

Research Article

Improvement and Application of Generative Adversarial Networks Algorithm Based on Transfer Learning

Fangming Bi,^{1,2} Zijian Man,^{1,2} Yang Xia ,^{1,2} Wei Liu,^{1,2} Wenjia Yang,^{1,2} Xuanyi Fu,^{1,2} and Lei Gao^{1,2}

¹School of Computer Science and Technology, China University of Mining Technology, Xuzhou 221000, China

²Mine Digitization Engineering Research Center of the Ministry of Education, China University of Mining and Technology, Xuzhou 221116, China

Correspondence should be addressed to Yang Xia; yxia@cumt.edu.cn

Received 19 April 2020; Revised 28 May 2020; Accepted 3 June 2020; Published 13 July 2020

Guest Editor: Chi-Hua Chen

Copyright © 2020 Fangming Bi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Generative adversarial networks are currently used to solve various problems and are one of the most popular models. Generator and discriminator are characteristics of continuous game process in training. While improving the quality of generated pictures, it will also make it difficult for the loss function to be stable, and the training speed will be extremely slow compared with other methods. In addition, since the generative adversarial networks directly learns the data distribution of samples, the model will become uncontrollable and the freedom of the model will become too large when the original data distribution is constantly approximated. A new transfer learning training idea for the unsupervised generation model is proposed based on the generation network. The decoder of trained variational autoencoders is used as the network architecture and parameters to generative adversarial network generator. In addition, the standard normal distribution is obtained by sampling and then input into the model to control the degree of freedom of the model. Finally, we evaluated our method on using the MNIST, CIFAR10, and LSUN datasets. The experiment shows that our proposed method can make the loss function converge as quickly as possible and increase the model accuracy.

1. Introduction

GAN is a new image-generating model based on game theory, which innovatively combines the generative model and adversarial model, and proposes a useful training method based on model features to make the output resulting images clearer and sharper than other methods.

Compared with the previous complex image generation methods, the generation of adversarial network does not need to model its original dataset, but only needs to use generators to approach the original data distribution. Its generator and discriminator also do not need complex network structure, and the original deep neural network can achieve better generation effect. Although researchers have made a lot of improvements to the generation of adversarial network, there are still some points that need to be improved based on its own characteristics. For example, the model

training speed is slow and the model freedom is too large. So, the purpose of our study is speeding up model training and reducing model freedom.

This is a challenging problem because (1) the generation of adversarial network is a process of continuous game between the generator and discriminator during training. While constantly improving the generation of images, it is difficult for the model to reach equilibrium or even stability due to the violent fluctuation of gradient, and the training speed will be too slow. Traditional GAN training on the CPU takes about 3 hours. (2) The generation of adversarial network requires no modeling in advance. Although the process of generating pictures is simplified, the model is too uncontrollable and the freedom of the model is too large.

To counter the problems above, we use the transfer learning method in the unsupervised deep model and combine with other variational autoencoders of the image

generation model to reduce the resource consumption and time of training generative adversarial network. In addition, based on the particularity of the decoder input of the variational autoencoder, the freedom of the model is further limited by limiting the input data distribution of the generator, which provides an improved method for the future training of other unsupervised generator models. In recent years, based on the great potential of generating adversarial network, the method has been widely improved and applied by researchers. Some studies use GAN in semisupervised learning. Finally, some research studies combine GAN with specific applications and achieve good results.

We combine transfer learning and unsupervised learning to make up for the shortcomings of both sides and use standard normally distributed sampling as parameters to increase input of the model. The specific combination and control methods are as follows:

- (1) The decoder of the trained variational autoencoders is used as the network architecture and initial parameters to generative adversarial network generator. The generator is provided with a new starting

point for learning and make the loss function fall down as quickly as possible. Experiments show that the method can reduce the training time by about half.

- (2) Take standard normally distributed sampling as the input data of the generator, and the problem of too much freedom of the model is improved by limiting the input data of the model.

2. Related Works

The idea of generative adversarial networks (GANs) is derived from game theory's Nash equilibrium [1]. The model is shown in Figure 1. GAN can be seen as a collection of two networks: a generator G , which is responsible for fetching data close to real data, and D , a discriminator, determines whether the sample is a real sample or a false sample, t is sampled from the real data, and n is the data generated by the generator. Two networks are trained by gaming each other, and the loss function is

$$\min_G \max_D V(D, G) = E_{t \sim p_{data}(t)}[\log D(t)] + E_{n \sim p_g(n)}[\log(1 - D(G(n)))]. \quad (1)$$

GAN can be found from conferences and journals in recent years [2–8] and is already a hot topic in artificial intelligence [9–11]. It has many papers related to GANs. The original GAN [1] requires no Markov chain or expanded approximate inference network during training. DCGAN uses CNN to replace the basic nerve structure in GAN and uses global pooling layer to reduce the computation. PROGAN starts to train and gradually generates clear images from blurred images [12]. SAGAN uses the weighted sum of the elements of all positions as the response of that location by self-attention and better handles the dependencies between different regions [13]; LP-WGAN [14] replaces Lipschitz [15] to lp-norm to constrain weight search and effectively close the generated distribution and the true distribution. WGAN-GP [16] uses improved gradient penalty to solve the problem of associating parameters with limits to achieve real Lipschitz constraints. The LSGAN changes the target function of the model, more accurately describes the loss of the model and solves the problems of low picture quality and unstable training process [17].

F-GAN proves that any divergence is suitable for generating models [18]. UGAN can remove the source category information retained in the generated image, making the source category of the generated image more difficult to track [19]. LSGAN uses Lipschitz regular conditions to further normalize its loss function to the density of real data [20]. MRGAN takes advantage of the unique geometry of real data, especially manifold information [21]. Geometric GAN [22] uses SVM to separate hyperplanes to maximize margin. RGAN [23] considers that the probability that the real data is true should be reduced because this can reduce the gap between true and false data and make the data smoother.

The autoencoder designs a neural network architecture that imposes bottlenecks on the network and forces the original input to compress knowledge representation. The autoencoder is shown in Figure 2. The autoencoder forces the model to retain only the changes of data required by the reconstruction input, but not the redundancy of input. Therefore, the loss function is referred to as

$$L = \min_{\theta} \|x - f(x)\|^2 + \frac{1}{2} \left(\text{tr} \left(\sum (x) \right) + \mu(x)^T \mu(X) - K - \log \det \left(\sum (x) \right) \right). \quad (2)$$

3. Methods

Transfer learning is to transfer the parameters of the model learned and trained to the new model to help the training of the new model. Most of the data or tasks are correlated.

Through transfer learning, the model parameters can be shared with the new model in some way to speed up and optimize the learning efficiency of the model without learning from zero as most networks do. Transfer learning mainly includes the concept of data domain and task. A data

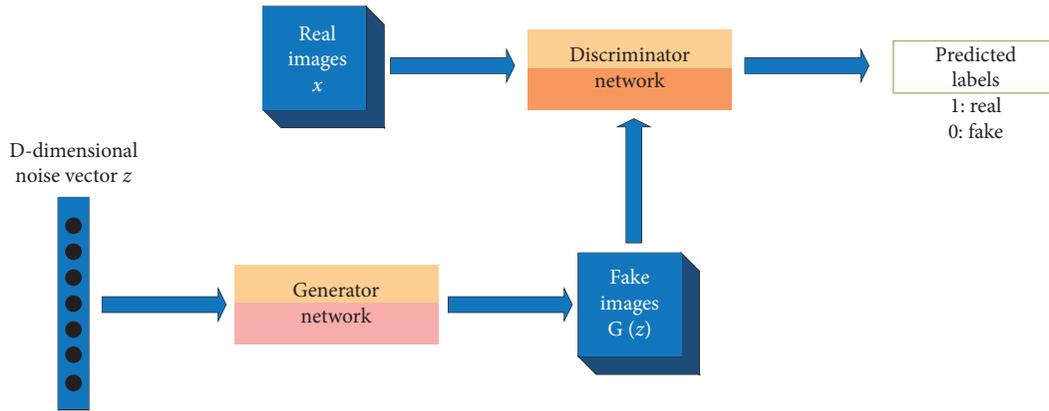


FIGURE 1: Structural figure of generative adversarial networks.

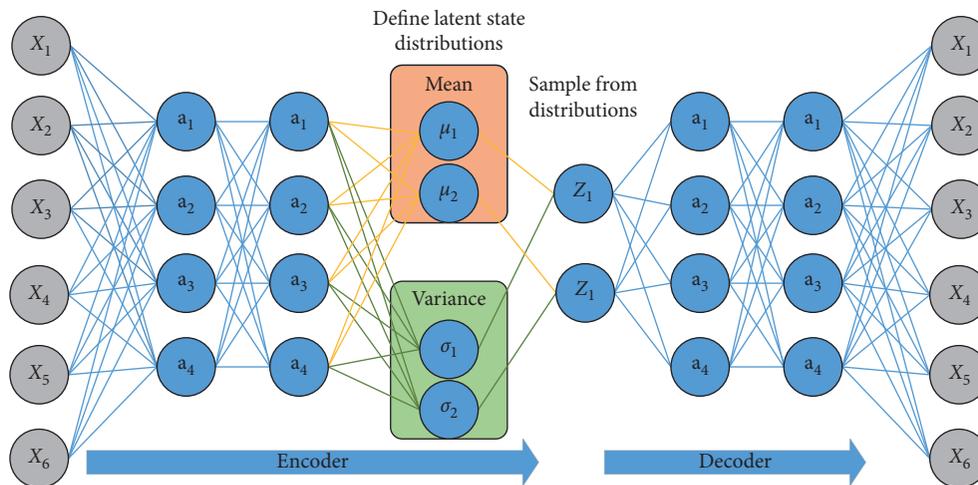


FIGURE 2: Variational autoencoders include the encoder and decoder.

domain D consists of the feature space X and the probability distribution $P(X)$, and $X = x_1, x_2, \dots, x_n$. A task T consists of a label space Y and an objective function F . Different models can be transferred according to the correlation between the data domains and tasks of the source model and the target model in transfer learning.

3.1. *Transfer Learning Theory Based on CNN.* Transfer learning based on convolutional neural network can be divided into two methods.

In the development model, we need to select a relevant predictive modeling problem with rich data first. There is a relationship between the input data of the original task and the target task, the output data, and the concepts learned by adjusting the model parameters with the input data. Then, we need to define an excellent network model for the original task to ensure that the network can enhance the representation of features of original dataset as much as possible. Then, the network model of the original task can be selected as the training starting point of the network model of the new task. Part or all of the original network model can be

selected according to the needs of the new task. Finally, according to the training results of the new model and the needs of the target task, the network model can be adjusted appropriately to make it perform better in the new task.

In the pretraining model, we select the source model first. The pretraining model is different from the development model because the basic model does not need to be trained. Many research institutions have already trained the basic model in the super-large dataset, and the caller only needs to select the network model suitable for the new task. Then, we reuse the model, select the network model of the original task as the training starting point of the network model of the new task, and select part or all of the original network model according to the needs of the new task. Finally, according to the training results of the new model and the needs of the target task, the network model can be adjusted appropriately to make it perform better in the new task.

3.2. *Transfer Learning Based on Unsupervised Generation Model.* Transfer learning is generally applied to the supervised discriminant model. The source model is

Input: noise data $\{n^{(1)}, \dots, n^{(m)}\}$ from standard normal distribution $p_{g(n)}$ and $\{t^{(1)}, \dots, t^{(m)}\}$ from the generated distributed data $p_{\text{data}(t)}$.

Output: The losses of generator and discriminator.

For generative adversarial network training times do

 For k times do

 Select m small batch samples from standard normal distribution $p_{g(n)}$.

 Select m small batch samples from the generated distributed data $p_{\text{data}(t)}$.

 Optimize discriminator weights by stochastic gradient ascent algorithm: $\nabla_{\theta_D} (1/m) \sum_{i=1}^m [\log D(t^{(i)}) + \log(1 - D(G(n^{(i)})))]$

 end For

 Select m random samples from standard normal distribution $p_{g(n)}$.

 Optimize generator weights by random gradient descent algorithm: $\nabla_{\theta_G} (1/m) \sum_{i=1}^m \log(1 - D(G(n^{(i)})))$

end For

ALGORITHM 1: Improved original generative adversarial network Algorithm. D and G are the two parts that make up the GANs. Really sampled data t and the noise data n . $G(n)$ is the sample generated by G that is closest to real data. $D(t)$ is the discriminator of really sampled data, and $D(G(n))$ is the discriminator of fake data from the generator.

transferred to the target model after training through similar datasets, and a new training starting point is given to the target model to accelerate the training speed of the target model. By combining generative models with transfer learning, a new transfer method is presented which can be applied to the unsupervised generation model of deep learning. In this paper, two unsupervised generation models of generation adversarial network and variational autoencoder and their improvement are studied, and MNIST and LSUN datasets are selected as input data. The variational autoencoder is selected as the source model of transfer learning, and the original generative adversarial network and WGAN-GP are selected as the target model. This experiment adopts a new training method of transfer learning based on the unsupervised generation model:

- (1) Select the source model: the variational autoencoder is chosen as the source model of training
- (2) Reuse model: the model parameters of the decoder of the variational autoencoder are saved
- (3) Adjust model: the generative adversarial network is adjusted so that the input of the generative model is obtained from the standard positive attitude distribution, and then the iterative training is carried out according to the generative adversarial network algorithm

The improved original generation adversarial network adopts the small-batch stochastic gradient algorithm. The training times of the discriminator are k , which is a hyperparameter. The dataset is input into the encoder of the variational autoencoder so that the encoder learns mean and variance. The potential variable n is selected from the standard normal distribution and input into the decoder to train the variational autoencoder so that its loss function is stable or even minimal. The network structure and network parameters of the decoder of the variational autoencoder after training are transferred to the generative adversarial network as its initial generator for the following training. The improved original generative adversarial network algorithm is described in Algorithm 1.

4. Experiments

In this experiment, the MNIST, CIFAR10, and LSUN datasets are used to compare different models. The MNIST is a computer vision dataset, which contains 70,000 grayscale images of handwritten digits, each of which contains $28 * 28$ pixels, and each image has a corresponding label that is the corresponding number of images. The CIFAR10 has a total of 60,000 color images, and the image pixels are $32 * 32$, which are divided into 10 categories with 6,000 images in each category. The LSUN mainly includes 10 scene categories and 20 object categories, and each category has about one million pictures. It is a scene understanding image dataset, mainly including bedroom, house, living room, classroom, and other scene images. We improve the generative adversarial network, deep convolutional network, and WGAN-GP, respectively, in three datasets. The details are shown in Table 1. The original generative adversarial network trains the MNIST dataset, and the improved original generative adversarial network adopts the same network structure. The leaky ReLU activation function is used in the original network activation function. Deep convolution generative adversarial network is to change the structure of the original generative adversarial network. The improved deep convolution generative adversarial network adopts the same network structure. In the discriminator of deep convolution generative adversarial network, step convolution is used to replace pooling layer for down sampling, and in the generator, transposed convolution is used to replace upsampling. Removing the full connection layer and replacing the full connection layer with global pooling, the improved WGAN-GP and WGAN-GP adopt the same network structure as deep convolution generative adversarial network, while the WGAN-GP improves the loss function of network to avoid the model collapse.

4.1. Improved Original Generative Adversarial Network.

In this experiment, the low-pixel MNIST dataset is used for training, and the network structure of the generator and discriminator is relatively simple. In order to make a comparison with the original generative adversarial

TABLE 1: The experimental comparison table.

Dataset	Algorithm	Instruction
MNIST	Original generative adversarial network	The experimental results of the original network and the improved network are shown, which show the effectiveness of improved network
	Improved original generative adversarial network	
CIFAR10	Deep convolution generative adversarial network	
	Improved deep convolution generative adversarial network	
LSUN	WGAN-GP	
	Improved WGAN-GP	

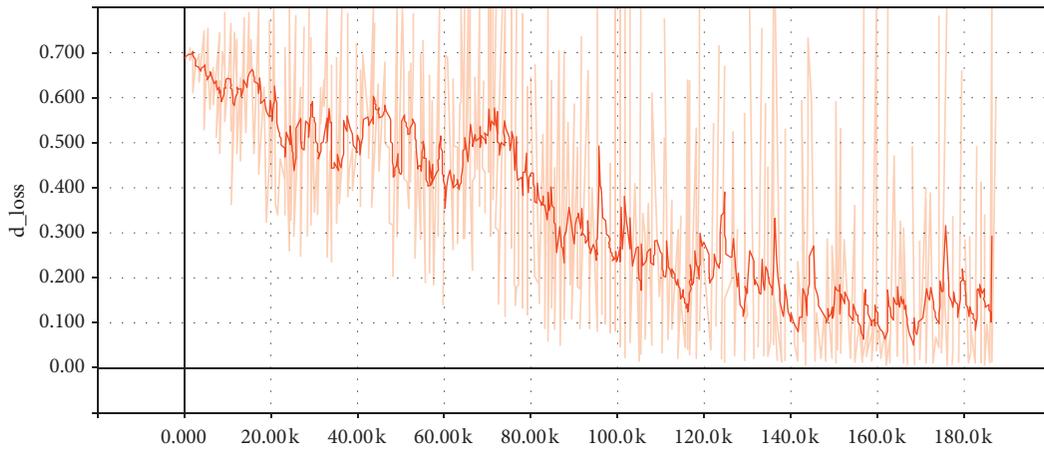


FIGURE 3: The discriminator loss function of generative adversarial networks.

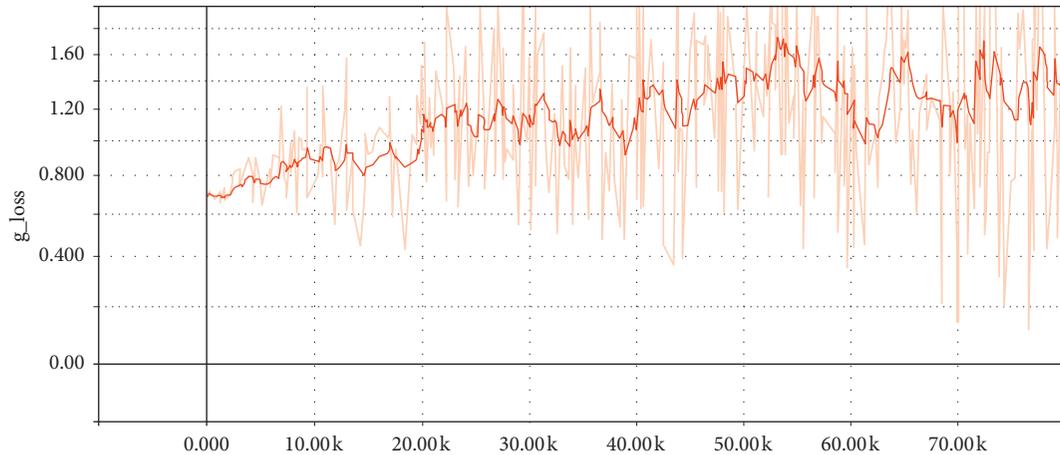


FIGURE 4: The generator loss function of generative adversarial networks.

network, this experiment uses the MNIST dataset to train the two algorithms, respectively. The MNIST dataset generate the training result graph of the adversarial network in the original phase in Figures 3 and 4. The MNIST dataset in the generation of improved counter network training loss in Figures 5 and 6. Figures 7(a) and 7(b) show the generated images, respectively. In the experimental figure, d_loss is the loss value of the discriminator, g_loss is the loss value of the generator, and the x -coordinate is the number of iterations

during training. As the training time increases, the loss of the network gradually decreases and tends to be stable when the training times of the original generative adversarial network reach to 120 k. And generative adversarial network in training reach to 40k, network convergence, generator, and discriminant loss function tends to be stable; therefore, when using a transfer generative adversarial network-based algorithm while the MNIST dataset for network training and greatly improves the convergence speed.

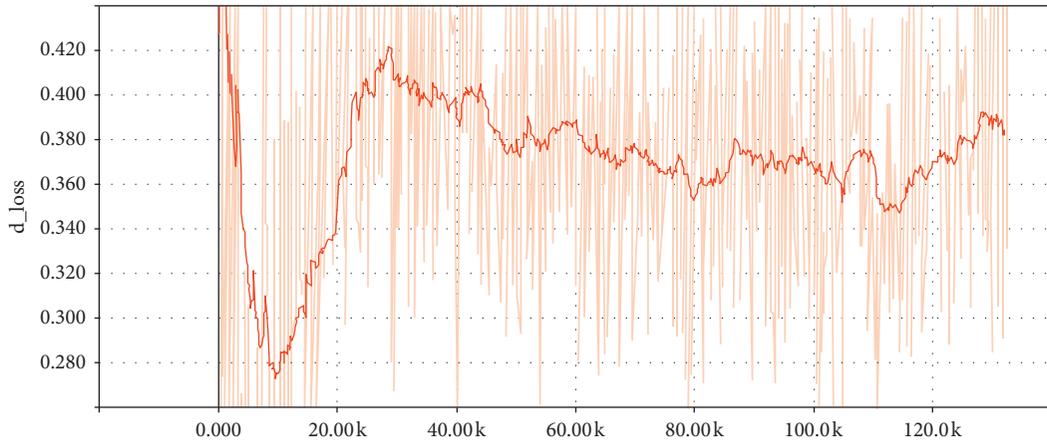


FIGURE 5: The discriminator loss function of improved generative adversarial networks.

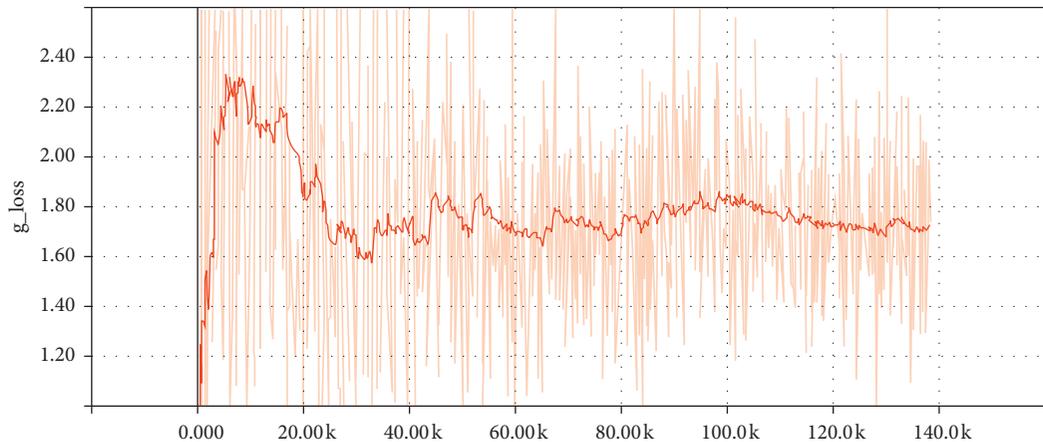


FIGURE 6: The generator loss function of improved generative adversarial networks.

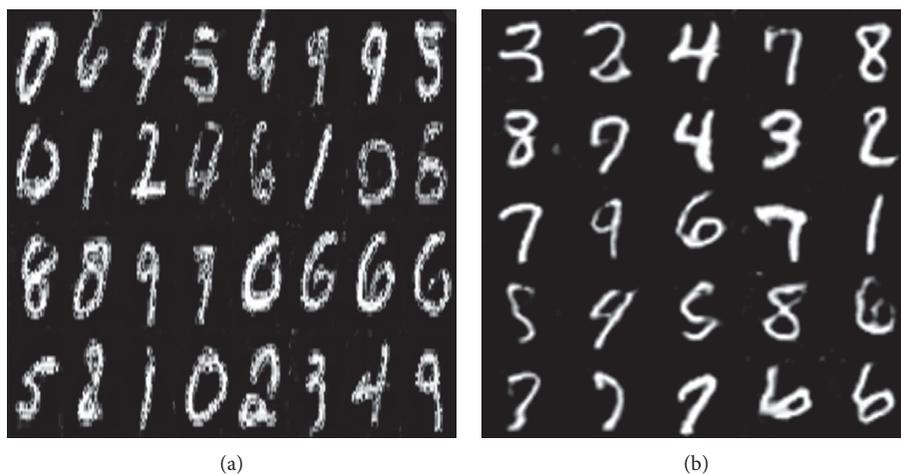


FIGURE 7: (a) Generative adversarial network-generated pictures. (b) Improved generative adversarial network-generated pictures.

4.2. Improved Deep Convolution Generative Adversarial Network. In this experiment, we use the CIFAR10 for training and deep convolution which is adopted by the

generator and discriminator to generative adversarial network. For the purpose of making a comparison with the adversarial network generated by the original deep

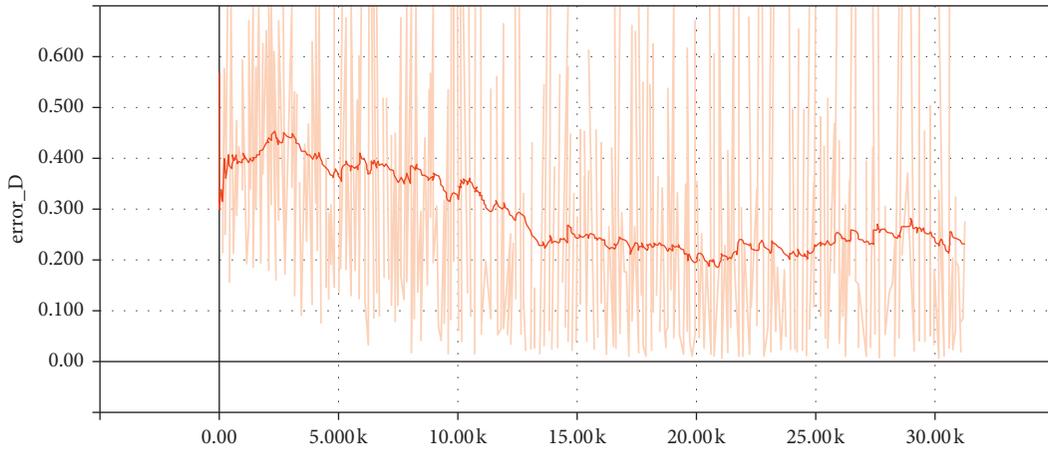


FIGURE 8: The discriminator loss function of deep convolutional generative adversarial networks.

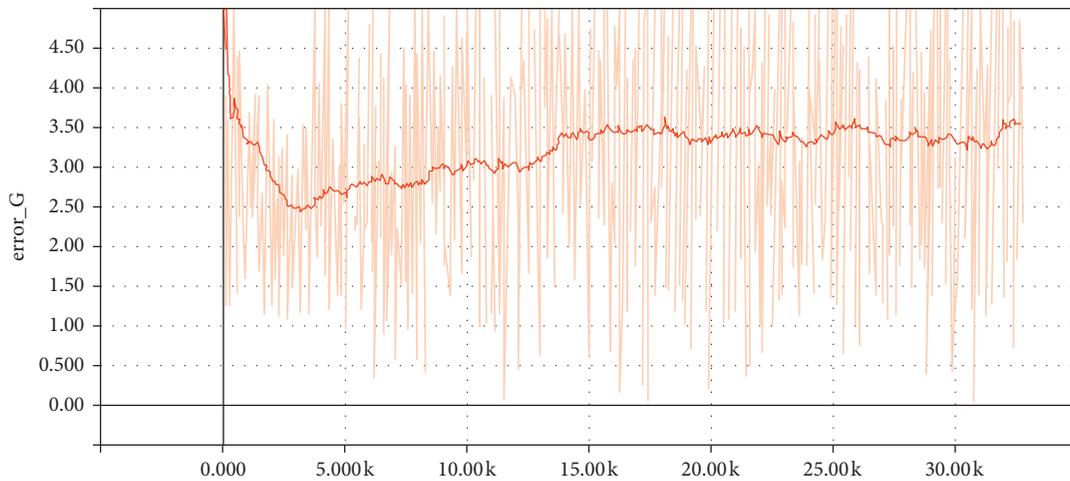


FIGURE 9: The generator loss function of deep convolutional generative adversarial networks.

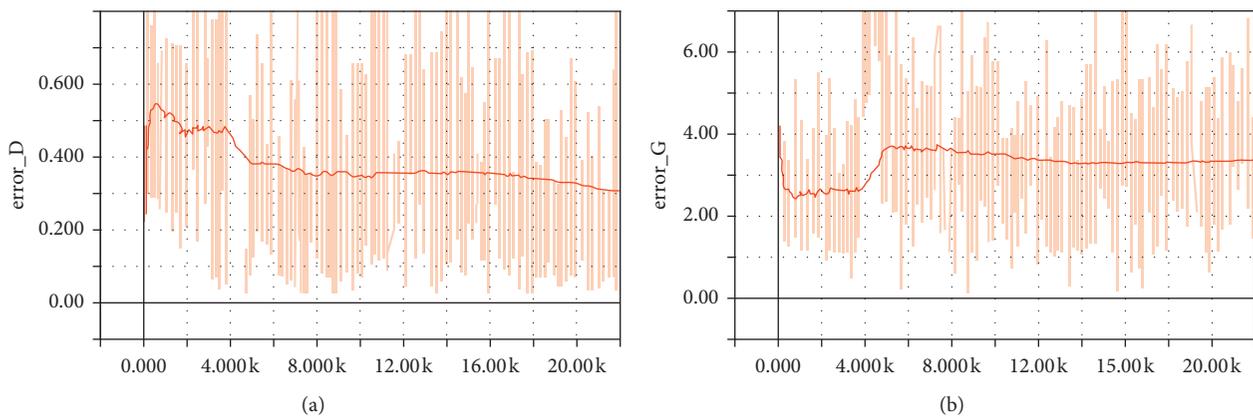


FIGURE 10: (a) The discriminator loss function of improved deep convolutional generative adversarial networks. (b) The generator loss function of improved deep convolutional generative adversarial networks.

convolution, this experiment uses the CIFAR10 dataset to train the two algorithms, respectively. The CIFAR10 dataset generates training result graph of countermeasure network in the original deep convolution in Figures 8 and 9. The

CIFAR10 dataset generates the training result graph of adversarial network in the improved deep convolution in Figures 10(a) and 10(b). Generated images are shown in Figures 11 and 12. In the experimental figure, *error_D* is the

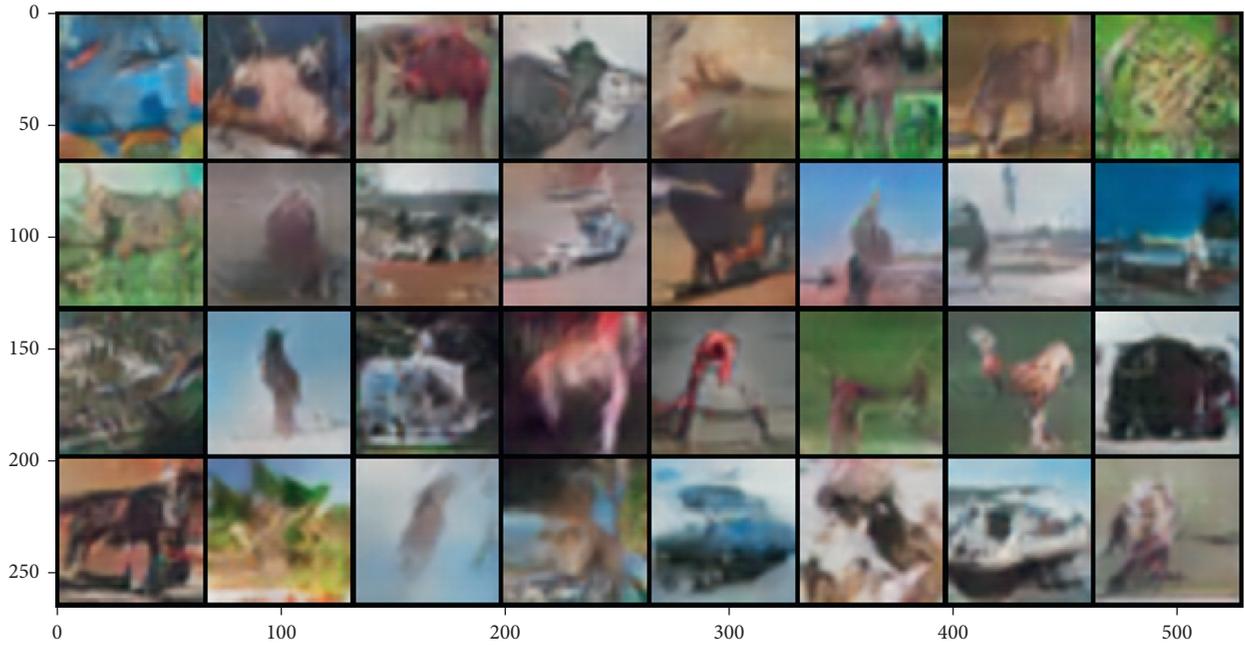


FIGURE 11: Deep convolutional generative adversarial network-generated pictures.

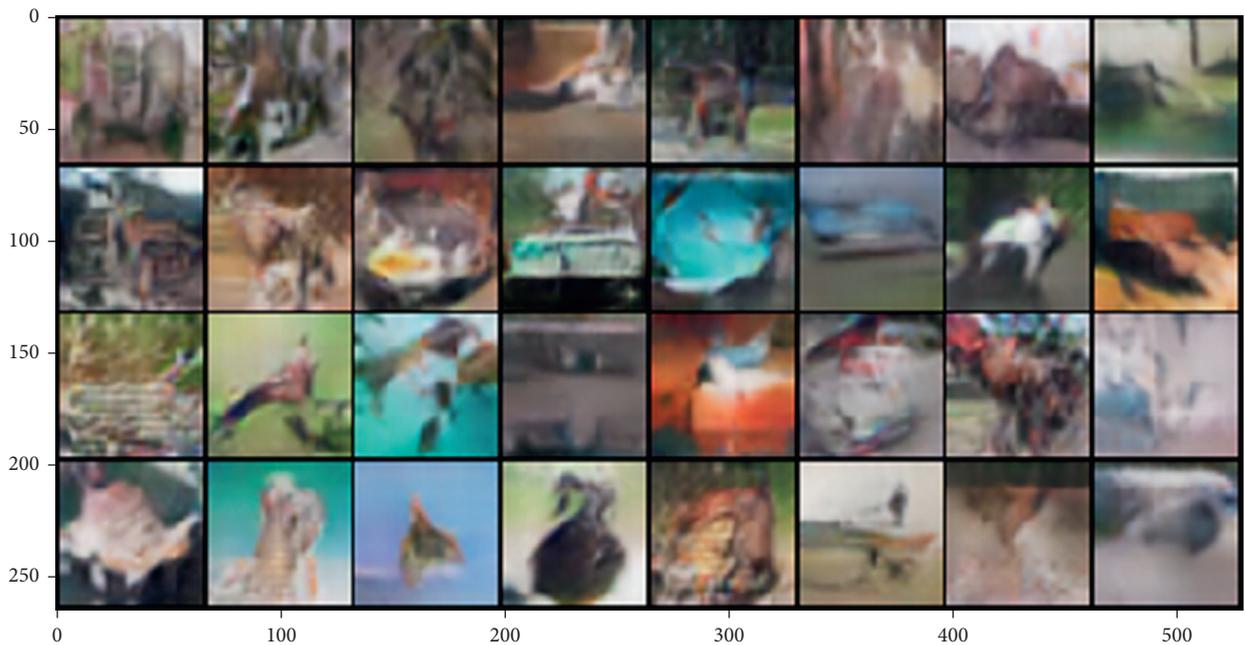


FIGURE 12: Improved deep convolutional generative adversarial network-generated pictures.

loss value of the discriminator, $error_G$ is the loss value of the generator, and the x -coordinate is the number of iterations during training. When the training times of the original deep convolution generation adversarial network reaches 15 k, the network reaches convergence and the loss function of the generator and discriminator tends to be stable and improves the deep convolution generative adversarial network in training reach to 8k, network convergence, generator, and discriminant loss function tends to be stable. Therefore, when deep convolution generative

adversarial network based on transfer learning is used for training in CIFAR10, the convergence speed is improved. In addition, from the result graph of generated images, the quality of generated images is also improved to some extent.

4.3. *Improved WGAN-GP.* In this experiment, the LSUN dataset is used for training, and the generator and discriminator used deep convolution to generate the adversarial network. In order to make a comparison with the original

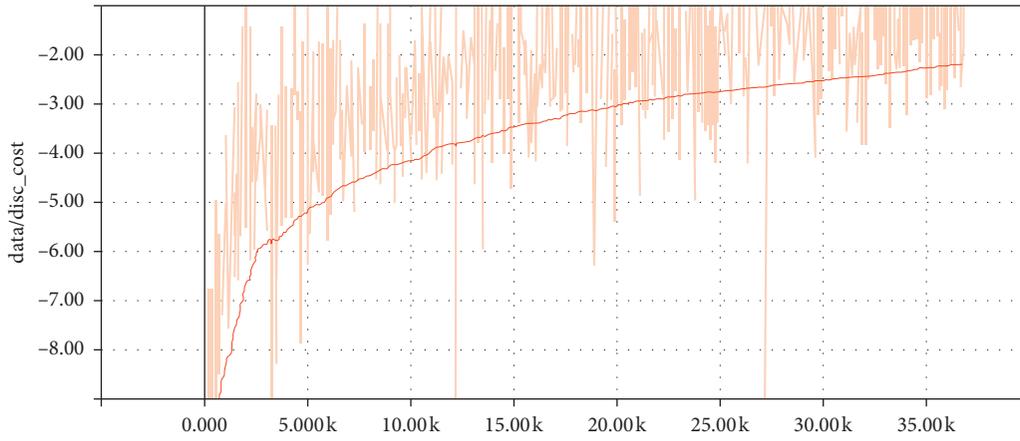


FIGURE 13: WGAN-GP discriminator loss function.

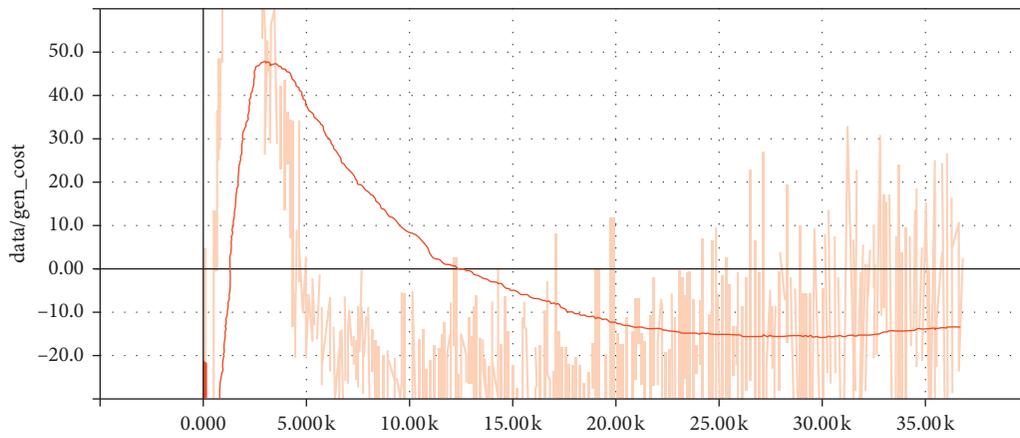


FIGURE 14: WGAN-GP generator loss function.

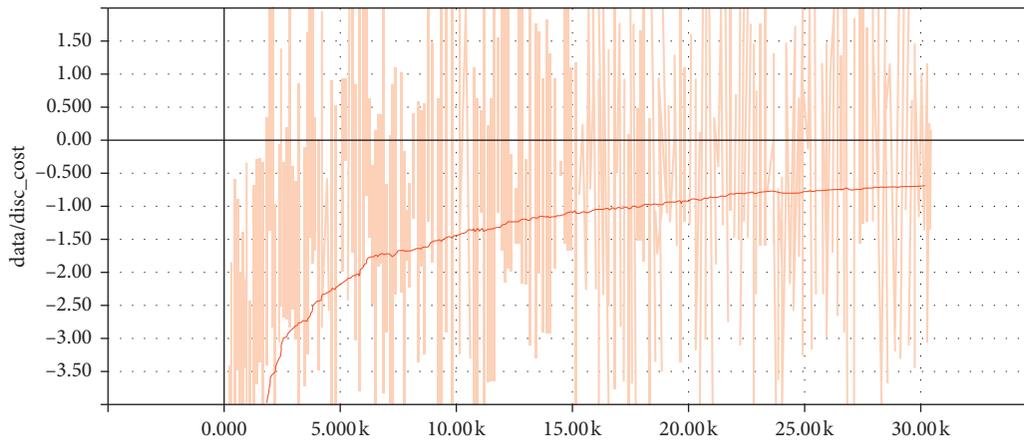


FIGURE 15: Improved WGAN-GP discriminator loss function figure.

WGAN-GP. The LSUN dataset is used for training in the two algorithms in this experiment. The LSUN dataset in the original WGAN-GP training result graphs is shown in Figures 13 and 14. The LSUN dataset in the improved WGAN-GP training result graphs are shown in Figures 15 and 16. Generated images are shown in Figures 17 and 18,

respectively. In the experimental figure, $data/disc_cost$ is the loss value of the discriminator, $data/gen_cost$ is the loss value of the generator, and the x -coordinate is the number of iterations during training. In the loss function graph of the generator and discriminator, when the training frequency of original deep WGAN-GP reaches 25k, the network reaches

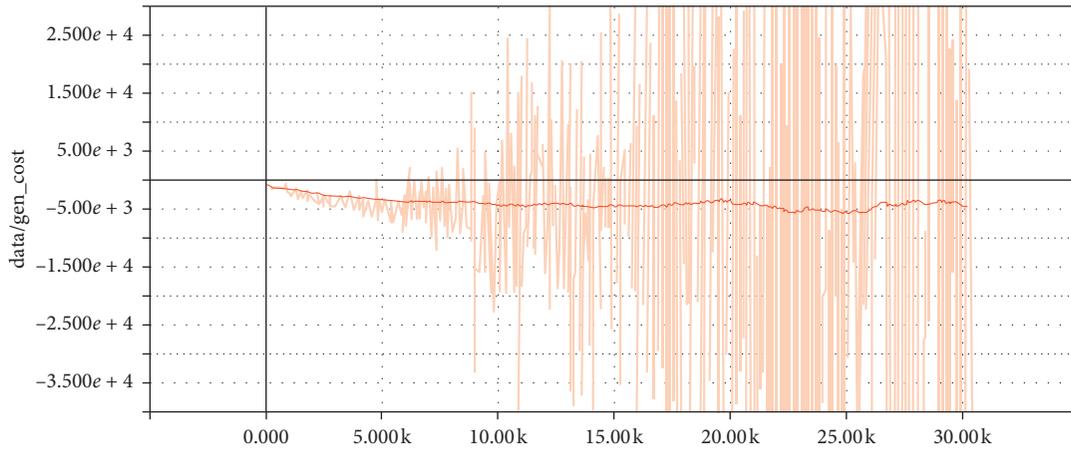


FIGURE 16: Improved WGAN-GP generator loss function.



FIGURE 17: WGAN-GP generated pictures.



FIGURE 18: Improved WGAN-GP generated pictures.

convergence and the loss function of the generator and discriminator tends to be stable. And improved WGAN-GP in training reaches to 20 k, network convergence, generator, and the discriminant loss function tends to be stable; therefore, when using WGAN-GP algorithm based on migration study in LSUN dataset for network training, improve the convergence speed, the other by the result of the generated image, and generated images are also better than the previous images, and images become more acute.

5. Conclusions

In all unsupervised generative models, the generation of adversarial network has the advantages of clearer and

sharper image generation, and does not need to use Markov chain to sample the input dataset repeatedly, and the model is more concise than other unsupervised network models. However, the problems of long training time and too much freedom of the model have not been improved. So, we put forward the transfer method between unsupervised generation models. First, we transfer the decoder of the variational autoencoder to the generator that generates the adversarial network and the convergence speed of the adversarial network is greatly accelerated. Second, our model restricts the generation of the input of the antagonistic network and uses the standard normal distribution, which further limits the freedom of the model and makes it more instructive and closer to the real picture. Experiments show that generated

pictures are more instructive and closer to the real pictures. The image resolution is still the focus of research, so we will design experiment to building a new network that makes the generated images more realistic and speeds up training in our future work.

Data Availability

The MNIST, CIFAR10, and LSUN data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was funded by the Open Program of Shanghai Key Lab of Intelligent Information Processing under Grant no. IIPL-2019-10.

References

- [1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial networks," 2014, <https://arxiv.org/abs/1406.2661>.
- [2] O. Goudet, D. Kalainathan, P. Caillou et al., "Learning functional causal models with generative neural networks," *Explainable and Interpretable Models in Computer Vision and Machine Learning*, pp. 39–80, Springer Link, New York, NY, USA, 2018.
- [3] Y.-Y. Zhang, C.-M. Shen, H. Feng, P. T. Fletcher, and G.-X. Zhang, "Generative adversarial networks with joint distribution moment matching," *Journal of the Operations Research Society of China*, vol. 7, no. 4, pp. 579–597, 2019.
- [4] S. Song, W. Zhang, J. Liu, and T. Mei, "Unsupervised person image generation with semantic parsing transformation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2357–2366, IEEE, Long Beach, CA, USA, June 2019.
- [5] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 2642–2651, Sydney, Australia, July 2017.
- [6] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6721–6729, IEEE, Honolulu, HI, USA, July 2017.
- [7] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *Proceedings of the European Conference on Computer Vision*, pp. 597–613, Springer Link, New York, NY, USA, September 2016.
- [8] C. Lassner, G. Pons-Moll, and P. V. Gehler, "A generative model of people in clothing," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 853–862, IEEE, Venice, Italy, October 2017.
- [9] H. Gan, Z. Luo, M. Meng, Y. Ma, and Q. She, "A risk degree-based safe semi-supervised learning algorithm," *International Journal of Machine Learning & Cybernetics*, vol. 7, no. 1, pp. 85–94, 2016.
- [10] W. Fedus, I. Goodfellow, and A. M. Dai, "MaskGAN: better text generation via filling," 2018, <https://arxiv.org/abs/1801.07736>.
- [11] N. Jetchev, U. Bergmann, and R. Vollgraf, "Texture synthesis with spatial generative adversarial networks," 2016, <https://arxiv.org/abs/1611.08207>.
- [12] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8789–8797, IEEE, Salt Lake City, UT, USA, June 2018.
- [13] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," 2018, <https://arxiv.org/abs/1805.08318>.
- [14] C. Zhou, J. Zhang, and J. Liu, "Lp-WGAN: using Lp-norm normalization to stabilize Wasserstein generative adversarial networks," *Knowledge-Based Systems*, vol. 161, pp. 415–424, 2018.
- [15] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.
- [16] X. Wei, B. Gong, Z. Liu, W. Lu, and L. Wang, "Improving the improved training of wasserstein gans: a consistency term and its dual effect," 2018, <https://arxiv.org/abs/1803.01541>.
- [17] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2794–2802, IEEE, Venice, Italy, October 2017.
- [18] S. Nowozin, B. Cseke, and R. Tomioka, "f-GAN: training generative neural samplers using variational divergence minimization," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 271–279, Barcelona, Spain, 2016.
- [19] Y. Wang, B. Yu, L. Wang et al., "3D conditional generative adversarial networks for high-quality PET image estimation at low dose," *NeuroImage*, vol. 174, pp. 550–562, 2018.
- [20] C. Wu, L. Li, Z. Yang, P. Yan, and J. Jiao, "Image-to-Image local feature translation using double adversarial networks based on CycleGAN," in *Proceedings of the International Conference in Communications, Signal Processing, and Systems*, pp. 907–915, Springer Link, New York, NY, USA, August 2019.
- [21] Y. Li, N. Xiao, and W. Ouyang, "Improved generative adversarial networks with reconstruction loss," *Neurocomputing*, vol. 323, pp. 363–372, 2019.
- [22] J. H. Lim and J. C. Ye, "Geometric GAN," 2017, <https://arxiv.org/abs/1705.02894>.
- [23] A. Jolicœur-Martineau, "The relativistic discriminator: a key element missing from standard GAN," 2018, <https://arxiv.org/abs/1807.00734>.