Hindawi
Shock and Vibration
Volume 2020, Article ID 8888578, 10 pages
https://doi.org/10.1155/2020/8888578

Research Article

Application of Generative Adversarial Nets (GANs) in Active Sound Production System of Electric Automobiles

Kai Liang[1] **and Haijun Zhao**[2,3]

[1]*Information Technology Center, Luoyang Institute of Science &Technology, Luoyang 471023, China*
[2]*School of Automotive and Transportation, Tianjin University of Technology and Education, Tianjin 300222, China*
[3]*National Joint Engineering Research Center of Intelligent Vehicle Infrastructure Cooperation and Safety Technology, Tianjin University of Technology and Education, Tianjin 300222, China*

Correspondence should be addressed to Haijun Zhao; hjzhaotj@sina.com

Received 28 July 2020; Revised 10 October 2020; Accepted 19 October 2020; Published 28 October 2020

Academic Editor: Vasudevan Rajamohan

To improve the diversity and quality of sound mimicry of electric automobile engines, a generative adversarial network (GAN) model was used to construct an active sound production model for electric automobiles. The structure of each layer in the network in this model and the size of its convolution kernel were designed. The gradient descent in network training was optimized using the adaptive moment estimation (Adam) algorithm. To demonstrate the quality difference of the generated samples from different input signals, two GAN models with different inputs were constructed. The experimental results indicate that the model can accurately learn the characteristic distributions of raw audio signals. Results from a human ear auditory test show that the generated audio samples mimicked the real samples well, and a leave-one-out (LOO) test show that the diversity of the samples generated from the raw audio signals was higher than that of samples generated from a two-dimensional spectrogram.

1. Introduction

Electric automobiles can create traffic safety risks as their engines emit low sound level during low-speed driving [1]. Standards for noise produced by electric automobiles during low-speed driving are being drafted in many countries [2]. Active sound production systems have been proposed to mimic the sound of an internal combustion engine. In one study [3], a multi-parameter-controlled mimicking algorithm based on digital audio signals was proposed for producing the sound effects of an engine based on multiple parameters including engine speed, driving speed, and acceleration. In another study [4], the superposition theory of speech synthesis technology was used to design a self-adaptive active sound production system based on engine speed. An engine sound-mimicking system based on a sine wave that can truly mimic the sound of a specific engine was proposed in [5]. The models in prior studies are all based on spectrum analyzer technology of sound signals from specific internal combustion engines, and a vector signal processing algorithm is used for adaptive sound mimicking, but there are three common problems:

(1) The vectorization algorithm of sound signals divides the raw sound into audio frames and solves the mapping relationship between corresponding audio frames. The sound generated by these methods is not the same as real engine noise to human ears, leading to the problem of discontinuity and poor authenticity.

(2) Without considering the overall dynamic characteristics of electric automobiles, the sound mimicking is poor when the engine parameters fluctuate, resulting in poor authenticity.

(3) Consumers with different driving experience and of different ages, genders, and occupational backgrounds have diversified demands for interior sounds of electric vehicles. The mimicked sound is

based on the characteristics of sound signals of specific internal combustion engines, and the diverse needs of different kinds of internal combustion engines in terms of sound mimicking cannot be satisfied.

Generative adversarial networks (GANs) have been used in image generation, semantic segmentation, and speech generation. Goodfellow and Pouget-Abadie [6] proposed an image generation algorithm based on GAN, and training datasets such as the Mixed National Institute of Standards and Technology (MNIST) database were used to generate images that could be recognized by humans. Jin et al. [7] used a GAN to remove rain stripes from images. Donahue et al. [8] proposed the WaveGAN approach to generate audio signals, where a GAN was used for unsupervised synthesis of raw audio waveforms to generate a drumbeat and sounds made by birds. The use of GANs for active sound production in electric automobiles has not been studied.

In this study, two GAN models were trained with raw sound signals and processed frequency domain signals from specific internal combustion engines in different conditions. An experiment showed that the reproduced sound of an internal combustion engine has a high similarity to the real sound.

2. Design Principle

2.1. GAN. As a deep learning model, a GAN can be trained using a discriminator and generator [9]. The generator is responsible for generating samples and sending them with the real samples to the discriminator for training, aiming to select the optimal generated sample with the maximum probability. Through training against real and generated samples, the discriminator identifies real samples and refuses generated samples as much as possible. This principle is shown in Figure 1.

The most direct model of a GAN is the multilayer perceptron, which learns mapping from a low-dimensional potential vector $z \in Z$ (a priori variable of independent samples with the same distribution) to the midpoint in the real data χ. Goodfellow and Pouget-Abadie [6] proposed that the loss function of the discriminator D is actually the regular cross-entropy loss function related to the following binary classifier:

$$\text{loss}_D = -(y \log(p) + (1 - y)\log(1 - p)). \quad (1)$$

The results of the loss function were different depending on the input sample types. When one or the other term in the loss function is approaching 0, the result will be the negative logarithm of the probability that the discriminator predicts the sample to be correctly classified. Note that $y = 1$ for real samples. The quantities p and $1 - p$ are the respective prediction probabilities of the real and false samples. If $D(x)$ denotes p and (z) stands for the generated sample x, then the loss function of the discriminant model can be written as follows [6]:

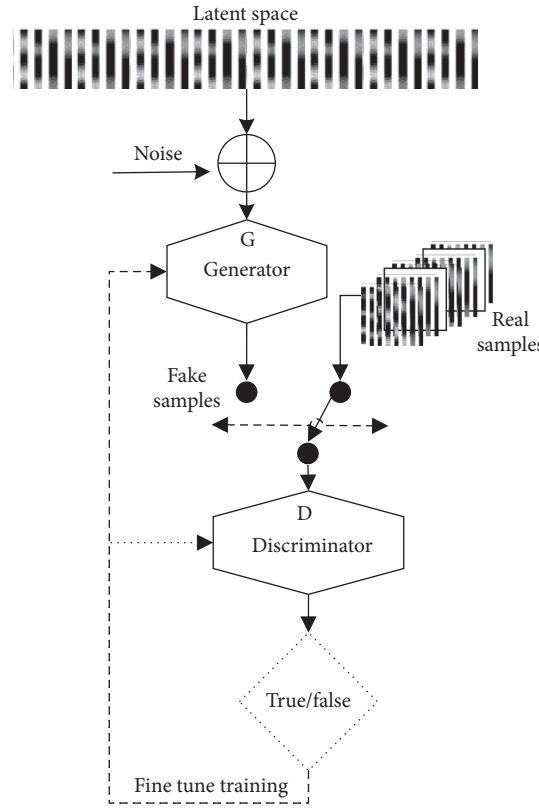

Figure 1: GAN training process.

$$\text{loss}_D = -(y \log(D(x)) + (1 - y)\log(1 - D(G(z)))). \quad (2)$$

The generator G is designed to maximize the loss function of the discriminator D [6]. Since $y \log(D(x))$ has nothing to do with the generator, the loss function of G can be expressed as

$$\text{loss}_G = (1 - y)\log(1 - D(G(z))). \quad (3)$$

There is a metacompetitive relationship between the generator G: Z↦X and D: X↦[0, 1], with the objective function [9]:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim P_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim P_z(z)}[\log(1 - D(G(z)))], \quad (4)$$

where x is the real data with probability distribution $p_{\text{data}}(x)$, z is the noise data with probability distribution $p(z)$, and $\mathbb{E}$ is the mathematical expectation of real data x and noise data z. Often represented by a fully connected or convolutional neural network, the generator G can obtain a distribution $p_g(x)$ of generated data through the noise distribution $p(z)$, aiming to make $p_g(x)$ as close to $p_{\text{data}}(x)$ as possible, i.e., to ensure a minimum number of generated samples are judged as false. The discriminator D is trained to maximize the discriminant generation of false samples, i.e., to measure the gap between $p_g(x)$ and $p_{\text{data}}(x)$. Equation (1) tries to locate the minimum Jensen–Shannon divergence between $p_{\text{data}}(x)$ and P_G, where P_G is the generator's implicit distribution of $z \sim P_z(z)$.

2.2. Solution to GAN. The objective function [9] V in equation (1) is continuous. With the mathematical expectation expressed in the integral form of V, we obtain

$$V(D,G) = \int_{-x}^{x} p_{\text{data}}(x)[\log D(x)]\mathrm{d}x + \int_{-z}^{z} p_z(z)[\log(1 - D(G(z)))]\mathrm{d}z. \tag{5}$$

If the data generated by $G(z)$ are x, then we obtain

$$V(D,G) = \int_{-x}^{x} p_{\text{data}}(x)[\log D(x)]\mathrm{d}x + \int_{-z}^{z} p_z\left(G^{-1}(x)\right)[\log(1 - D(x))]\left(G^{-1}\right)'(x)\mathrm{d}x. \tag{6}$$

$p_g(x)$ is defined as the generation distribution of z:

$$p_g(x) = p_z\left(\left(G^{-1}(x)\right)\left(G^{-1}\right)\right)'(x). \tag{7}$$

Substituting equation (7) into (6) yields equation (8):

$$V(D,G) = \int_{-x}^{x} p_{\text{data}}(x)[\log D(x)]\mathrm{d}x + \int_{-x}^{x} p_g(x)[\log(1 - D(x))]\mathrm{d}x. \tag{8}$$

The maximum value of D in the objective function [9] is

$$\frac{\partial}{\partial D(x)}\left(p_{\text{data}}(x)[\log D(x)]\mathrm{d}x + p_g(x)[\log(1 - D(x))]\right) = \frac{p_{\text{data}}(x)}{D(x)} - \frac{p_g(x)}{1 - D(x)} = 0 \Rightarrow D^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}, \tag{9}$$

which is the optimal solution of $D(x)$. If the distribution of samples generated by G is consistent with the distribution of real samples, i.e., $p_{\text{data}}(x) = p_g(x)$, then $D^*(x) = 1/2$. By substituting the optimal solution of D in equation (8) to solve for the optimal value of G, we obtain

$$C(G) = \int_{-x}^{x} p_{\text{data}}(x)\log\left(\frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}\right) + p_g(x)\log\left(\frac{p_g(x)}{p_{\text{data}}(x) + p_g(x)}\right)\mathrm{d}x. \tag{10}$$

We turn equation (10) into the KL divergence expression to obtain Theorem 2.5 of [10]:

$$C(G) = \mathrm{KL}\left[p_{\text{data}}(x)\middle\|\frac{p_{\text{data}}(x) + p_g(x)}{2}\right] + \mathrm{KL}\left[p_g(x)\middle\|\frac{p_{\text{data}}(x) + p_g(x)}{2}\right] - \log 4. \tag{11}$$

In equation (11) [10], when the objective of the optimal solution of G, $p_g(x)$, is equal to the real distribution $p_{\text{data}}(x)$, KL = 0 and the minimum value of G is –log4. Hence, when the discriminator approaches the optimal solution, G also approaches the minimum value.

3. Design of the Active Sound Production System

We propose a GAN model to generate the sound from an internal combustion engine. The main process is as follows. The preprocessed audio samples and tags are divided into training and test sets. The GAN model is trained with the training set data and the corresponding tags as the raw audio input. The test set is used to validate the iterated training model, so as to filter and save the optimal model. The generator model in the saved GAN model is used to generate new audio samples (Figure 2).

3.1. Design of the GAN Model. The generator and discriminator have a convolutional neural network structure, and the convolutional layer of the generator is referred to as transposed convolutional layer [11], i.e., the feature map is upsampled, which is similar to a reverse gradient calculation in an ordinary convolutional layer. The GAN structure to train with raw audio samples is as follows. The generator consists of an input layer, a fully connected layer, and five convolutional layers. In the GAN model, if the number of convolutional layers of the generator and the discriminator is larger, the generated sample is closer to the real sample, and at the same time, the training time of the model is also longer. The generator structure setting 5 convolutional layers is an optimized selection result, which takes into account the training time and the quality of the generated samples [8]. The ReLU activation function is used between two convolutional layers. Since the input raw audio samples are one-dimensional vectors, only the width of the convolution kernel is denoted (Table 1).

The raw sample discriminator consists of five convolutional layers and one fully connected output layer. The ReLU function serves as the activation function between two layers, a phase conversion operation is added to each convolutional layer, and the discriminator contains a reconstruction layer and a fully connected layer (Table 2).

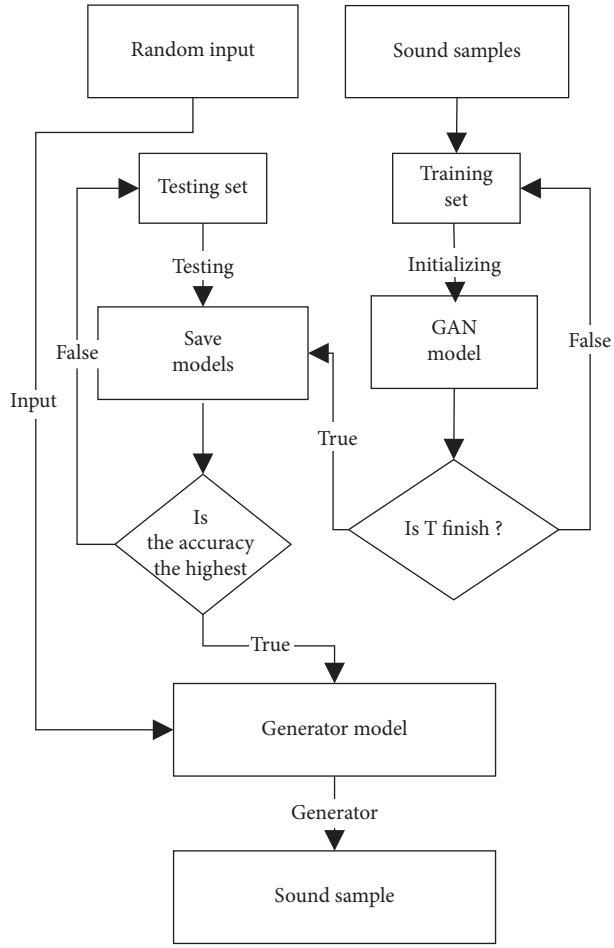

Figure 2: Active sound production process.

A GAN model using spectrograms of samples has the following network structure. The generator consists of one fully connected layer and five convolutional layers. The ReLU function serves as the activation function between two layers, and the last layer is activated with a $\tan h$ function. Since the input spectrograms are two-dimensional vectors, the size of the convolution kernel is represented by length and width (Table 3).

The discriminator consists of five convolutional layers, one reconstruction layer, and one output layer. Activation layers alternate between convolutional layers. The ReLU function serves as the activation function. The output layer is fully connected. Table 4 shows the structure.

4. Sample Generation

4.1. Processing of Audio Samples. Principal component analysis indicates that the audio samples are periodic. Therefore, long audio signal samples are decomposed into corresponding frequency bands during processing. Lee et al. [12] and Sainath et al. [13] proposed using audio signals in the time domain to complete training in semisupervised audio classification, and their results show that classification can be performed in the time or frequency domain with the same accuracy. The sound signal produced when the engine starts is nonstationary. It varies in time and may contain the audio characteristics of the motor and engine speed. Therefore, in short-term processing of audio samples, the two-dimensional time-domain spectrogram signals and audio signals in the time domain are involved in a comparative experiment.

The working audio samples of the internal combustion engine were recorded in an ordinary laboratory environment. A total of 1200 audio samples were collected in the morning and afternoon to form an experimental sample library. This contained 400 sets of steady-state sound signals produced by a Toyota HR16DE gasoline engine, Hyundai D4BH diesel engine, and Mitsubishi 4G6 MIVEC gasoline engine, whose running speed increased from 800 rpm to 4500 rpm. Each audio sample was processed to be monophonic with a sampling rate of 16 kHz and a duration of 1 s.

When processing audio samples in the time domain, a real signal waveform was converted to a one-dimensional

Table 1: Generator structure of the raw audio GAN model.

Layer	Type	Length and width of convolution kernels	Number of convolution kernels
Input layer	—	—	—
Fully connected layer	Fully connected layer	1*100	16384
ReLU	Activation layer	—	—
Conv1D (stride = 4)	Convolutional layer	1*25	512
ReLU	Activation layer	1*25	
Conv1D (stride = 4)	Convolutional layer	1*25	256
ReLU	Activation layer	—	—
Conv1D (stride = 4)	Convolutional layer	1*25	128
ReLU	Activation layer	—	—
Conv1D (stride = 4)	Convolutional layer	1*25	64
ReLU	Activation layer	—	—
Conv1D (stride = 4)	Convolutional layer	1*25	1
ReLU	Activation layer	—	—

Table 2: Discriminator structure of the raw audio GAN model.

Layer	Type	Length and width of convolution kernels	Number of convolution kernels
Input layer $G(z)$	Input layer	—	—
Conv1D (stride = 4)	Convolutional layer	1*25	64
LReLU ($\alpha = 0.2$)	Activation layer	—	—
Phase conversion	Phase conversion	—	—
Conv1D (stride = 4)	Convolutional layer	1*25	128
LReLU ($\alpha = 0.2$)	Activation layer	—	—
Phase conversion ($n = 2$)	Phase conversion	—	—
Conv1D (stride = 4)	Convolutional layer	1*25	256
LReLU ($\alpha = 0.2$)	Activation layer	—	—
Phase conversion ($n = 2$)	Phase conversion	—	—
Conv1D (stride = 4)	Convolutional layer	1*25	512
LReLU ($\alpha = 0.2$)	Activation layer	—	—
Phase conversion ($n = 2$)	Phase conversion	—	—
Conv1D (stride = 4)	Convolutional layer	1*25	1024
LReLU ($\alpha = 0.2$)	Activation layer	—	—
Reshape	Reconstruction layer	—	—
Fully connected layer	Fully connected layer	16384*1	64

Table 3: Generator structure of the spectrogram GAN model.

Layer	Type	Length and width of convolution kernels	Number of convolution kernels
Input layer	Input layer	—	—
Fully connected layer	Fully connected layer	1*100	16384
ReLU	Activation layer	—	—
Trans Conv2D (stride = 2)	Convolutional layer	5*5	64*8
ReLU	Activation layer	—	—
Trans Conv2D (stride = 2)	Convolutional layer	5*5	64*4
ReLU	Activation layer	—	—
Trans Conv2D (stride = 2)	Convolutional layer	5*5	64*2
ReLU	Activation layer	—	—
Trans Conv2D (stride = 2)	Convolutional layer	5*5	64*1
ReLU	Activation layer	—	—
Trans Conv2D (stride = 2)	Convolutional layer	5*5	1
Tan$h$	Activation layer	—	—

vector. Extreme value normalization was used to adjust the data, i.e.,

$$X^* = \frac{X}{\mathrm{Max}(X)}, \tag{12}$$

so as to complete the GAN model training using the end-to-end learning mode.

The short-time Fourier transform (STFT) was the most commonly used method to process the spectrogram in the two-dimensional time-frequency domain. This idea has been

Table 4: Discriminator structure of the spectrogram GAN model.

Layer	Type	Length and width of convolution kernels	Number of convolution kernels
$G(z)$	Input layer	—	—
Conv2D (stride = 2)	Convolutional layer	5*5	64
LReLU ($\alpha = 0.2$)	Activation layer	—	—
Conv2D (stride = 2)	Convolutional layer	5*5	128
LReLU ($\alpha = 0.2$)	Activation layer	—	—
Conv2D (stride = 2)	Convolutional layer	5*5	256
LReLU ($\alpha = 0.2$)	Activation layer	—	—
Conv2D (stride = 2)	Convolutional layer	5*5	512
LReLU ($\alpha = 0.2$)	Activation layer	—	—
Conv2D (stride = 2)	Convolutional layer	5*5	1024
LReLU ($\alpha = 0.2$)	Activation layer	—	—
Reshape	Adjustment layer	—	—
Fully connected layer	Fully connected layer	16384*1	64

adapted by multiple researchers to allow for an objective measure of various generative systems [14–16]:

$$\mathrm{STFT}_x(t, f) = \int_{-\infty}^{+\infty} x(u) w^*(u - t) e^{-j2\pi f u} \, \mathrm{d}u, \tag{13}$$

where $x(t)$ is a continuous signal, $w^*(t)$ is a window function that varies with time, and the superscript " $*$ " denotes the complex conjugate. Equation (13) was used to transform the audio signal from the time domain into the frequency domain in discrete time bins [14]. This provides the amplitude and phase of each frequency contained in the signal at any time. The $w(t)$ window function is

$$\begin{aligned} w(n) &= \frac{1}{2}\left[1 - \cos\left(\frac{2\pi n}{N-1}\right)\right] R_N(n), \\ R_N(n) &= \begin{cases} 1, & 0 \le n \le N-1, \\ 0, & \text{other.} \end{cases} \end{aligned} \tag{14}$$

When preprocessing the signals, the frame was divided by windowing at an interval of 16 ms with 8 ms. This process yields data in a 129×1999 matrix. In order to ensure that the obtained sound spectrum can reflect human auditory sensitivity, a mel-scale filter was used to process the two-dimensional time-frequency signals:

$$\mathrm{mel}(f) = 2595 * \log_{10}\left(1 + \frac{f}{700}\right). \tag{15}$$

The resulting signals effectively reflect the sensitivity of the human ear, i.e., the mel spectrum changes rapidly at low frequencies and slowly at high frequencies. Figure 3 shows the processed mel spectrogram.

The sample data were translated and normalized as follows:

$$X^* = \frac{X - \mu}{\sigma}. \tag{16}$$

In other words, the mean value of the dataset was set to 0, the standard deviation was set to 1, and the values of the processed dataset ranged from −1 to 1. In equation (16), μ is the mean value of all samples, and σ is the variance. A human ear auditory test on the processed data indicated no auditory difference between the samples.

4.2. Training and Optimization. To guarantee the comparability of the test results from the two signal inputs, the inputs from raw audio samples and spectrograms both lasted 1 s, and the data input to the generator consisted of 100-dimensional potential vectors. The generator trained against the raw samples produced the generator structure in Table 1. The fully connected layer converted the 100-dimensional noise vectors into a 16×1024 feature map. A deconvolution operation similar to upsampling was conducted based on the length and number of convolution kernels in Table 1. After five deconvolution and activation instances, a 16384-dimensional vector was obtained and input to the discriminator in the next step. The discriminant structure shown in Table 2 was used to train the discriminator against the raw samples. The 16384-dimensional vector generated by the generator and the 16384-dimensional data from reading real samples were input to the discriminator for convolution and corresponding activation operations. The LReLU function [15] served as the activation function, which reduced the sparsity of ordinary ReLU. We set $\alpha = 0.2$ in this study. After five convolution operations, the fully connected layer and discriminator were connected to determine the authenticity of the samples.

While training the raw sample discriminator, due to common frequency overlap in the real data, tone noises are inevitably produced during upsampling, which is like the "chessboard" artifact [17] caused by deconvolution of two-dimensional images. Since tone noises often occur at a specific stage, the discriminator will probably learn a rule to reject these noise samples, thereby suppressing the overall optimization and weakening the accuracy of the discriminator. To solve this problem, a phase disturbance operation was used during training, which randomly disturbed the phase of the data in each activation layer through n samples, so that the feature map could be unified before being input to the next layer, thereby decreasing the effect of noise on the discriminator.

The phase disturbance operation produces a uniform sample in each layer of the discriminator by filling the left or

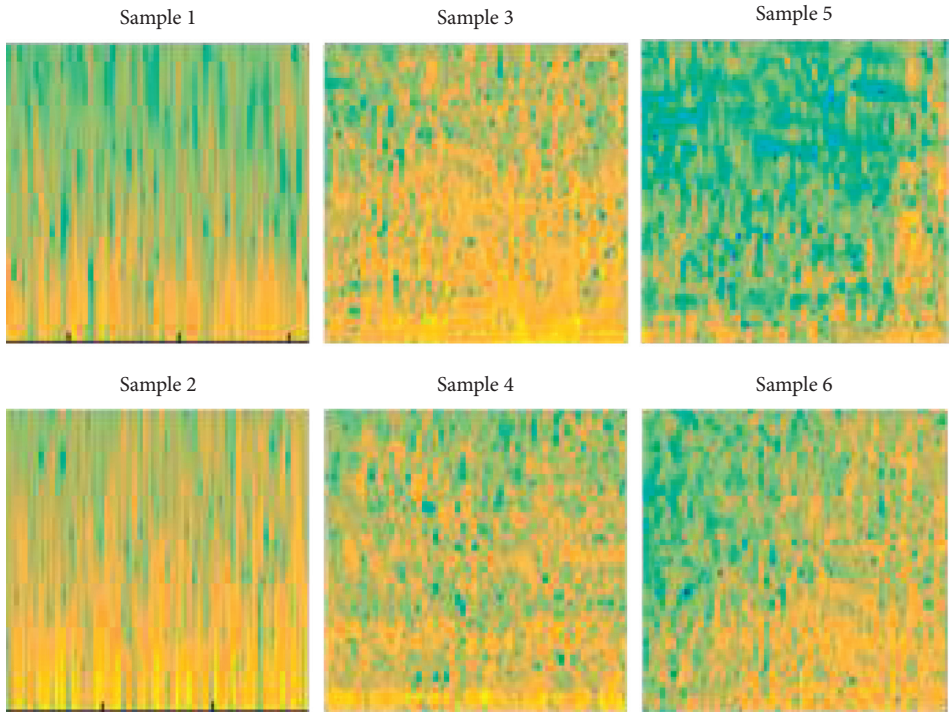

Figure 3: Mel spectrograms of some samples.

right boundary mapping of the feature map into the missing part of the sample after upsampling. Figure 4 shows all the possible outputs from five feature maps when $n = 2$.

In the process of inputting and training against the spectrogram samples, the structure model shown in Table 3 was used for generator training, and the fully connected layer transformed the input 100-dimensional noise vector samples to a $4 \times 4 \times 1024$ feature map. The 5×5 two-dimensional convolution kernel, with the number of convolution kernels decreasing layer-by-layer, was used for deconvolution. After five deconvolution and activation operations, the obtained 128×128 two-dimensional spectrogram was used as the generated sample input to the discriminator. The structure model shown in Table 4 was used for generator training, and the spectrograms generated by the generator and read from the real samples were unified as the input for convolution and corresponding activation operations. The activation function was the same as used for raw audio discriminator training, and the discriminator output was processed in the same way.

To ensure the comparability of the training of the two GAN models, 64 samples in each batch were selected to predict the gradient in two model experiments with a learning rate of 0.0002. To shorten the training time of the sparse gradient problem for convex functions during the training process, the adaptive moment estimation (Adam) algorithm [18] was used to optimize the gradient descent with $\beta = 0.5$.

5. Experimental Analysis and Results

5.1. Experimental Process. The computing environment was an NVIDIA GeForce GTX 1070 GPU and CUDA 9.0 toolkit. Eighty percent of the samples were selected as the training set, 10% as the validation set, and the remaining 10% as the test set. After 80 batches of samples were used for training, the data tended to converge in about nine hours. In Figure 5, G_W and G_S are the generator loss curves for training against the raw audio and spectrogram samples, respectively, and D_W and D_S are the corresponding discriminator loss curves for training. As shown in the figure, when 20 batches of samples were trained, the training loss function values of the two pairs of generators and discriminators tended to stabilize.

5.2. Evaluation and Analysis of Experimental Results. To test the quality of the training model, the samples generated by training were evaluated qualitatively and quantitatively. Humans were asked to perform qualitative evaluation. The sample set for the listening test was divided into two groups: one was composed of 10 audio samples generated by the raw audio GAN model and 10 real sounds, and the other was composed of 10 audio samples generated by the spectrogram GAN model and another 10 real sounds. 25 volunteers randomly recruited on campus participated in the listening test, including 18 boys and 7 girls. The listeners were told in advance that part of audio samples they heard were machine generated. The evaluation was conducted through the online voting system, and each listener voted on the authenticity of samples after listening to them. The degree of authenticity of the samples is divided into 11 levels, with a scale from 0 to 1, where 0 represents a completely generated sound sample, and 1 represents a completely real sound sample. Altogether, 25 people participated in the evaluation, whose results are shown in Figures 6 and 7. Samples with attribute values of 0

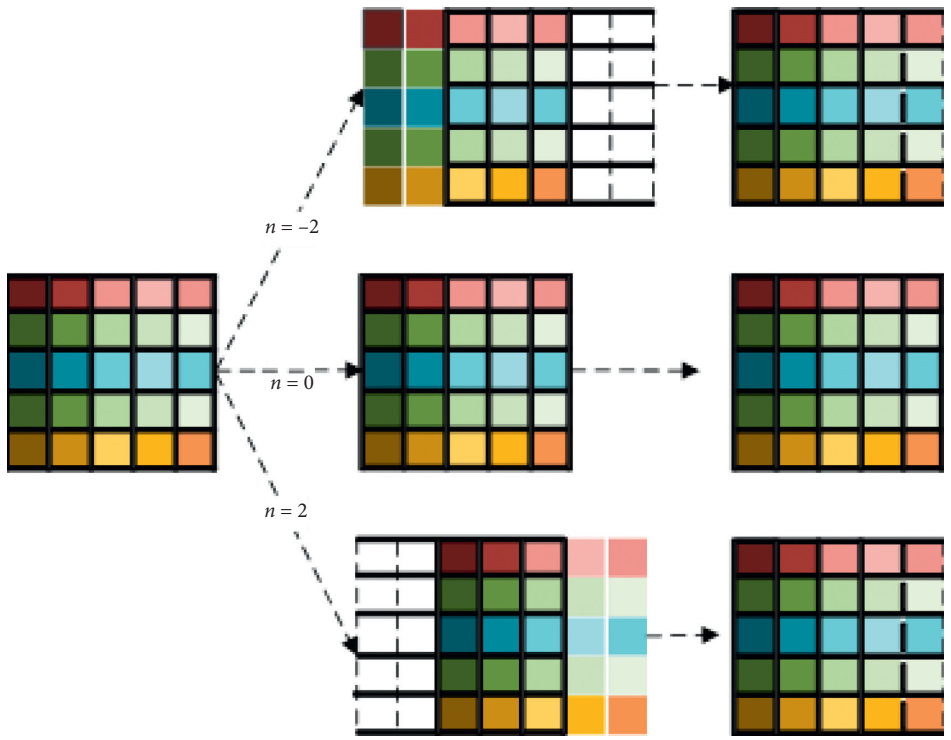

Figure 4: Schematic diagram of phase disturbance operation of feature map.

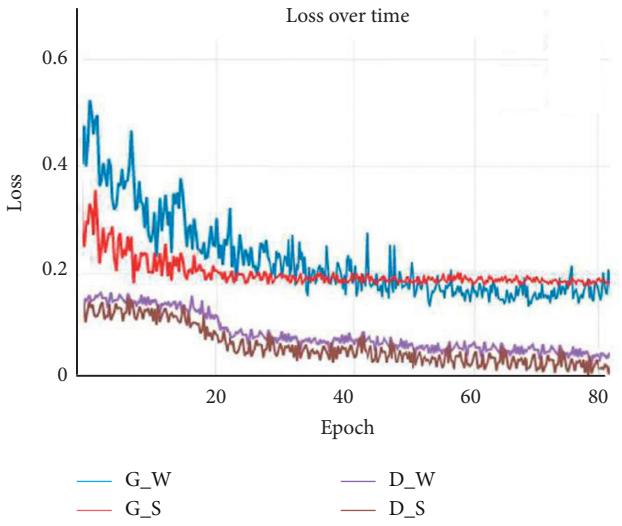

Figure 5: Loss curve.

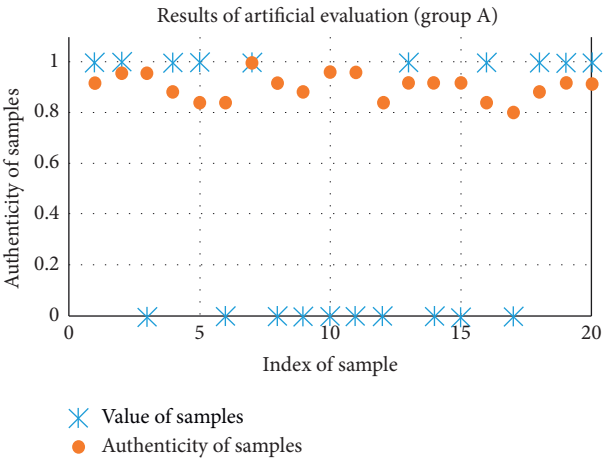

Figure 6: Qualitative evaluation results for Group A.

and 1 identify generated and real samples, respectively. According to the voting results, the authenticity rate of most samples reached more than 90%, which means ordinary people were unable to effectively distinguish the difference between these two types of samples, nor could they evaluate the disparity between the raw audio model and the spectrogram model.

The generated samples were quantitatively evaluated by leave-one-out (LOO), which used a 1-NN classifier between generated and real samples [19]. If the samples generated by the model are qualified and their distribution perfectly matches that of the real samples, then the 1-NN classifier should exhibit an LOO of approximately 50%. No matter how the validation and training sets are allocated, there is only a probability of 50% that the 1-NN classifier could correctly predict the distribution of samples. All 1200 real samples were used as positive samples, and all 1200 generated samples as negative samples. The LOO method was used to conduct circuit training on the 1-NN classifier. As shown in Figure 8, the LOO values for all validation sets with the two models are on an upward curve, indicating great reliability of the model.

Table 5 shows evaluation results for samples generated by the two models. All validation sets have LOO values greater than or close to 50%, which implies there is no overfitting when training the raw audio GAN and the

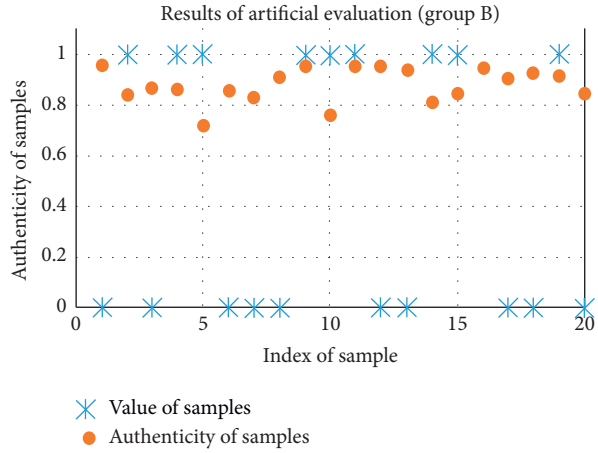

Figure 7: Qualitative evaluation results for Group B.

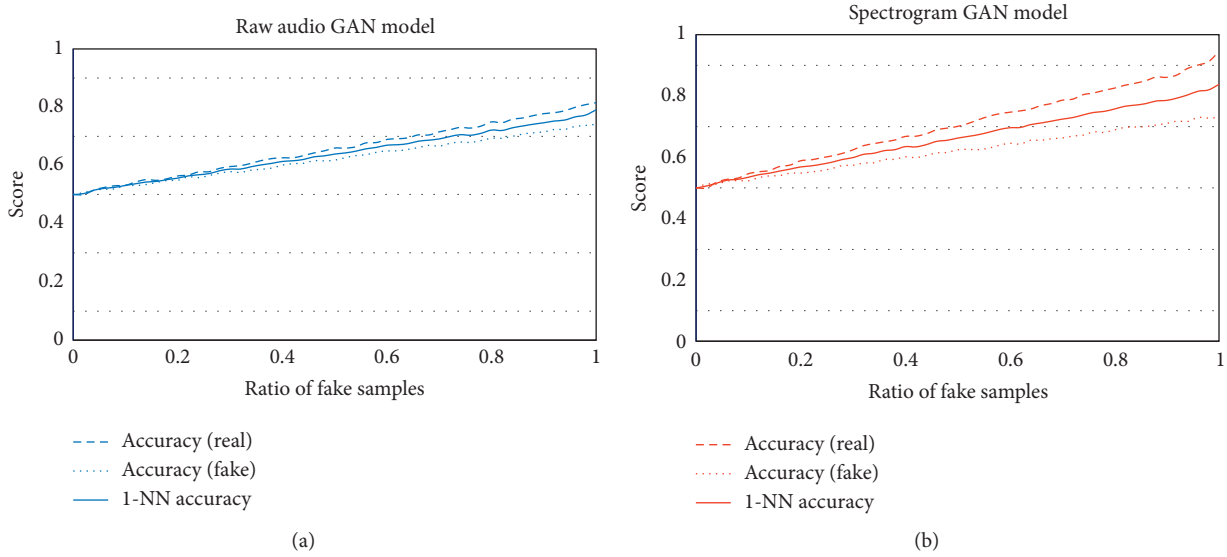

Figure 8: Qualitative evaluation of the GAN model with (a) raw audio and (b) spectrograms.

Table 5: Quantitative evaluation results.

Validation set	Real samples	Raw audio GAN	Spectrogram GAN
Mixed samples	0.485	0.791	0.813
Real samples	0.495	0.735	0.881
Generated samples	0.504	0.814	0.945

spectrogram GAN models. For both models, the validation sets of real samples have an LOO value smaller than that of the validation sets of the generated samples, suggesting that both GAN models can be used to capture the important characteristics of engine sounds from the training distribution. The validation sets of real samples for the two GAN models have relatively low LOO values because the distribution of real samples can usually be captured by the generating model. Consequently, the majority of real samples are surrounded by generated samples, which leads to a comparatively low LOO value for the validation set. The validation set for the generated samples has a relatively high LOO value because the generated samples tend to gather in a small number of pattern centers, and these patterns are surrounded by generated samples in the same category. Hence, when they serve as validation sets, the discriminator will make the correct decision on the negative sample, resulting in a relatively high LOO value. As shown in Table 5, when the samples generated by the GAN model trained against spectrograms serve as the validation set, the LOO value is 0.945, which is 0.131 greater than that of the validation set for samples generated by the GAN model trained against the raw audio signals. This implies that the spectrogram GAN model may have caused mode collapse during

training, thereby failing to fully learn the true distribution of all the samples. The input samples used in the GAN model trained against the raw audio signals are only normalized, and their diversity is higher. Therefore, the samples generated by the raw audio GAN model are also distributed in multiple mode centers, and its model collapse rate is lower than the spectrogram GAN model, which shows that the raw audio GAN model is better than the spectrogram GAN model in the diversity of generated samples. Although human ears are unable to distinguish generated sounds from real sounds, there are insufficient types of generated audio samples by two types of GAN mode which to some extent is correlated with the small number and limited types of training samples. Future studies should increase the number and types of training samples.

6. Conclusions

(1) An active GAN model with corresponding hierarchical structures for the generator and discriminator networks is proposed for producing internal combustion engine sounds in electric automobiles. In experiments, audio samples from internal combustion engines during startup were used as inputs to train a GAN model. Based on the evaluation of the 1-NN classifier, this model can be used to accurately learn the characteristic distribution of the raw audio signals. Human evaluation results show that the generated audio samples closely mimic the real sounds.

(2) Results from LOO tests show that a GAN model trained against raw audio samples exhibited a lower collapse rate than the GAN model trained against spectrograms. Overall the samples generated with the GAN model trained against raw audio samples were of higher diversity than those generated with the GAN trained against spectrograms.

Data Availability

Some or all data, models, or code generated or used during the study are available from the corresponding author by request (list items).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was financially supported by the National Natural Science Foundation of China (U1604141).

References

[1] K. Heather, S. Manabu, S. Todd et al., "Development of approaching vehicle sound for pedestrians (vsp) for quiet electric vehicles," *Sae International Journal of Engines*, vol. 4, no. 1, pp. 1217–1224, 2011.

[2] K. Genuit and W. R. Bray, "Prediction of sound and vibration in a virtual automobile," *Sound and Vibration*, vol. 36, no. 7, pp. 12–19, 2002.

[3] P. Boussard, S. Molla, and F. Orange, "Comprehensive process for car engine sound design: from signal processing to an audio system integrated in the vehicle," in *Proceedings of the 41st International Congress and x Position on Noise Control Engineering*, pp. 9848–9855, New York, NY, USA, August 2012.

[4] J. Jagla, J. Maillard, and N. Martin, "Sample-based engine noise synthesis using an enhanced pitch-synchronous overlap-and-add method," *The Journal of the Acoustical Society of America*, vol. 132, no. 5, pp. 3098–3108, 2012.

[5] D. Min, B. Park, and J. Park, "Artificial engine sound synthesis method for modification of the acoustic characteristics of electric vehicles," *Shock and Vibration*, vol. 2018, pp. 1–8, 2018.

[6] J. I. Goodfellow and J. Pouget-Abadie, "Generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 3, pp. 2672–2680, 2014.

[7] X. Jin, Z. Chen, and W. Li, "AI-GAN: asynchronous interactive generative adversarial network for single image rain removal," *Pattern Recognition*, vol. 100, p. 107143, 2020.

[8] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," 2018, http://arxiv.org/abs/1802.04208.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, and S. Ozair, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, pp. 2672–2680, MIT Press, Cambridge, MA, USA, 2014.

[10] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarialnetworks," *Stat*, vol. 1050, 2017.

[11] C. Guo and J. He, "Improved single shot multibox detector based on the transposed convolution," *Journal of Computer Applications*, vol. 38, no. 10, pp. 2833–2838, 2018.

[12] J. Lee, J. Park, K. L. Kim, and J. Nam, "Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms," 2017, http://arxiv.org/abs/1703.01789.

[13] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform CLDNNs," in *Proceedings of the Sixteenth Annual Conference of the International Speech Communication Association*, Dresden, Germany, September 2015.

[14] K. Liang, H. J. Zhao, and W. Z. Song, "Research on evaluation method of internal combustion engine sound quality based on convolutional neural network," *Civil Engineering Technology*, vol. 40, no. 02, pp. 67–75, 2019.

[15] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, and R. C. Moore, "CNN architectures for large-scale audio classification," in *Proceedings of the 2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pp. 131–135, IEEE, New Orleans, LA, USA, March 2017.

[16] L. Yang and A. Lerch, "On the evaluation of generative models in music," *Neural Computing and Applications*, pp. 1–12, Springer, Berlin, Germany, 2018.

[17] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, vol. 1, no. 10, p. e3, 2016.

[18] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *Computer Science Repository*, vol. 2012, no. 12, pp. 5701–5711, 2012.

[19] Q. Xu, G. Huang, Y. Yuan, C. Guo, Y. Sun, and F. Wu, "An empirical study on evaluation metrics of generative adversarial networks," 2018, http://arxiv.org/abs/1806.07755.