization+PReLU	256 × (1 × 32)	256 × 64
Group 9	deConv1d+Batch Normalization+PReLU Concatenate Group 7 to Group 9	128 × (1 × 32)	128 × 128
Group 10	deConv1d+Batch Normalization+PReLU Concatenate Group 6 to Group 10	128 × (1 × 32)	128 × 256
Group 11	deConv1d+Batch Normalization+PReLU Concatenate Group 5 to Group 11	64 × (1 × 32)	64 × 512
Group 12	deConv1d+Batch Normalization+PReLU Concatenate Group 4 to Group 12	64 × (1 × 32)	64 × 1024
Group 13	deConv1d+Batch Normalization+PReLU Concatenate Group 3 to Group 13	32 × (1 × 32)	32 × 2048
Group 14	deConv1d+Batch Normalization+PReLU Concatenate Group 2 to Group 14	32 × (1 × 32)	32 × 4096
Group 15	deConv1d+Batch Normalization+PReLU Concatenate Group 1 to Group 15	16 × (1 × 32)	16 × 8192
Group 16	deConv1d	1 × (1 × 32)	1 × 16384
Group 17 (Output Audio)	Tanh	/	1 × 16384

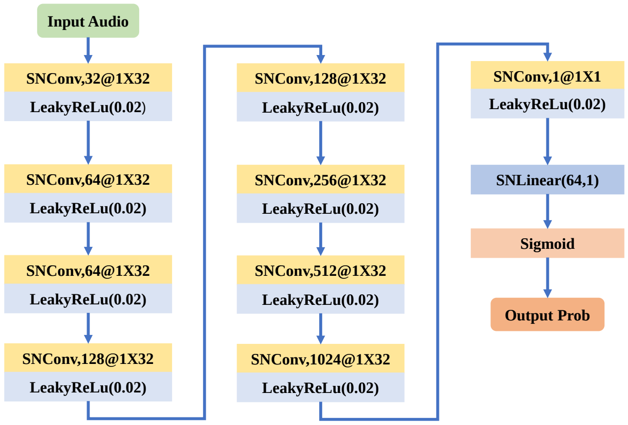


FIGURE 4. The architecture of discriminator in the proposed method.

capability, and also facilitate the cover audio to resist the perturbation caused by embedding secret messages. Therefore, we divide the training process into two stages: in stage 1, only discriminator and generator are incorporated in the adversarial training, and in stage 2, the trained steganalyzer begins to join in the remaining adversarial training. Correspondingly, we elaborately design loss functions for two stages, respectively. Especially, in order to define the loss function, we set the corresponding labels of steganographic audio and stego audio both as 0, and the corresponding labels of original audio and cover audio both as 1.

For stage 1, the loss function can be expressed as

$$\mathcal{L}_{\text{stage1}} = \mathcal{L}_{\mathcal{D}} + \mathcal{L}_{\mathcal{G}_1} \tag{1}$$

where $\mathcal{L}_{\text{stage1}}$ represents the loss of GAN framework in stage 1. The losses $\mathcal{L}_{\mathcal{D}}$ and $\mathcal{L}_{\mathcal{G}_1}$ are computed in terms of binary cross entropy. The loss for generator in stage 1 can be calculated by

$$\mathcal{L}_{\mathcal{G}_1} = \mathbb{E}_x[\log(1 - \mathcal{D}(\mathcal{G}(x)))]. \tag{2}$$

where x denotes original audio (the input of generator), $\mathbb{E}_x[\cdot]$ is the expectation operator over the input geniue audio clips. The discriminator aims at distinguishing the steganographic cover audio from the original one. Thus the loss of discriminator can be computed by

$$\mathcal{L}_{\mathcal{D}} = -\{\mathbb{E}_x[\log\mathcal{D}(\mathcal{G}(x))] + \mathbb{E}_x[\log(1 - \mathcal{D}(x))]\}. \tag{3}$$

For stage 2, the loss function is exhibited as follows.

$$\mathcal{L}_{\text{stage2}} = \alpha\mathcal{L}_{\text{GAN}} + \beta\mathcal{L}_{\text{Sim}}, \tag{4}$$

where $\mathcal{L}_{\text{stage2}}$ represents the loss of GAN framework in stage 2, and $\mathcal{L}_{\text{Sim}}$ is the similarity loss function to measure the similarity between steganographic cover audio and original audio. The hyperparameters α and β balance the importances between the two parts. More specifically, $\mathcal{L}_{\text{GAN}}$ is composed of the loss of generator and discriminator, which can be denoted as

$$\mathcal{L}_{GAN} = \mathcal{L}_{\mathcal{D}} + \mathcal{L}_{\mathcal{G}_2} \tag{5}$$

The losses $\mathcal{L}_{\mathcal{D}}$ and $\mathcal{L}_{\mathcal{G}_2}$ are computed in terms of binary cross entropy. For the generator, its goal is to yield steganographic audio with no differences in auditory compared with original audio, and the stego audio produced by embedding operations is sent to the steganalyzer for predicting the probability belonging to cover audio. The loss between the cover audio label and the prediction probability should be minimized, which is used to propagate to the generator for updating its parameters. This is devoted to forcing the generator to yield preferable steganographic cover audio for embedding message so that the stego audio cannot easily be distinguished by steganalyzer. Therefore, the loss of generator can be calculated by

$$\mathcal{L}_{\mathcal{G}_2} = \mathbb{E}_x[\log(1 - \mathcal{D}(\mathcal{G}(x)))] + \mathbb{E}_x[\log(1 - \mathcal{S}(\mathcal{F}(\mathcal{G}(x))))], \tag{6}$$

where x denotes original audio. $\mathbb{E}_x[\cdot]$ is the expectation operator over the input original audio clips. $\mathcal{F}(\cdot)$ denotes the traditional information method, e.g., LSBM, and $\mathcal{D}(\cdot)$, $\mathcal{S}(\cdot)$, and $\mathcal{G}_2(\cdot)$ denote the discriminator, steganalyzer and generator, respectively. The similarity loss term $\mathcal{L}_{Sim}$ shall has two parts: The first part measures the differences between steganographic cover audio and the original cover audio, and the second measures the differences between the stego audio and original audio. This can be expressed as

$$\mathcal{L}_{Sim} = \mathbb{E}_x[\| \mathcal{G}(x) - x \|_1] + \mathbb{E}_x[\| \mathcal{F}(\mathcal{G}(x)) - x \|_1]. \tag{7}$$

Here, $\mathcal{L}_1$ norm is applied to measure the similarity loss between steganographic cover audio and original audio. In our experiment, $\mathcal{L}_2$ norm is also used to measure the aforementioned similarity loss, then we find that the steganographic cover audio with $\mathcal{L}_1$ norm enjoys slightly better perceptual quality compared to that generated with $\mathcal{L}_2$ norm. Finally, we would like to remark that the steganalyzer is a well-trained neural network based on Lin-Net [21]. Hence in stage 2, the steganalyzer is involved, but all its model parameters are fixed; it is only responsible for back-propagating the prediction errors via gradients. The back-propagated misclassification errors are used to update the parameters of generator, which may force the generator into learning to generate steganographic cover audio that suitable for message embedding, and attempt to deceive the steganalyzer.

E. TRAINING STRATEGY

In our proposed method, the training process includes two stages. In stage 1, we have trained the GAN framework (i.e., generator $\mathcal{G}$ and $\mathcal{D}$) for N epochs in advance (We empirically set N as 30). In stage 2, the steganalyzer joins in the training process to start post-training, which takes around another 100 epochs. This is for the suppose of guaranteeing the discriminator to possess stronger discriminant ability for distinguishing the steganographic cover audio from the original one, and also prompt the generator to yield cover audio with superior perceptual quality. The generator and discriminator are trained alternatively, that is to say, when training the generator, the weight parameters of discriminator are fixed and vice versa. It should be pointed out that the parameters of steganalyzer are not updated in the entire training process. The role of steganalyzer is orientated to output the confidence that stego audio belongs to the original one. Then the produced misclassification loss can be back-propagated via gradients to generator $\mathcal{G}$, which is applied to update the parameters of generator. The training strategy is briefly described in Algorithm 1.

Algorithm 1 Training Strategy of the Proposed Method

```
Input: original audio $x$, traditional information embedding method $\mathcal{F}$, trained steganalyzer $\mathcal{S}$, loop variable $i$, pre-training epochs $N$ for stage 1, total training epochs $M$, learning rate $\gamma$.
Output: The well-trained generator $\mathcal{G}^*$.
Initialization: Initialize the weight parameters of generator $\theta_{\mathcal{G}}$ and discriminator $\theta_{\mathcal{D}}$ using Xavier method.
for $i$ = 1 to $M$ do
    Generate fake cover audio $c$=$\mathcal{G}(x)$.
    if $i \leq N$ then
        Update the weight parameters of generator and discriminator by gradient descent optimizer in stage 1, respectively. $\theta_{\mathcal{G}} = \theta_{\mathcal{G}} - \gamma\nabla_{\theta_{\mathcal{G}}} L_{\text{stage1}}, \theta_{\mathcal{D}} = \theta_{\mathcal{D}} - \gamma\nabla_{\theta_{\mathcal{D}}} L_{\text{stage1}}$.
    else
        Embed secret message using information embedding algorithm to yield stego audio $s = \mathcal{F}(c)$.
        Update the weight parameters of generator and discriminator by gradient descent optimizer in stage 2, respectively. $\theta_{\mathcal{G}} = \theta_{\mathcal{G}} - \gamma\nabla_{\theta_{\mathcal{G}}} \mathcal{L}_{\text{stage2}}, \theta_{\mathcal{D}} = \theta_{\mathcal{D}} - \gamma\nabla_{\theta_{\mathcal{D}}} \mathcal{L}_{\mathcal{D}}$. // trained steganalyzer joins in the training process for backpropagating gradients to generator.
    end if
end for
```

IV. EXPERIMENTAL RESULTS

A. EXPERIMENTAL SETUP

TIMIT corpus [36] and UME corpus [37] are two widely-used datasets in speech recognition speaker recognition and so on. Both TIMIT and UME contain uncompressed mono audio with a sampling frequency of 16 kHz. We have conducted experiments on the two datasets to verify the effectiveness of the proposed method, respectively. TIMIT is used for training the generator in the proposed method, and UME is used for evaluating the undetectability performance by using different deep learning based steganalysis methods. For the sake of facilitating to design the framework of generator, we tailored the audio files into small clips with 16384 sampling points. In the training process, 15000 small clips are used for training the generator, the mini-batch size is set to 32. The Adam optimizer is used with the learning rate 0.0001. Empirically, the hyper-parameter α and β in the loss function are both set to be 1. The input is normalized firstly before feeding to the generator, the maximum and minimum normalization trick is used for normalizing the input audio into $[-1, 1]$. The proposed method is implemented using PyTorch and trained on four NVIDIA RTX1080 Ti GPUs with 11 GB memory.

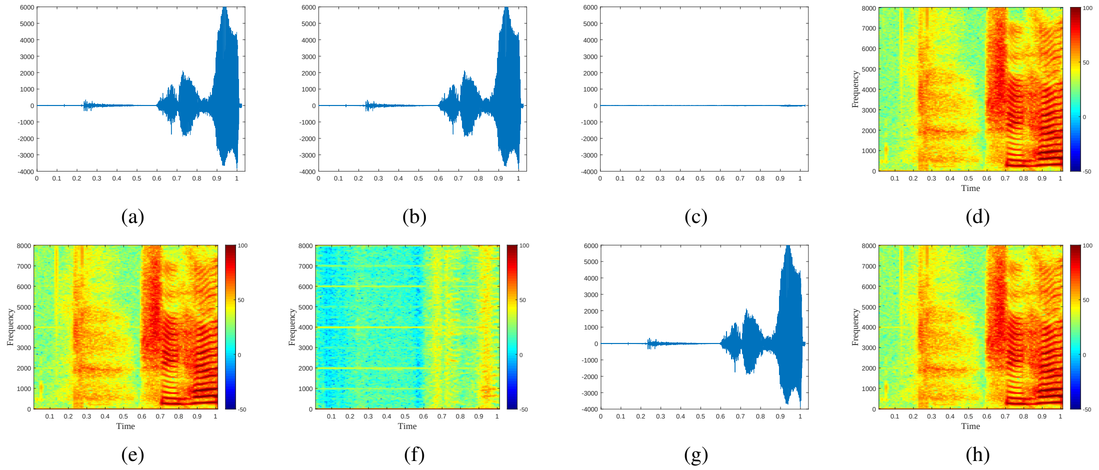
B. PERCEPTION QUALITY OF STEGANOGRAPHIC COVER AUDIO

Recall that one goal of our method is to incur minor perturbation on the steganographic cover audio. In other words, the steganographic cover audio shall be acoustically indistinguishable from the original cover audio, and also the stego audio (produced by embedding message on the steganographic cover audio) should be with no differences compared with the original cover audio when hearing it. To show this, the visualization results of one randomly selected original cover audio, steganographic cover audio and corresponding stego audio on UME are illustrated in Figure 5. As can be seen, the waveform and spectrogram of steganographic cover audio are almost the same as that of the original cover audio. The residual waveform validate that the magnitude of the perturbation is quite small when comparing with the original cover audio. The waveform and spectrogram for stego audio are also similar to the steganographic cover audio.

Furthermore, to quantitatively assess the audio perception quality, we employ the widely-used reference audio quality metrics, i.e., the subjective metric PESQ [38] and the objective metric SNR (Peak signal-to-noise ratio). PESQ score ranges from -0.5 to 4.5, and higher value indicates better perception quality. SNR characterizes the average power ratio between the intrinsic signal and the noise. We randomly select 100 test audio samples from UME as references and corresponding steganographic cover audio for evaluation. The average PESQ score is 4.4235 and the SNR is 83.275 dB. This means that the steganographic cover audio cannot be distinguished from the original audio in human hearing, which verifies the effectiveness in generating steganographic cover audio with high perceptual quality by the means of our proposed method.

C. COMPARISON WITH EXISTING METHODS

To demonstrate the performance of our proposed method, two experiments on TIMIT and UME are conducted in our work. We compare the detection accuracy with LSBM [3], STC [8], and Yang *et al.*'s GAN-based method [30] . Two state-of-the-art deep learning based steganalysis methods, Lin-Net [21] and Chen-Net [22] are used to evaluate the undetectability performance of these steganography methods, respectively. For the experiments on TIMIT, 15000 audios clips from TIMIT are selected as the input to the generator trained on TIMIT for generating corresponding 15000 steganographic cover audio samples. Then the bitstream secret messages are embedded into the steganographic cover audio samples. This finally yields 15000 cover-stego pair samples. 12000 cover-stego pairs are used as training set, and the remaining 3000 pairs are for testing set. We have considered five embedding rates for testing, i.e., 0.5 bit per sample (bps), 0.4 bps, 0.3 bps, 0.2 bps, and 0.1 bps. In order to reduce the randomness of the experiment results, we repeat



(a) (b) (c) (d)

(e) (f) (g) (h)

FIGURE 5. The visualization results of one randomly selected original audio, steganographic cover audio and corresponding stego audio on UME. (a) (b) and (c) are the original waveform, the steganographic audio waveform and the residual waveform between (a) and (b), respectively. Similarly, (d) (e) and (f) are the original spectrogram, the steganographic audio spectrogram and the residual spectrogram between (d) and (e), respectively. (g) and (h) are the waveform and spectrogram for stego audio, respectively.

all the experiments for 10 times under randomly splitting training set and test set and then average the detection accuracies. The similar experiments as mentioned above are also conducted on UME. The only difference is that we take the audio clips from UME as the input to the generator trained on TIMIT for generating corresponding steganographic cover audio samples.

The detection accuracy results are tabulated in Table 2. As one can see, generally, for all embedding rates, all test datasets, and all the deep learning based steganalyzers, our method attains lower detection accuracy consistently. This means the proposed method could generate better steganographic cover audio for message embedding, benefiting the conventional steganography methods. With more careful comparisons, for low embedding rates, e.g., 0.1 bps, the detection accuracy of our method ranges from 48.14% to 49.25%, closing to the random guess (i.e., 50%), and a similar phenomenon can be observed for conventional STC. Instead, the detection accuracies for conventional method LSBM all exceed 58%. This suggests that, for lower embedding rates, the conventional method LSBM is more vulnerable to deep learning based methods, while both our proposed method and STC retain good undetectability. However, for large embedding rates, e.g., 0.5 bps, the superiority of our proposed method becomes more pronounced. For instance, for the case of steganalyzer Lin-Net on UME dataset, the detection accuracy's for LSBM and STC are 75.24% and 71.08%, respectively. In contrast, our method yields 63.25%, still enjoying lower undetectability. In addition, compared with Yang *et al.*'s GAN-based method, our proposed method has achieved lower detection accuracies under various embedding rates. For example, when training the generator on TIMIT and evaluating undetectability with UME using Chen-Net, the detection accuracy of our proposed method is 3.23% lower than Yang *et al.*'s method under 0.5 bps. Similarly, our proposed method enjoys preferable undetectability performance when the embedding rate is 0.1 bps. Therefore, whether compared with conventional audio steganogrpahy methods or the existing GAN-based audio steganogrpahy schemes, our proposed method has witnessed excellent undetectability performance under various embedding rates.

D. ABLATION EXPERIMENT

In this section, we have conducted ablation experiments on the proposed framework's main architecture variants as shown in Table 3. Figure 6 illustrates the corresponding PESQ score of steganographic cover audio from UME when these main variants are modified. We can easily find that our proposed framework can steganographic cover audio with the largest PESQ score 4.4235, compared with the other 6 generative steganography models. This means that our proposed framework is the most effective in generating cover audio with excellent perception quality. In addition, variants $\#2$ and $\#4$ could impose prominent influences on the perceptual quality of steganographic cover audio, the PESQ scores are 3.8315 and 3.9256, respectively. We may perceive that there exists obvious noise in the steganographic cover audio when hearing it, which enjoys low auditory experience. To sum up, the architecture variants in the proposed framework are optimal.

V. CONCLUSION

In this work, we proposed to generate a better steganographic cover audio for using the generative adversarial network. Embedding messages on such steganographic cover audio could yield more secure stego audio, which is able to resist the deep learning based steganalyzers. The training framework of the proposed method contains three principal modules: generator, discriminator, and an off-the-shelf deep learning based steganalyzer. We deliberately devised the network architecture of the generator and discriminator, and propose an effective training strategy for adversarial training among

TABLE 2. Comparison of the detection accuracy (%) using Lin-Net [21] and Chen-Net [22] steganalyzers. For each cell, the top number is for Lin-Net and the bottom number is for Chen-Net. Lower detection accuracy indicates better undetectability performance.

Dataset	Steganography	Embedding rates (bps)				
		0.5	0.4	0.3	0.2	0.1
TIMIT	LSBM	76.28 72.15	72.54 70.22	69.15 67.23	67.45 63.35	60.24 59.41
	STC	70.12 68.22	68.72 63.48	62.35 60.25	55.18 54.95	52.32 50.19
	Yang *et al.*'s method	68.12 66.39	62.24 60.69	59.43 57.34	54.56 52.64	51.02 50.11
	Proposed method	**64.39** **61.25**	**61.58** **55.80**	**55.28** **54.23**	**52.33** **51.29**	**49.25** **48.62**
UME	LSBM	75.24 71.65	72.35 65.21	70.24 63.49	67.38 60.14	60.15 58.31
	STC	71.08 65.21	68.27 62.08	60.12 59.49	56.49 52.65	51.13 50.89
	Yang *et al.*'s method	66.42 65.62	62.06 61.20	59.04 58.47	55.19 53.10	51.36 50.09
	Proposed method	**63.25** **62.39**	**61.42** **59.56**	**55.13** **55.46**	**52.11** **50.49**	**49.03** **48.14**

TABLE 3. The main modified architecture variants in the proposed method

Index	Modified variants
#1	Proposed framework
#2	Remove spectral normalization
#3	Remove similarity loss
#4	Remove skipping connection in the generator framework
#5	Remove PReLU in the generator framework
#6	Remove batch normalization in the generator framework
#7	Remove LeakyReLU in the discriminator framework

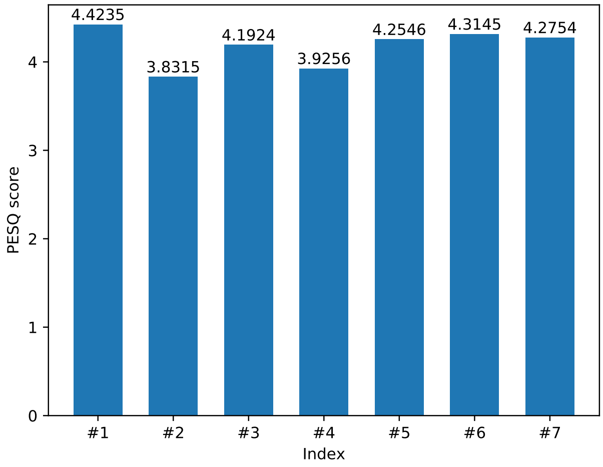


FIGURE 6. The corresponding PESQ score of steganographic cover audio from UME when the main architecture variants of the proposed method are modified.

the three modules in the proposed framework. Once the adversarial training is completed among these three parties, one can obtain a well-trained generator, which could generate steganographic cover audio for subsequent message embedding. By using the well-trained generator, one can use conventional steganography for embedding secret the message as usual. Experimental results show that the generator of the proposed audio steganography method can yield steganographic cover audio with high perception quality, while retaining reasonably good undetectability performance, even under large embedding rates.

REFERENCES

[1] P. Mathivanan, S. E. Jero, P. Ramua, and A. B. Ganesh, "QR code based patient data protection in ECG steganography," *Australas. Phys. Eng. S.*, vol. 41, no. 4, pp. 1057–1068, Nov. 2018.

[2] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM SYST. J.*, vol. 35, no. 3. 4, pp. 313–336, 1996.

[3] T. Sharp, "An implementation of key-based digital signal steganography," in *Proc. Int. Workshop Inf. Hiding*, 2001, pp. 13–26.

[4] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2012, pp. 234–239.

[5] T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Proc. Int. Workshop Inf. Hiding*. Berlin, Germany: Springer, 2010, pp. 161–177.

[6] B. L, S. Tan, M. Wang, and J. Huang, "Investigation on cost assignment in spatial image steganography," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 8, pp. 1264–1277, May. 2014.

[7] V. Holub and J. Fridrich, "Digital image steganography using universal distortion," in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, 2013, pp. 59–68.

[8] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 920–935, Apr. 2011.

[9] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 868–882, May. 2012.

[10] G. Xu, H. Z. Wu, and Y. Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Singal Proc. Let.*, vol. 23, no. 5, pp. 708–712, Mar. 2016.

[11] S. Wu, S. Zhong, and Y. Liu, "Deep residual learning for image steganalysis," *Multimed. Tools Appl.*, vol. 77, no. 9, pp. 10437–10453, Feb. 2018.

[12] T. Pevny, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 215–224, Mar. 2010.

[13] V. Holub and J. Fridrich, "Random projections of residuals for digital image steganalysis," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 12, pp. 1996–2006, Oct. 2013.

[14] D. Volkhonskiy, I. Nazarov, and E. Burnaev, "Steganographic generative adversarial networks," 2017, *arXiv:1703.05502*. [Online]. Available: https://arxiv.org/abs/1703.05502.

[15] J. Hayes and G. Danezisv, "Generating steganographic images via adversarial training," 2017, *arXiv:1703.00371*. [Online]. Available: https://arxiv.org/abs/1703.00371.

[16] Z. Zhang, J. Liu, Y. Ke, Y. Lei, J. Li, M. Zhang, and X. Yang, "Generative steganography by sampling," *IEEE Access*, vol. 7, pp. 118586–118597, May. 2019.

[17] W. Tang, S. Tan, B. Li, and J. Huang, "Automatic steganographic distortion learning using a generative adversarial network," *IEEE Singal Proc. Let.*, vol. 24, no. 10, pp. 1547–1551, Aug. 2017.

[18] Y. Zhang, W. Zhang, K. Chen, J. Liu, Y. Liu, and N. Yu, "Adversarial examples against deep neural network based steganalysis," in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, 2018, pp. 67–72.

[19] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*. [Online]. Available: https://arxiv.org/abs/1412.6572.

[20] W. Tang, B. Li, S. Tan, M. Barni, and J. Huang, "CNN-based adversarial embedding for image steganography," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 8, pp. 2074–2087, Jan. 2019.

[21] Y. Lin, R. Wang, D. Yan, L. Dong, and X. Zhang, "Audio steganalysis with improved convolutional neural network," in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, 2019, pp. 210–215.

[22] B. Chen, W. Luo, and H. Li, "Audio steganalysis with convolutional neural network," in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, 2018, pp. 85–90.

[23] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process Syst.*, 2014, pp. 2672–2680.

[24] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: https://arxiv.org/abs/1701.07875.

[25] G. J. Qi, "Loss-sensitive generative adversarial networks on lipschitz densities," *Int. J. Comput. Vis.*, vol. 128, pp. 1118–1140, May. 2020.

[26] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: https://arxiv.org/abs/1411.1784.

[27] H. Shi, J. Dong, W. Wang, Y. Qian, and X. Zhang, "SSGAN: Secure steganography based on generative adversarial networks," 2017, *arXiv:1707.01613*. [Online]. Available: https://arxiv.org/abs/1707.01613.

[28] J. Yang, D. Ruan, J. Huang, X. Kang, and Y. Q. Shi, "An embedding cost learning framework using GAN," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 839–851, Jun. 2019.

[29] D. Ye, S. Jiang, and J. Huang, "Heard more than heard: an audio steganography method based on GAN," 2019, *arXiv:1907.04986*. [Online]. Available: https://arxiv.org/abs/1907.04986.

[30] J. Yang, H. Zheng, X. Kang, and Y. Q. Shi, "Approaching optimal embedding in audio steganography with GAN," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*, 2020, pp. 2827–2831.

[31] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026-1034.

[33] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2010, pp. 249-256.

[34] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," 2018, *arXiv:1802.05957*. [Online]. Available: https://arxiv.org/abs/1802.05957.

[35] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1-6.

[36] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," *STIN.*, vol. 93, p. 27403, Feb. 1993.

[37] "Priority areas 'advanced utilization of multimedia to promote higher education reform' speech database - English speech database read by Japanese students,". 2002.

[38] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality- a new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE Int. Conf . Acoust Speech Signal Process*, 2001, pp. 749–752.

LANG CHEN received the B.S. degree from Central South University of Forestry and Technology, Changsha, China, in 2019. He is currently pursuing the master's degree with the Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo, China. His research interests include deep learning, multimedia information security, information hiding and steganalysis.

RANGDING WANG (Member, IEEE) received the B.S. and M.S. degrees from Northwestern Polytechnical University, Xi'an, China, in 1984 and 1987, respectively, and the Ph.D. degree from Tongji University, Shanghai, China, in 2004. He is currently a Full Professor with the Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo, China. His research interests include deep learning, multimedia information security, information hiding and steganalysis.

DIQUN YAN (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from Ningbo University, Ningbo, China, in 2002, 2008 and 2012, respectively. He was a Visiting Scholar with the New Jersey Institute of Technology, Newark, NJ, USA, from 2014 to 2015. He is currently an Associate Professor with the Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo, China. His research interests include deep learning, multimedia information security and digital forensics.

JIE WANG (Student Member, IEEE) received the B.S. degree from the College of Science and Technology, Ningbo University, Ningbo, China, in 2018. He is currently pursuing the master's degree with the Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo, China. His research interests include deep learning, multimedia information security, information hiding and steganalysis.

• • •