*Journal of Network Intelligence* ©2024 ISSN 2414-8105 (Online)
**Taiwan Ubiquitous Information** Volume **9**, Number **1**, February **2024**

# Generative Adversarial Networks Based on Cooperative Games

Lie Luo

College of Computer and Information Engineering
Xiamen University of Technology
600 Polytechnic Road, Houxi Town, Xiamen City, Fujian Province, China
luo.lie@foxmail.com

Jiewei Cai

School of Economics and Management
Xiamen University of Technology
600 Polytechnic Road, Houxi Town, Xiamen City, Fujian Province, China
2063568463@qq.com

Zouyang Fan

School of Economics and Management
Xiamen University of Technology
600 Polytechnic Road, Houxi Town, Xiamen City, Fujian Province, China
143631998@qq.com

Yumin Chen

College of Computer and Information Engineering
Xiamen University of Technology
600 Polytechnic Road, Houxi Town, Xiamen City, Fujian Province, China
ymchen@xmut.edu.cn

Hongbo Jiang*

School of Economics and Management
Xiamen University of Technology
600 Polytechnic Road, Houxi Town, Xiamen City, Fujian Province, China
hbjiang@xmut.edu.cn

*Corresponding author: Hongbo Jiang

Received June 14, 2023, revised August 12, 2023, accepted October 16, 2023.

Abstract. *Generative adversarial networks (GANs) have become a hot research topic in recent years, representing an unsupervised learning method based on zero-sum games. Due to the high complexity of real samples, GANs still face challenges in training stability and the quality of generated samples. Mode collapse is a common problem in GANs. To overcome the drawbacks of mode collapse, this paper introduces a novel approach called Cooperative-GAN (Coop-GAN) by incorporating a beating discriminator system instead of the original discriminator model. Cooperative games, which differ from zero-sum games, are employed in Coop-GAN. Unlike traditional GANs' zero-sum games, Coop-GAN incorporates the concept of cooperative games, which involve non-adversarial interactions, distinguishing it from traditional GANs' adversarial nature. In cooperative games, the overall benefits of the game system increase, and all parties can benefit from cooperation, achieving a win-win or mutually beneficial outcome.This collaborative framework forms the foundation of Coop-GAN. In Coop-GAN, the discriminator model learns the real data distribution exclusively by assigning high scores to real samples and refraining from assigning low scores to generated samples. This cooperative mode enables the discriminator model to better guide the generator, enhancing the discriminator's capabilities without increasing the generator's loss. Through mutual cooperation, Coop-GAN achieves an overall gain in the cooperative game, contributing to improved diversity and authenticity of the generated samples. The beating discriminator system in Coop-GAN reduces the frequency of rejecting generated samples, introducing a new cooperative mode where the beating discriminator system and the generator model mutually collaborate and learn together. Through multiple experiments on the Fashion-MNIST dataset, the results demonstrate that Coop-GAN satisfies superadditivity in cooperative games. As the batch size increases, the proposed Coop-GAN model exhibits greater stability compared to traditional GANs. Particularly, when the batch size is 512, Coop-GAN exhibits outstanding performance, reducing the FID value by 10% and increasing the IS score by 4% compared to the multi-discriminator model GMAN.In conclusion, Coop-GAN generates samples with lower FID values and fewer mode collapse phenomena compared to various other GAN models.*

**Keywords:** Cooperative game theory, Generative adversarial networks, Image generation,

## 1. Introduction.

Generative adversarial networks (GAN) [1], proposed by Goodfellow et al. in 2014, are a type of generative model. Inspired by zero-sum games in game theory [2], GAN models consist of a generator and a discriminator, both of which can be implemented using deep neural networks [3].

The generator model in GANs does not directly estimate or fit the distribution of real samples. Instead, it samples data from an implicitly defined distribution [4]. During the training process, the generator model learns to transform a simple input noise distribution (e.g., uniform distribution, multivariate normal distribution) into the target image space distribution. By using this transformation, the generator model generates synthetic samples from the noise distribution. The goal is to generate samples that closely resemble real samples, making it difficult for the discriminator model to distinguish between real and generated samples. The discriminator model aims to learn to distinguish real samples from generated samples, guiding the generator model in generating samples that approximate the real sample distribution [5]. In this way, the generator and discriminator models compete with each other. Over time, the discriminator model becomes better at distinguishing real and generated samples, while the generated samples become increasingly similar to real samples. Overall, both the generator and discriminator models minimize their own losses, and their ultimate goal is to achieve a Nash equilibrium [6].

Mode collapse [7] is a common problem in the process of image generation with generative adversarial networks. It occurs when the generator model produces a single output that the discriminator model consistently classifies as real, even before the discriminator model has learned effective discrimination.

Batch size is an important parameter during training as it can affect the level and speed of model optimization. As batch size gradually increases, training time decreases and computation becomes more stable. However, excessively large batch sizes can lead to a decline in model performance [8]. As shown in Figure 1, when the batch size exceeds a certain threshold, convergence to sharp minima becomes more likely, while smaller batch sizes are more likely to converge to flat minima. Additionally, smaller batch sizes introduce more randomness and exploration during training, which can enhance the network's generalization ability [9]. Training with excessively large batch sizes often leads to mode collapse issues.

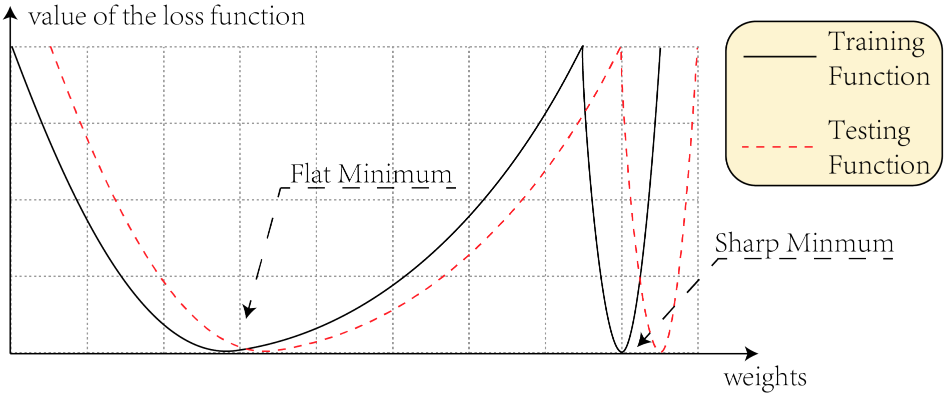


Figure 1. A Conceptual Sketch of Flat and Sharp Minimizers

To address the issue of training instability in GANs, researchers have proposed various improvements to the training process. For example, Topology-aware GAN (TopoGAN) [10] and Focal Frequency Loss GAN (FFLGAN) [11] introduced new loss functions to replace the original GAN's cross-entropy loss. DRAGAN incorporated a novel gradient penalty algorithm during training. In terms of structural improvements, GMAN [12] integrated multiple discriminator models within the GAN framework, while BE-GAN [13] replaced the discriminator with an autoencoder to enhance the final generation results. Most of these modifications focus on the objective function or network structure, rather than specific training strategies.

Traditional GANs employ a zero-sum game strategy, which belongs to non-cooperative games. In a zero-sum game, the gains of one player result in the losses of the other player, and the sum of their gains and losses is always zero. Therefore, the objective of each player is to maximize their own gain by increasing the other player's loss. In contrast, cooperative games [14] are characterized by non-adversarial interactions, where multiple parties can benefit or at least have their interests increased. As a result, the overall benefits of the game system increase.

To address the issue of mode collapse, this paper introduces a novel type of Generative Adversarial Network called Cooperative-GAN (Coop-GAN). Coop-GAN differs from traditional GANs in its adversarial mode by incorporating a new cooperative approach. In this approach, the discriminator model learns the real data distribution exclusively by assigning high scores to real samples and refraining from assigning low scores to generated samples. This cooperative mode allows the discriminator model to better guide the generator, thereby enhancing the discriminator's capabilities without increasing the generator's loss. Through mutual cooperation, Coop-GAN achieves an overall gain in the cooperative game, contributing to improved diversity and authenticity of the generated samples.

Additionally, this paper introduces a beating discriminator system, inspired by the density variation of neurons in the human brain as a person ages [15]. During the training process of Coop-GAN, the size of the beating discriminator system's discriminator model varies periodically. Since different-sized discriminator models fit functions with varying complexities, this design helps to avoid the generator being consistently identified as real, thereby reducing the likelihood of mode collapse.

By incorporating cooperative mode and the beating discriminator system, Coop-GAN exhibits significant differences from traditional GANs in the adversarial generation process and effectively overcomes the problem of mode collapse, leading to improved quality and diversity of generated samples.

The contributions of this paper are as follows: 1. Designed a novel Generative Adversarial Network, Cooperative-GAN (Coop-GAN), based on cooperative game theory, enabling better guidance of the generator by the discriminator model. 2. Introduced the beating discriminator system, inspired by the growth pattern of the human neural system, with periodically varying discriminator model size to prevent mode collapse. 3. Experimental results demonstrate that Coop-GAN shows significant improvements in IS (Inception Score) and FID (Fréchet Inception Distance) values compared to other traditional structured GAN networks.

## 2. **Related Work.**

### 2.1. **Generative adversarial network.**

A Generative Adversarial Network (GAN) consists of a generator model ($G$) and a discriminator model ($D$), and it is an unsupervised learning method. In GANs, the discriminator model is used to distinguish between real and generated samples, while the generator model generates target images and aims to deceive the discriminator model. The payoff for the discriminator model is denoted as $V(\theta^{(G)}, \theta^{(D)})$ and the payoff for the generator model is $-V(\theta^{(G)}, \theta^{(D)})$, Both the generator and discriminator models strive to maximize their own payoff during training, resulting in the following optimization objective:

$$G^* = \arg\min \max_G \max_D V(G, D)$$

Here, $p_{data}(x)$ represents the data distribution of real images, $p_z$represents the input noise variable, and $V(\theta^{(G)}, \theta^{(D)})$is defined as:

$$V(\theta^{(G)}, \theta^{(D)}) = E_{x\sim \mathrm{pant}(x)}[logD(x)] + E_{z\sim p_x(z)}[\log(1 - D\big(G(z)\big))]$$

The structure and update process of a traditional GAN are illustrated in Figure 2. The training of a GAN involves several steps: 1) Fix the parameters of the generator model $G$ and optimize the parameters of the discriminator model $D$ to find that maximizes $V(\theta^{(G)}, \theta^{(D)})$, i.e., $\max\limits_D V(G, D)$ 2) Fix the parameters of the discriminator model $D$ and optimize the parameters of the generator model $G$ to find $G^*$ that minimizes $\max_D V(G, D)$, i.e., $\min_G \max_D V(G, D)$ 3) Repeat steps (1) and (2) in a cycle, alternating between training the generator and discriminator models.

### 2.2. **Progress in adversarial generative networks.**

The training process of GANs is unstable and prone to model collapse. The optimization objective of traditional GANs can be derived as minimizing the Jensen-Shannon (JS) divergence between the generated

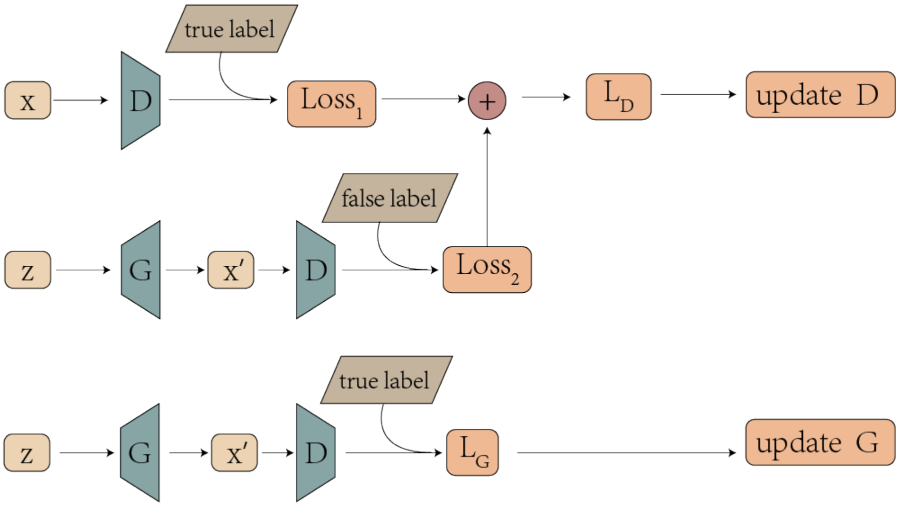


Figure 2. The structure of traditional GAN and the update process

data distribution $p_g$ and the real data distribution $p_{data}$. However, when the JS divergence becomes a constant, the gradient of the generator model becomes zero, causing the network to stop updating [16]. Wasserstein GAN (WGAN) [17, 18] addresses this issue by using the Earth-Mover (EM) distance, which is a more smoothly varying similarity measure compared to the relative JS divergence and Kullback-Leibler (KL) divergence [19]. WGAN-GP [20] further improves upon WGAN by replacing weight clipping with a gradient penalty and introducing Gaussian noise in the generator model. It also uses the Adam optimizer [21] instead of RMSProp to enhance the stability of GAN training. Energy-Based GAN (EB-GAN) treats the discriminator model as an energy function, assigning low energy to real samples and high energy to generated samples during training. Boundary Equilibrium GAN (BE-GAN) introduces the concept of boundary equilibrium and uses an autoencoder as the discriminator. It minimizes the reconstruction error between real and generated samples as the training objective. Least Squares GAN (LSGAN)[22] replaces the original cross-entropy loss function with the least squares loss function. DRAGAN proposes a new gradient penalty algorithm and views the GAN training process as regret minimization rather than traditional consistent minimization. It aims to address mode collapse phenomenon caused by local equilibrium in non-convex situations. GMAN [12] integrates multiple discriminator models in GANs to improve the quality of generated samples. TopoGAN (Topology-aware GAN) algorithm introduced by Wang et al. incorporates new loss functions in the topological feature space. FFLGAN (Focalfrequency loss GAN), proposed by Jiang et al., effectively measures the distribution of generated and real images in the data space. Additionally, DCLAGAN (Dual contrastive loss and attention for GAN) [23] by Yu et al. generalizes image representations and enhances the discriminative power of the discriminator, indirectly improving the image generation capability of the generator.

DCGAN (Deep Convolutional GANs) [24], introduced by Alex Radford et al. in 2016, attempts to understand and visualize the learning process of GANs by incorporating deep convolutional neural networks (CNNs) into GANs. DCGAN (Deep Convolutional GANs) has made the training process of GANs more stable. It introduces convolutional neural networks (CNNs) without pooling layers, which are utilized in both the discriminator and generator models. Furthermore, to address the issue of overfitting caused by the large number of parameters and computational complexity of fully connected layers, DCGAN removes the fully connected layers altogether. Batch normalization is employed to map the input sample data to a range with zero mean and unit variance, making the training process easier and faster. Additionally, DCGAN incorporates various activation functions instead of using a single fixed activation function. The generator structure of DCGAN is illustrated in Figure 3.

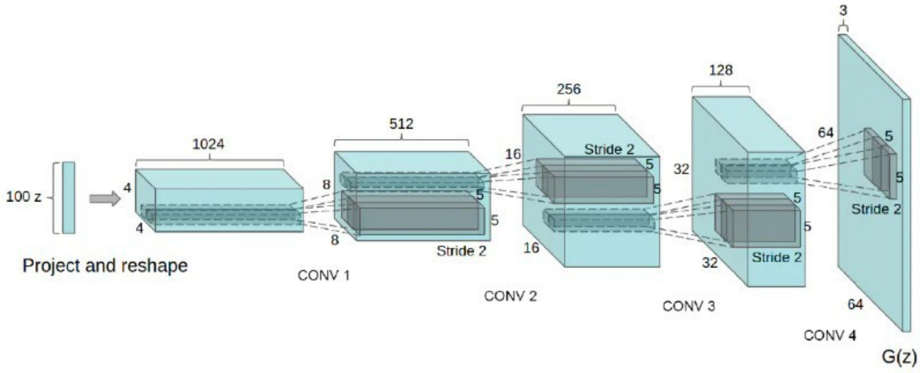


FIGURE 3. The generator structure of DCGAN

2.3. **Cooperative game.** Cooperative game theory is a theoretical framework within game theory. It is based on the exchange of information among multiple players and involves enforceable constraints. It establishes mechanisms for cooperation, mutual trust, and mutual constraints. In cooperative games, participants form alliances and cooperate, which can increase the benefits for the cooperating parties as well as the overall system. The cooperative surplus generated in cooperative games can be shared among the participants. The establishment of cooperative game mechanisms primarily requires the following four points: 1) The participants in the game share common interests. 2) The participants in the game need necessary information exchange. 3) The participants in the game engage voluntarily, on equal footing, and with mutual benefits. 4) The game requires a binding contract or agreement. Cooperative game theory can be represented by a binary tuple $(N, v)$, where $N$ is a set consisting of a finite number of participants, and $v : 2^N \to \mathbb{R}$ represents the payoff function for all possible cooperative combinations $S$. In the game $G = (N, v)$ if for any $S$, $T \subset N$, and $S \cap T = \emptyset$, it holds that $v\,(S \cup T) \geq v\,(S) + v(T)$, then the game is said to be superadditive.

3. **Cooperative Game-Based Generative Adversarial Networks (Coop-GAN).** The traditional generative adversarial network consists of a generative model and a discriminative model. The discriminant model continuously denies the images generated by the generative model. The generation model makes it impossible for the discriminant model to distinguish the generated samples from the real ones. The models in the generative adversarial network learn by increasing each other's losses. They are prone to the problem that one side of the model has zero loss and the other has too much loss during the training process.

This paper proposes a cooperative game-based generative adversarial network structure that uses a cooperative model to train the models in the generative adversarial network. In the traditional generative adversarial network, the discriminative model needs to deny the samples generated by the generative model continuously. In the cooperative model, the discriminative model learns the real samples first. After which, the discriminative model guides the generative model to learn the distribution of the real samples. Compared with the traditional generative adversarial network, this network can effectively train the generative and discriminative models.

In Coop-GAN, the discriminative model is a beating discriminative model system, which further reducing the possibility of mode collapse. The overall architecture of the network is shown in Figure 4.

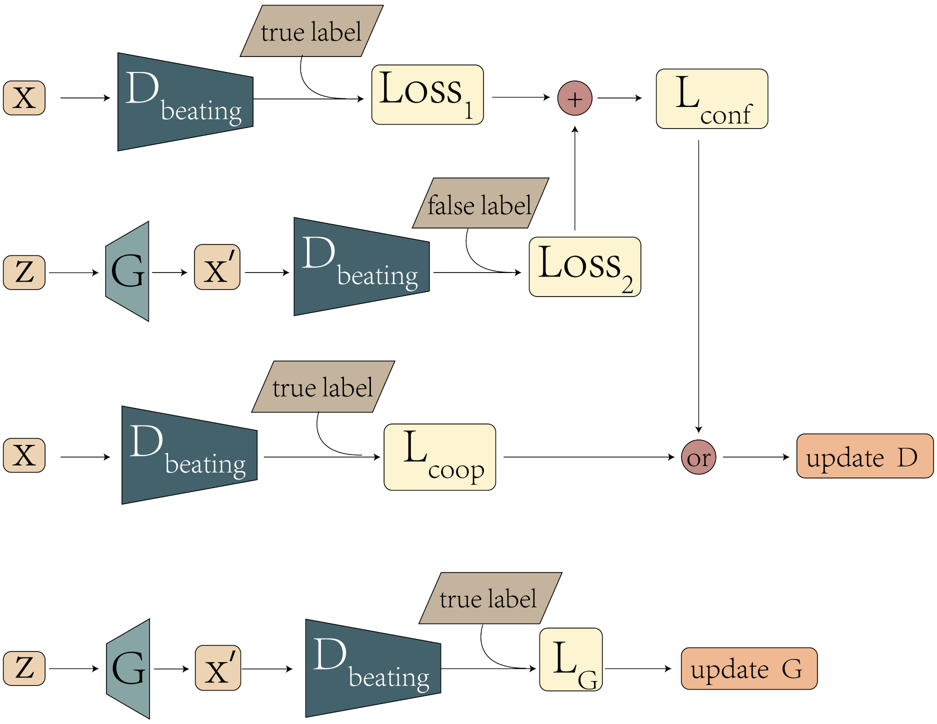


Figure 4. The Coop-GAN structure. $D_{beating}$ is the discriminative model system of beating.

3.1. **Overall network architecture.** The Coop-GAN model structure is divided into three parts including the beating discriminative model system, the generative model, and the cooperative loss.

The beating discriminant model system consists of two discriminative models of different sizes. The input samples (real samples $x$ or generated samples $x'$) enter the beating discriminant model system. After that, the discriminant models of different sizes within the beating discriminant model system periodically alternate receiving input samples and updating the network according to the loss function. The network depth of the two discriminant models is different, and their functional ability to fit real samples is inconsistent. This pairing reduces the cases where the generative models fall into a single generative pattern, thereby reducing the possibility of mode collapse.

The generative model first augments the random input vector by a multilayer perceptron, after which the generative samples are obtained using transpose convolution, regularization, and activation functions. The loss of the generative model is calculated by the beating discriminant model system. The generative model is updated based on the computed loss.

The beating discriminant model system in Coop-GAN periodically cooperates with the generative model. In the cooperative model, the beating discriminant model system only learns the probability distribution of the real samples without denying the images generated by the generative samples. The beating discriminant model system learns the probability distribution of the real samples. The generative model continuously learns from the beating discriminant model system to make its distribution close to the distribution of the beating discriminant model system. In this way, the generative model indirectly learns the probability distribution of real samples. The characteristic based on the cooperative game can avoid the problem that one party loses a lot and the other party loses nothing in a zero-sum game. The experiments show that the final sample effect obtained by the Coop-GAN is better than that of the full adversarial model. The Coop-GAN satisfies the superadditivity in the cooperative game.

3.2. **A discriminative model system of beating.** The discriminative model system of beating can consist of several discriminative models with different depths. This paper uses two discriminant models of different depths in the discriminant model system of beating. The two models are the primary discriminant model $D_{big}$ and the secondary discriminant model $D_{small}$. In the beating discriminator system, these two discriminator models alternately receive image inputs and output corresponding judgment scores. The frequency of alternation is determined by a hyperparameter.

In the Fashion-MNIST[25] dataset (the image size is 28*28), the discriminant model system of beating is shown in Figure 5.

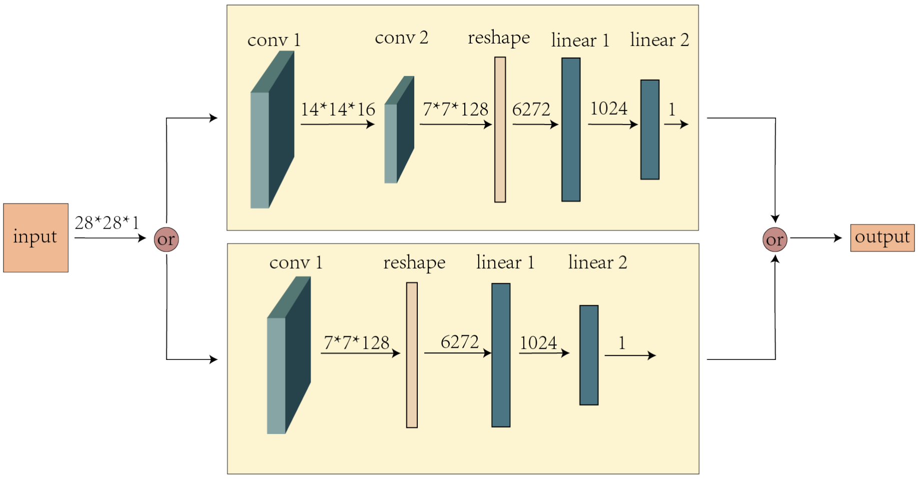


FIGURE 5. A discriminative model system of beating. Discriminators of different sizes alternate in receiving image inputs and outputting their judging scores during network training. The frequency of alternation is a hyperparameter.

When using $D_{big}$ as the discriminant model, the following operations are available. 1) The input sample of size (28,28,1) is passed through a convolutional layer (conv1) with a convolutional kernel size of 4x4 and a step size of 2 to obtain a 3D tensor of size (14,14,16). 2) The (14,14,16) tensor inputs to a convolution layer (conv2) with a convolution kernel size of 4x4 and a step size of 2, resulting in a 3D tensor of size (7,7,128). 3) Reshape the (7,7,128) tensor into a one-dimensional tensor of size 6272. 4) A one-dimensional tensor size 6272 is used as input to the fully connected layer (linear1). A tensor of size 1024 is output. 5) A one-dimensional tensor of size 1024 is used as the input of the fully connected layer (linear2). The output quantity of size 1 is the evaluation score of the discriminant model on the input samples.

When using $D_{small}$ as the discriminant model, the following operations are available. 1) The input sample of size (28,28,1) is passed through a convolutional layer (conv1) with a convolutional kernel size of 8x8 and a step size of 4 to obtain a 3D tensor of size (7,7,128). 2) Reshape the tensor of (7,7,128) into a one-dimensional tensor of size 6272. 3) A one-dimensional tensor of size 6272 as input to the fully connected layer (linear1) and an output tensor of size 1024. 4) A one-dimensional tensor of size 6272 is used as the input of the fully connected layer (linear2), and the output quantity of size 1 is the evaluation score of the discriminant model on the input samples.

By introducing two discriminator models with different depths, namely $D_{big}$ and $D_{small}$, in the beating discriminator system, the diversity of the model is increased. In this study, since $D_{big}$ has an additional convolutional layer compared to $D_{small}$, the two discriminator models have different capabilities in fitting functions. This structure helps reduce the occurrence of mode collapse in the generator by providing diverse feedback and guidance during the training process. The generative model training process may occasionally find distributions that make the discriminative model consistently identify as correct. This problem can trigger the phenomenon of mode collapse. In the beating discriminant model system, the discriminant models are multiple models with different fitting capabilities. Thus the likelihood of mode collapse is less than that of a single discriminant model. Indeed, such design features make Coop-GAN more stable and capable of generating more diverse and realistic samples.

3.3. **Cooperative loss.** This paper adopts a different way of calculating loss from the traditional GAN. According to the cooperative game model, the original loss based on zero sum game is improved. The loss of this paper includes three parts, which are the discriminative model adversarial loss $L_{conf}$, the discriminative model cooperative loss $L_{coop}$ and the generative model loss $L_G$. The structure is as follows in Figure 6.

3.4. **Model training of the coop-GAN.** The task of the discriminant model is not only to distinguish the real samples from the generative samples but also to learn the real samples. After that, the generative model learns the distribution of the discriminant model. This approach can achieve the purpose of the discriminative model leading to generative learning. When the model is adversarial, the discriminant model loss is:

$$V(D) = E_{x\sim p_{data(x)}}[logD(x)] + E_{Z\sim p_{z(z)}}\left[\log(1 - D\big(G(z)\big))\right]$$

When the model is cooperative, the discriminant model no longer adds the evaluation scores of the generated model to the loss, but only learns the distribution of the true samples. The objective of the discriminant model at this point is

$$\max_D V(D) = E_{x\sim p_{data(x)}}[logD(x)]$$

The algorithm of Coop-GAN is in Algorithm 1.

## 4. **Experimental Results and Analysis.**

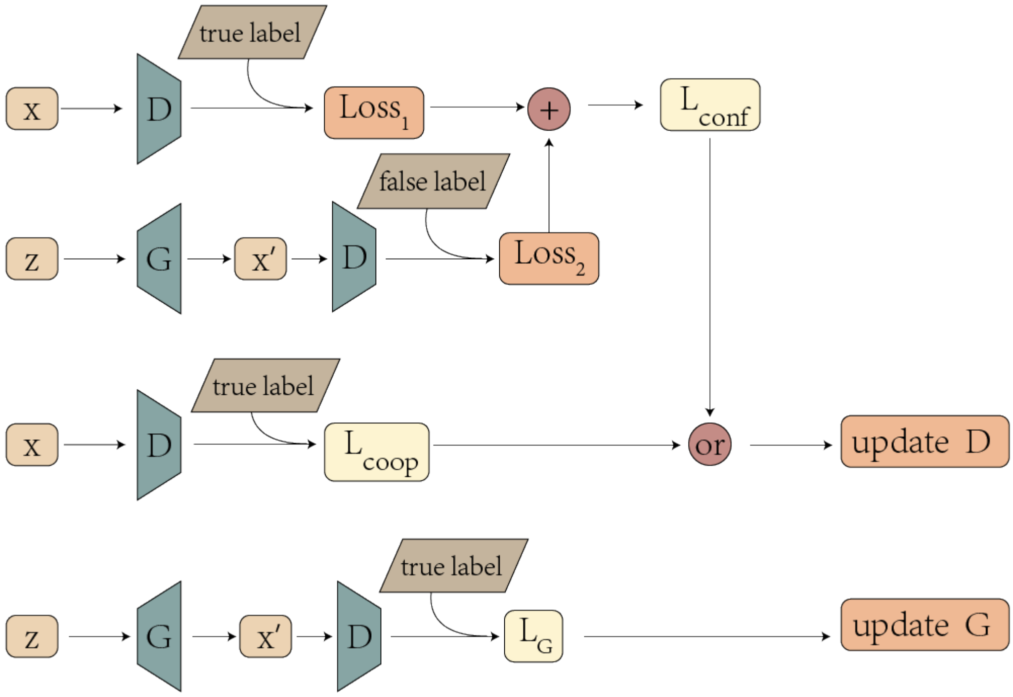


FIGURE 6. Cooperative loss. In the adversarial model, the discriminative model system loss $L_{conf}$ is the sum of the discriminative errors for the real and generated images. In the cooperative model, the discriminative model system loss $L_{coop}$ is only the score of the true image. Regardless of whether the model is cooperative or adversarial, the goal of the generative model is always to try to discriminate the model to determine the generative model as the true sample.

4.1. **Experimental environment and data set.** The operating system of this paper is Ubuntu Server 20.04 LTS, the CPU is 12-core Intel(R) Xeon(R) Silver 4310 CPU @ 2.10GHz, and the GPU is GRID A100-40C. The deep learning framework used is PyTorch, and the programming environment is python. The images of experimental training and comparison tests were obtained from the Fashion-MNIST dataset. The Fashion-MNIST dataset is an image dataset used as an alternative to the MNIST[10] dataset, which contains grayscale images of 70,000 different products in 10 categories, with image sizes (28,28,1).

4.2. **Fashion-MNIST experiment.** The network architecture used in the training process is shown in Figure 4, and the parameters are set as follows: 1) Batch is 32, 128, 512 2) Epoch set to 100 3) The Adam optimizer's learning rate is 0.0002. The exponential decay rate of first-order moment estimation is 0.5. The exponential decay rate of second-order moment estimation is 0.999. 4) The generating models using the Tanh function as the activation function for the last layer and the ReLU as the activation function for the other layers. 5) The last layer of the discriminant model in the discriminant model system of beating all use the Sigmoid function as the activation function. The other layers use the Leaky ReLU as the activation function. 6) Batch normalization is used for each layer to normalize the input of the hidden layer in batches. FID [26] (Fréchet Inception Distance) quantifies the difference between the final generated samples and the real samples using

**Algorithm 1** The Coop-GAN training process.

```
Set the default values $m$=512, $\alpha$ =0.0002, $\beta_1$=0.5, $\beta_2$=0.999, $j$=1/10, $t$=1/2, $epoch$=100
Specify: $m$ is the batch size, $\alpha$ is the learning rate, $\beta_1$ and $\beta_2$ are the Adam hyper-parameter, $j$ is the beating frequency, $t$ is the cooperation frequency, and $epoch$ is the number of iterations.
for $t = 1$ to $epoch$ do
  $count_{coop} = 0$
  $count_{conf} = 0$
  for the number of training iterations do
    Sampling from real distributions $\{x^{(i)}\}_{i=1}^{m} \sim p_{data}(x)$
    Obtaining noise samples $\{z^{(i)}\}_{i=1}^{m} \sim p_g(x)$
    if $(count_{coop} * t)\%1 == 0$ then
      $count_{coop} = count_{coop} + 1$
      if $(count_{coop} * t * j)\%1 = 0$ then
        $\omega_{D_{small}} \leftarrow Adam(\nabla_{\omega_D}\frac{1}{m}\sum_{i=1}^{m}\log\left(D_{small}(x^{(i)})\right))$
        $\omega_G \leftarrow Adam(\nabla_G\frac{1}{m}\sum_{i=1}^{m}\log(1 - D_{small}(G(z^{(i)}))))$
      else
        $\omega_{D_{big}} \leftarrow Adam(\nabla_{\omega_{D_{big}}}\frac{1}{m}\sum_{i=1}^{m}\log(D_{big}(x^{(i)})))$
        $\omega_G \leftarrow Adam(\nabla_G\frac{1}{m}\sum_{i=1}^{m}\log\left(1 - D_{big}(G(z^{(i)}))\right))$
      end if
    else
      $count_{conf} = count_{conf} + 1$
      if $(count_{conf} * j)\%1 == 0$ then
        $\omega_{D_{small}} \leftarrow Adam(\nabla_{\omega D_{small}}\frac{1}{m}\sum_{i=1}^{m}\log\left(D_{small}(x^{(i)}) + \log\left(1 - D_{small}\left(G\left(z^{(i)}\right)\right)\right)\right))$
        $\omega_G \leftarrow Adam(\nabla_G\frac{1}{m}\sum_{i=1}^{m}\log\left(1 - D_{small}(G(z^{(i)}))\right))$
      else
        $\omega_{D_{big}} \leftarrow Adam(\nabla_{\omega D_{big}}\frac{1}{m}\sum_{i=1}^{m}\log\left(D_{big}(x^{(i)}) + \log\left(1 - D_{big}\left(G\left(z^{(i)}\right)\right)\right)\right))$
        $\omega_G \leftarrow Adam(\nabla_G\frac{1}{m}\sum_{i=1}^{m}\log\left(1 - D_{big}(G(z^{(i)}))\right))$
      end if
    end if
  end for
end for
```

different beating frequencies versus different cooperation frequencies. The FID is a metric used to calculate the distance between the feature vectors of the real and generated images, which is calculated as follows:

$$d^2((m, C), (m_w, C_w)) = \|m - m_w\|_2^2 + Tr(C + C_w - 2(CC_w)^{1/2})$$

The FID score is denoted as $d^2$, which represents its distance as a squared term. Where $m$ is the feature mean of the real image. $m_w$ is the feature means of the generated image. $C$ and $C_w$ are the covariance matrices of the feature vectors of the real image and the generated image, respectively. In the measured results, the image quality is better when the FID is lower.

4.2.1. *Superadditivity verification.* In the structure of generative adversarial network based on cooperative game, the $D_{big}$, $D_{small}$ and $G$ form a participant set $N$, which is of size 3. The FID value of the final sample is taken as its payoff function $v$. Since the more minor the FID value, the better the effect, let the final payoff be:

$$f = 100 - FID$$

The FID values of the final generated samples under different cooperation strategies are shown in Table 1.

TABLE 1. The Fid Values Of The Final Generated Samples Under Different Cooperation Strategies

| Cooperation Strategy | Final Benefits $f$ |
|---|---|
| $v(D_{big}) + v(D_{small}) + v(D_G)$ | 79.56 |
| $v(D_{small} \cup G) + v(D_{big})$ | 80.38 |
| $v(D_{big} \cup G) + v(D_{small})$ | 80.65 |
| $v(D_{big} \cup D_{small} \cup G)$ | 82.03 |

$v(D_{big} \cup D_{small} \cup G)$ indicates that cooperative loss is used among $G$, $D_{small}$ and $D_{big}$. $v(D_{big} \cup G) + v(D_{small})$ indicates that cooperative loss is used between $G$ and $D_{big}$ and not between $G$ and $D_{small}$. $v(D_{small} \cup G) + v(D_{big})$ indicates that cooperative loss is used between $G$ and $D_{small}$ and not between $G$ and $D_{big}$. $v(D_{big}) + v(D_{small}) + v(D_G)$ indicates that cooperative loss is not used between any of $G$, $D_{small}$ and $D_{big}$. The experiment show that this method satisfies:

$$v(D_{big} \cup D_{small} \cup G) > v(D_{small} \cup G) + v(D_{big}) > v(D_{big}) + v(D_{small}) + v(D_G)$$

$$v(D_{big} \cup D_{small} \cup G) > v(D_{big} \cup G) + v(D_{small}) > v(D_{big}) + v(D_{small}) + v(D_G)$$

The experiment show that gains in the cooperative state are higher than gains in the adversarial state. The gains are higher for large alliances than for small alliances. They are satisfied with the game $G = (N, v)$, in which the payoffs are higher for any $S$, $T \subset N$, and $S \cap T = \emptyset$, both have $v\,(S \cup T) \geq v\,(S) + v(T)$. Therefore the method satisfies superadditivity.

4.2.2. *Fashion-MNIST experimental results.* The experiment to change the cooperation frequency and beating frequency in the Coop-GAN. Explore the influence on the quality of image generation and the process of image generation. In this paper, we take the cooperation frequency as 1/2, 1/3, 1/5, 1/10, 0, and the beating frequency as 1/2, 1/3, 1/5, 1/10, 0, respectively, under each batch size, for a total of 3*5*5 combinations. The smaller the cooperation frequency, the fewer times the cooperative approach is used to update the network. The cooperative loss is not used when the cooperative frequency and beating frequency are zero.

Table 2 shows the FID values of the final generated images for different combinations of cooperation frequency and different beating frequencies. It can be seen from Table 2 that when the batch size increases, the FID value of the samples generated by the original GAN network becomes larger and larger. The appropriate cooperation and beating frequency can significantly reduce the FID value. For example, at the batch size of 512, using the Coop-GAN with a cooperation frequency of 1/2 and a beating frequency of 1/10 can reduce the FID value by 24.2%. The FID values of the generated images with batch sizes 32, 128, and 512 are 18.48, 20.24, and 22.98, respectively. The optimal FID values of the Coop-GAN are 16.96, 17.35, and 17.43. With different combinations of cooperation frequency and beating frequency, there are different proportions of the Coop-GAN outperforming the traditional GAN in generating the final sample FID values. The proportions are 12.5%, 75%, and 100% for batch sizes of 32, 128, and 512, respectively.

Table 2. Fid Values Of The Final Generated Samples With Different Batch Sizes, Beating Frequency, And Cooperation Frequency

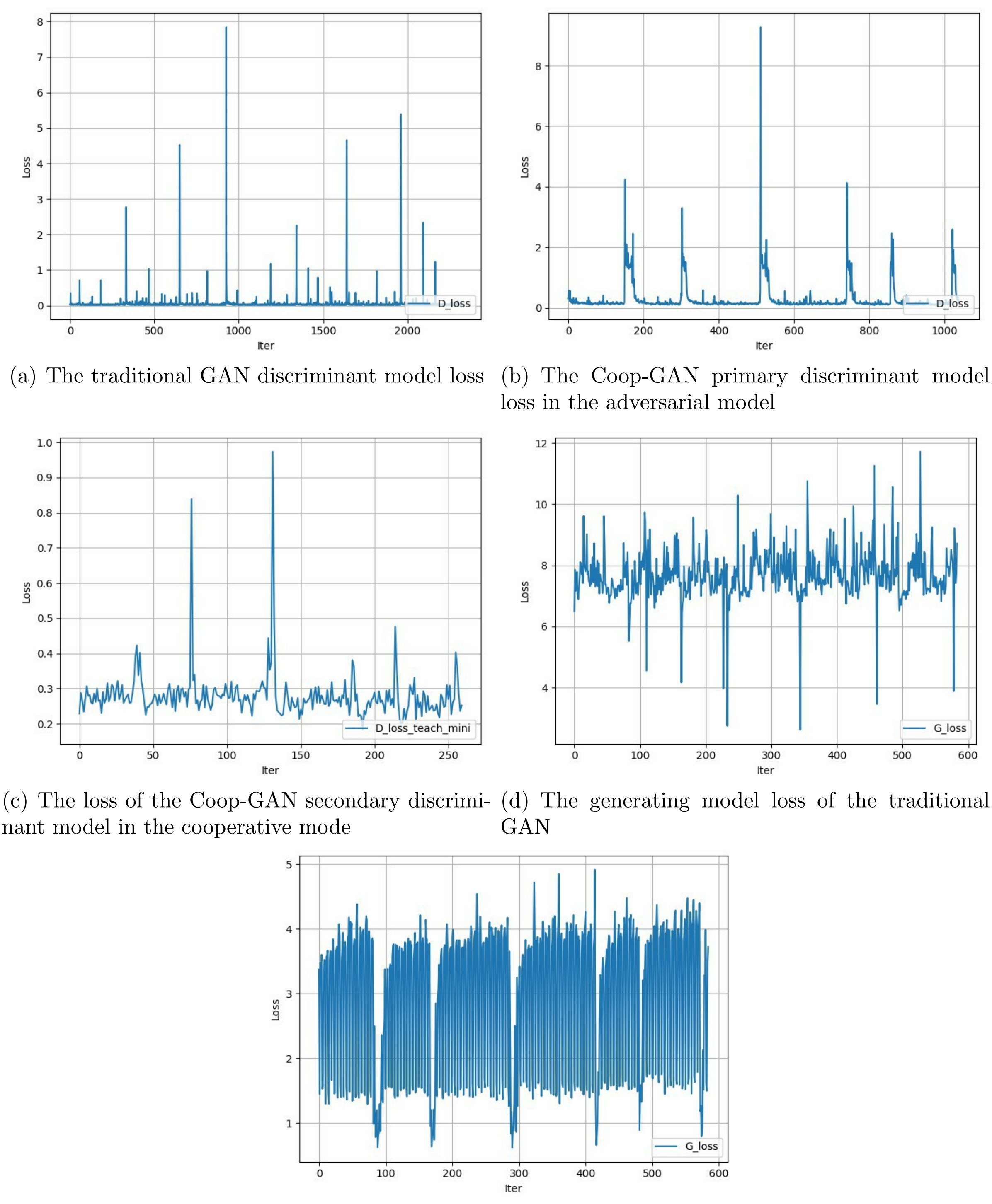
| Batch size | Beating frequency | Cooperation frequency | | | | |
|---|---|---|---|---|---|---|
| | | 1/2 | 1/3 | 1/5 | 1/10 | 0 |
| 32 | 1/2 | 22.60 | 18.87 | 19.99 | 20.74 | 19.02 |
| | 1/3 | 19.57 | 19.14 | 19.55 | 19.14 | 20.11 |
| | 1/5 | 21.74 | 18.18 | **16.96** | 19.24 | 18.90 |
| | 1/10 | 18.06 | 18.59 | 21.92 | 20.83 | 21.86 |
| | 0 | 19.94 | 20.90 | 19.51 | 20.44 | 18.48 |
| 128 | 1/2 | 18.05 | 18.75 | 19.63 | 17.74 | 25.15 |
| | 1/3 | 19.93 | 18.19 | 21.80 | 19.42 | 20.57 |
| | 1/5 | 17.76 | 19.73 | 18.99 | 18.36 | 19.52 |
| | 1/10 | 18.95 | **17.35** | 17.39 | 20.36 | 20.31 |
| | 0 | 20.29 | 19.43 | 19.52 | 18.72 | 20.25 |
| 512 | 1/2 | 20.28 | 18.94 | 17.73 | 19.95 | 21.19 |
| | 1/3 | 19.98 | 18.10 | 20.11 | 18.45 | 20.51 |
| | 1/5 | 18.16 | 19.53 | 18.76 | 21.09 | 21.15 |
| | 1/10 | **17.43** | 20.45 | 21.78 | 17.88 | 18.27 |
| | 0 | 19.01 | 19.08 | 20.17 | 20.91 | 22.99 |

Figure 7(a) represents the traditional GAN discriminant model loss. In terms of trend, the loss of traditional GAN decreases rapidly, and the loss values mostly converge to zero. The traditional GAN network discriminator can always distinguish the generative samples from the real ones. The over-excellent discriminative model makes it difficult to update the generative model.

Figure 7(b) denotes the Coop-GAN primary discriminant model loss in the adversarial model. The losses show periodic fluctuations. When the primary discriminant model performs too well, large but manageable fluctuations in losses occur due to perturbations in the jump structure. It can achieve the effect of facilitating model training.

Figure 7(c) shows the loss of the Coop-GAN secondary discriminant model in the cooperative model. The cooperative loss indicates that the generative model does not fight against the discriminant model and only learns the loss of the real samples. Its role is to learn the distribution of the real samples and pass it to the generative model. The secondary discriminant model refers to the discriminant model with a simpler model structure in the beating discriminant model system. The loss value of the secondary discriminant model is smaller and more stable.

The loss of the traditional GAN network generation model is shown in Figure 7(d), which shows high loss and oscillates dramatically. Figure 7(e) shows the loss of the Coop-GAN generation model. As the discriminative model system beats, its loss function also beats regularly within a range, and the extreme difference is much smaller than that of the traditional GAN. Due to the characteristics of the Coop-GAN structure, the generative model does not only fight with the discriminative model system during the learning process but also cooperates with the discriminative model system periodically. Therefore, the



(a) The traditional GAN discriminant model loss

(b) The Coop-GAN primary discriminant model loss in the adversarial model

(c) The loss of the Coop-GAN secondary discriminant model in the cooperative mode

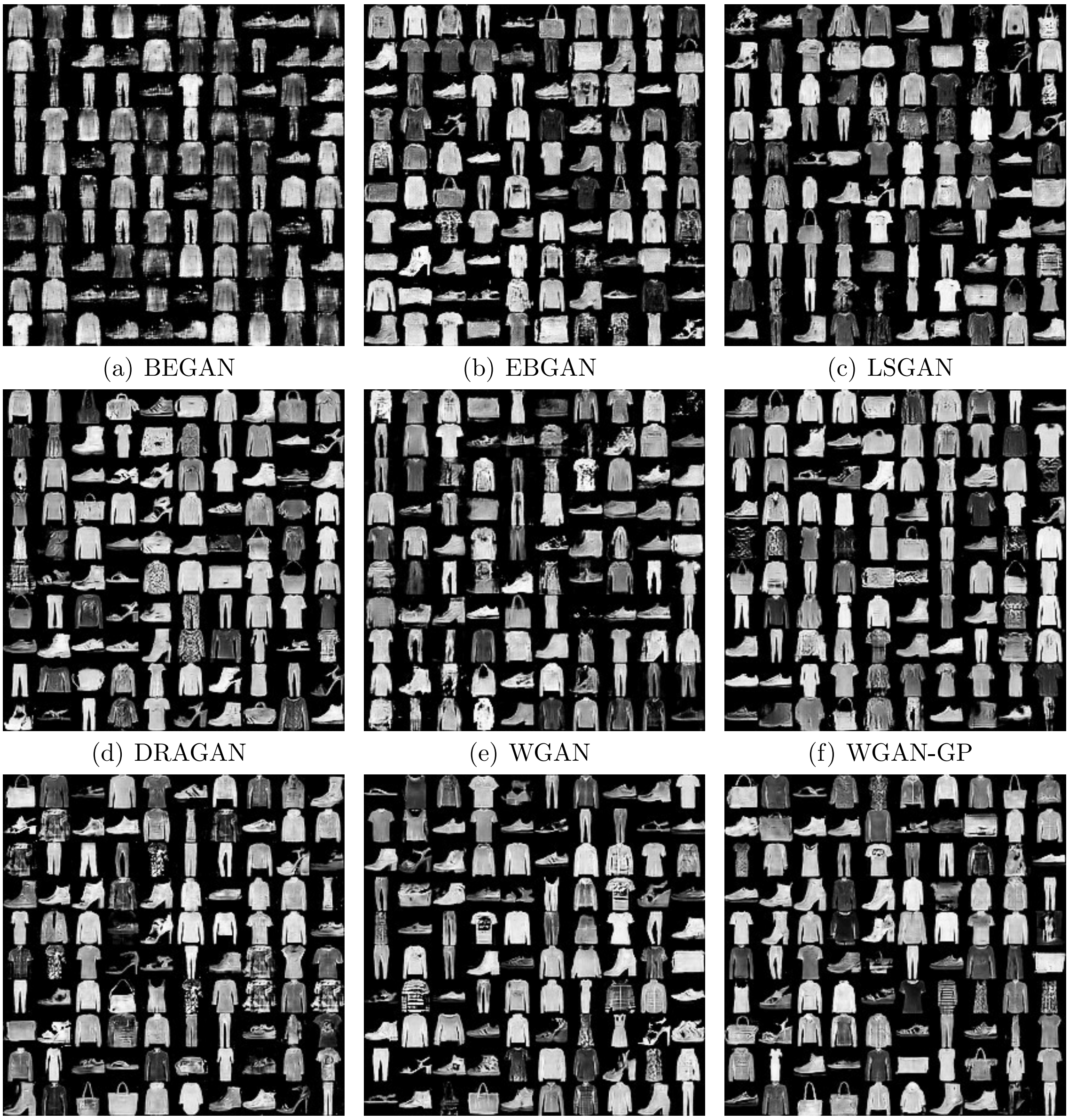
(d) The generating model loss of the traditional GAN

(e) The Coop-GAN generating model loss

Figure 7. Losses Of Various Models

Coop-GAN generative model loss has a similar periodicity to the Coop-GAN primary discriminant model cooperative loss. In the process of continuous cooperation to promote each other's learning. The beating structure makes the discriminant model not fall into a single discriminant model, thus reducing the problem of model collapse.

Figure 8 show the final generated samples of various GANs with the batch size of 512, in which mode collapse and poor sample quality are found. Figure 8(i) shows the final



(a) BEGAN (b) EBGAN (c) LSGAN

(d) DRAGAN (e) WGAN (f) WGAN-GP

(g) GAN (h) GMAN (i) CoopGAN

Figure 8. Final Generated Samples Of Different GANs With The Batch Size Of 512

generated sample of the Coop-GAN, which has richer diversity and less mode collapse compared with other Gans.

To comprehensively evaluate the stability and quality of the generated samples, two primary metrics, Fréchet Inception Distance (FID) and Inception Score (IS) [27], are utilized in this study. FID is a metric used to measure the difference between the distributions of the generated and real samples. It calculates the statistical distance between the features extracted from the Inception network for real and generated samples. A lower FID value indicates that the generated samples are closer to the real samples, indicating higher sample quality. IS is employed to assess the diversity and realism of the generated samples. A higher IS value indicates better diversity in the generated samples and that the classifier perceives the generated samples as more realistic. The computation of IS involves using the Inception network to evaluate the class and probability distribution of the generated samples.

Table 3. FID AND IS COMPARISON

| *GAN model* | *FID* | *IS* |
|---|---|---|
| *BEGAN* | 24.40 | 3.4±0.05 |
| *EBGAN* | 20.17 | 4.58±0.13 |
| *DRAGAN* | 19.58 | 4.13±0.14 |
| *LSGAN* | 20.75 | 4.24±0.14 |
| *WGAN* | 21.14 | 4.11±0.06 |
| *WGAN-GP* | 19.51 | 4.39±0.07 |
| *GMAN* | 20.76 | 4.56±0.12 |
| *Coop-GAN* | **18.67** | **4.75±0.11** |

From the results in 3, it can be observed that Coop-GAN performs exceptionally well in both FID and IS values. The FID value is 18.67, which is lower than all other GAN models, indicating that the samples generated by Coop-GAN have a smaller distribution difference from the real samples and higher quality. Furthermore, Coop-GAN exhibits an IS value of 4.75 ± 0.11, which is higher than all other GAN models, indicating that the generated samples from Coop-GAN possess better diversity and realism.

In summary, to comprehensively evaluate the stability and quality of the generated samples, this study employs two primary metrics, FID and IS. Coop-GAN demonstrates excellent performance in both FID and IS values. The lower FID value indicates the high quality of generated samples, while the higher IS value suggests improved diversity and realism.

Based on the experimental results above, Coop-GAN outperforms other popular GAN examples in terms of FID and IS values, which can be attributed to the design features of Coop-GAN and the advantages of using a cooperative game mode.

Firstly, Coop-GAN introduces a beating discriminator system, which includes two discriminator models of different depths, increasing model diversity and reducing the likelihood of mode collapse. Secondly, Coop-GAN adopts a cooperative loss, different from the traditional GAN's zero-sum game-based loss. In the cooperative mode, the objective of the discriminator model is solely to evaluate real samples, and the evaluation scores for the generated samples are no longer included in the loss. This cooperative loss allows the discriminator model to focus more on the distribution of real samples under the cooperative mode, rather than merely distinguishing between real and generated samples. This loss design enables the discriminator model to provide more meaningful feedback to the generator, helping the generator better mimic the real data distribution, further enhancing the quality and realism of the generated samples.

However, simultaneously, due to the introduction of the beating discriminator system, the resource requirements for training have increased. The inclusion of two discriminator models of different depths has certainly improved model diversity and reduced the likelihood of mode collapse, but it comes at the cost of additional computational resources.

Combining the above two points, the innovative design of Coop-GAN and the cooperative game mode enable the generator and discriminator to cooperate and learn together, thereby enhancing the diversity and quality of the generated samples. The generated samples are closer to the distribution of real samples and exhibit better diversity and realism. This contributes to Coop-GAN's outstanding performance in terms of FID and IS values.

5. **Improving StyleSwin Network With Coop-GAN Structure.** The Coop-GAN network structure can be applied to almost any generative adversarial network, using StyleSwin [28] as an example. StyleSwin is a GAN network that builds upon the structure of StyleGAN2 [29] but replaces its basic building blocks with Swin Transformer [30]. It incorporates multi-head self-attention (MSA) [31] within non-overlapping windows to create the GAN network. Its discriminative model can be replaced with a skipping discriminative model. The improved Coop-StyleSwin is shown in Figure 9.

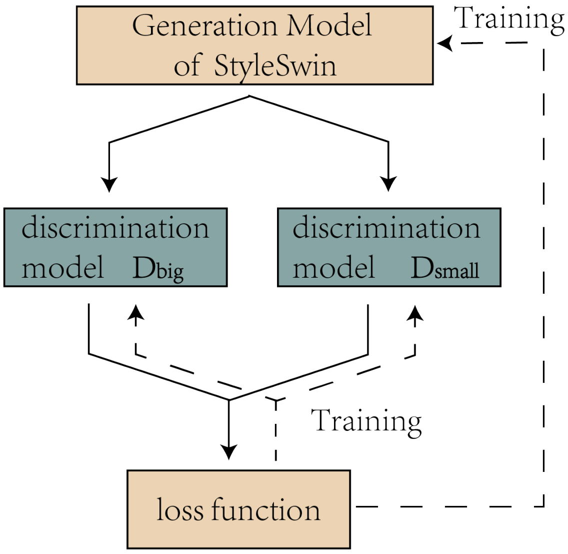

Generation Model of StyleSwin
Training
discrimination model $D_{big}$
discrimination model $D_{small}$
Training
loss function

FIGURE 9. Coop-StyleSwin structure

Figure 10 shows the experimental comparison of three networks on the task of generating 64*64 size face images in the CelebA dataset using FID value. The three networks are StyleSwin and two Coop-StyleSwin networks with different collaboration and skipping frequencies. The experiment shows that the FID values of images generated by Coop-StyleSwin continue to oscillate and quickly drop to around 18.5, which is better than the StyleSwin network with the same training time.

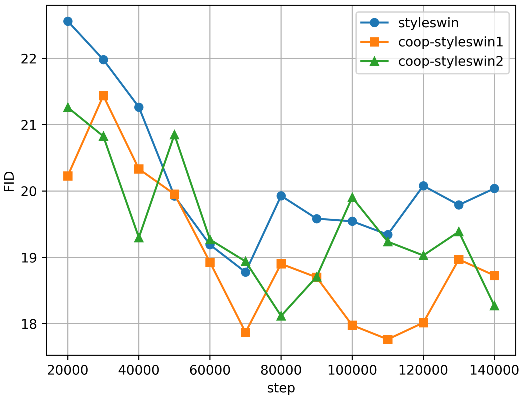


FIGURE 10. Comparison of FID values for image generation using StyleSwin and Coop-StyleSwin on the CelebA dataset

6. **Experimental Results and Analysis.** In this paper, the GAN network structure is improved based on the characteristics of cooperative games and the use of a beating structure. The cooperative loss can make the discriminative model and generative model learn image distribution from adversarial to cooperative The beating discriminative model system prevents the discriminative model from falling into a single discriminative model. Experiments show that the Coop-GAN satisfies the superadditivity of cooperative games. The gains from using cooperative model is higher than that from using only adversarial model. The Fashion-MNIST dataset performs better as the batch size increases. When the batch size is 512, the FID value of the final sample generated by the Coop-GAN is lower than that of the traditional GAN. The appropriate combination of cooperation frequency and beating frequency can significantly reduce the FID value at each batch size. The Coop-GAN has the best FID and IS values compared to the other exemplified popular GANs.

However, the research on Coop-GAN still faces some potential limitations or challenges. For instance, its generalizability may require further validation and exploration, especially when applied to larger-scale and more complex datasets. Additionally, due to the use of multiple discriminator models, Coop-GAN may demand higher computational resources. Lastly, the selection of beating frequency and cooperation frequency is not fixed and needs adjustment based on different tasks, increasing the number of hyperparameters that need to be tuned.

To further develop and enhance Coop-GAN, future research can explore the effects of different ranges of beating frequency and cooperation frequency by investigating various parameter combinations. Moreover, it is possible to study adaptive methods for adjusting the cooperation frequency and beating frequency to achieve more flexible model learning. Furthermore, incorporating other advanced GAN techniques such as attention mechanisms, adaptive normalization for the generator and discriminator, can further improve the performance and stability of Coop-GAN.

Overall, as a generative adversarial network based on cooperative game theory and a beating structure, Coop-GAN demonstrates potential in terms of generated sample quality and diversity. However, its development is still in its early stages, requiring further research and exploration to address existing limitations and promote its advancement and application in broader domains.

**Acknowledgments** This paper was supported partly by the National Natural Science Foundation of China under Grant No. 61976183; National Social Science Foundation of China (21BJY151); Major project of Fujian Social Science Research Base (FJ2022JDZ063); Educational Teaching Research Project of Undergraduate Universities in Fujian Province of China (FBJG20220233).

## REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[2] A. Washburn, *Two-Person Zero-Sum Games.* Springer US, New York, 2014.

[3] J. Heaton, "Ian goodfellow, yoshua bengio, and aaron courville: Deep learning," *Genetic Programming and Evolvable Machines*, vol. 19, no. 1-2, pp. 305–307, 2018.

[4] Y. Bengio, E. Laufer, G. Alain, and J. Yosinski, "Deep generative stochastic networks trainable by backprop," in *Proceedings of the 31st International Conference on Machine Learning (ICML).* PMLR, 2014, pp. 226–234.

[5] K. Kurach, M. Lučić, X. Zhai, M. Michalski, and S. Gelly, "A large-scale study on regularization and normalization in GANs," in *Proceedings of the 36th International Conference on Machine Learning (ICML).* PMLR, 2019, pp. 3581–3590.

[6] L.-J. Ratliff, S.-A. Burden, and S.-S. Sastry, “Characterization and computation of local nash equilibria in continuous games,” in *2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton).* IEEE, 2013, pp. 917–924.

[7] N. Kodali, J. Abernethy, J. Hays, and Z. Kira, “On convergence and stability of gans,” *arXiv:1705.07215*, 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1705.07215

[8] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, “Accurate, large minibatch sgd: Training imagenet in 1 hour,” *arXiv:1706.02677*, 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1706.02677

[9] N. Shirish-Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. Tak-Peter-Tang, “On large-batch training for deep learning: Generalization gap and sharp minima,” in *5th International Conference on Learning Representations (ICLR 2017).* OpenReview.net, 2017, pp. 1–16.

[10] F. Wang, H. Liu, D. Samaras, and C. Chen, “Topogan: A topology-aware generative adversarial network,” in *Europeon Conference on Computer Vision (ECCV).* Springer, 2020, pp. 118–136.

[11] L. Jiang, B. Dai, W. Wu, and C.-C. Loy, “Focal frequency loss for image reconstruction and synthesis,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).* IEEE, 2021, pp. 13 919–13 929.

[12] I. P. Durugkar, I. Gemp, and S. Mahadevan, “Generative multi-adversarial networks,” in *5th International Conference on Learning Representations (ICLR).* OpenReview.net, 2017, pp. 1–14.

[13] D. Berthelot, T. Schumm, and L. Metz, “Began: Boundary equilibrium generative adversarial networks,” *arXiv:1703.10717*, 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1703.10717

[14] N. N. Vorob’ev, “The present state of the theory of games,” *Russian Mathematical Surveys*, vol. 25, p. 77, 1970.

[15] R.-K. Lenroot and J.-N. Giedd, “Brain development in children and adolescents: Insights from anatomical magnetic resonance imaging,” *Neuroscience Biobehavioral Reviews*, vol. 30, no. 6, pp. 718–729, 2006.

[16] D. Endres and J. Schindelin, “A new metric for probability distributions,” *IEEE Transactions on Information Theory*, vol. 49, no. 7, pp. 1858–1860, 2003.

[17] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *Proceedings of the 34th International Conference on Machine Learning (ICML).* PMLR, 2017, pp. 214–223.

[18] M. Arjovsky and L. Bottou, “Towards principled methods for training generative adversarial networks,” in *5th International Conference on Learning Representations (ICLR).* OpenReview.net, 2017, pp. 1–17.

[19] I. Csiszar, “*i*-divergence geometry of probability distributions and minimization problems,” *The Annals of Probability*, vol. 3, pp. 146–158, 1975.

[20] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A.-C. Courville, “Improved training of wasserstein GANs,” in *Advances in neural information processing systems (NIPS).* Curran Associates Inc., 2017, pp. 1–11.

[21] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference for Learning Representations (ICLR).* OpenReview.net, 2015, pp. 1–15.

[22] X. Mao, Q. Li, H. Xie, R.-Y.-K. Lau, Z. Wang, and S.-P. Smolley, “Least squares generative adversarial networks,” in *IEEE International Conference on Computer Vision (ICCV).* IEEE Computer Society, 2017, pp. 2813–2821.

[23] N. Yu, G. Liu, A. Dundar, A. Tao, B. Catanzaro, L.-S. Davis, and M. Fritz, “Dual contrastive loss and attention for gans,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).* IEEE, 2021, pp. 6731–6742.

[24] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” in *4th International Conference on Learning Representations (ICLR).* OpenReview.net, 2016, pp. 1–16.

[25] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” *arXiv:1708.07747*, 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1708.07747

[26] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” in *Advances in neural information processing systems (NeurIPS).* Curran Associates Inc., 2017, pp. 6626–6637.

[27] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” in *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS).* Curran Associates Inc., 2016, p. 2234—2242.

[28] B. Zhang, S. Gu, B. Zhang, J. Bao, D. Chen, F. Wen, Y. Wang, and B. Guo, "Styleswin: Transformer-based gan for high-resolution image generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 2022, pp. 11 304–11 314.

[29] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 2020, pp. 8107–8116.

[30] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).* Curran Associates Inc., 2021, pp. 10 012–10 022.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30 (NIPS).* Curran Associates Inc., 2017, pp. 5998–6008.