CLR-GAN: Improving GANs Stability and Quality via Consistent Latent Representation and Reconstruction

Shengke Sun[1⋆], Ziqian Luan[2⋆], Zhanshan Zhao[1,3†], Shijie Luo[1], and Shuzhen Han[1]

[1] Tiangong University, 300387 Tianjin, China
{sunshengke,hanshuzhen,luoshijie}@tiangong.edu.cn, zhzhsh127@163.com
[2] Xidian University, 710126 Xi'an, Shaanxi, China
tyuwwe@foxmail.com
[3] Institute for Sustainable Industries & Liveable Cities, Victory University, Melbourne, VIC 8001, Australia
[†]Corresponding author

Abstract. Generative Adversarial Networks(GANs) have received considerable attention due to its outstanding ability to generate images. However, training a GAN is hard since the game between the Generator(G) and the Discriminator(D) is unfair. Towards making the competition fairer, we propose a new perspective of training GANs, named **C**onsistent **L**atent **R**epresentation and Reconstruction(CLR-GAN). In this paradigm, **we treat the G and D as an inverse process**, the discriminator has an additional task to restore the pre-defined latent code while the generator also needs to reconstruct the real input, thus obtaining a relationship between the latent space of G and the out-features of D. Based on this prior, we can put D and G on an equal position during training using a new criterion. Experimental results on various datasets and architectures prove our paradigm can make GANs more stable and generate better quality images(**31.22% gain of FID on CIFAR10 and 39.5% on AFHQ-Cat**, respectively). We hope that the proposed perspective can inspire researchers to explore different ways of viewing GANs training, rather than being limited to a two-player game. The code is publicly available at https://github.com/Petecheco/CLR-GAN.

Keywords: Generative Adversarial Networks · Image generation · Latent space consistency

1 Introduction

Generative models are an important part of computer vision, they can be widely used in fields such as image generation [12, 16, 20, 24], video editing [3, 5, 48, 50], style transfer [9, 27, 55] and even drug discovery [7, 35]. Among various generative

⋆ Equal contribution. Listing order is random.

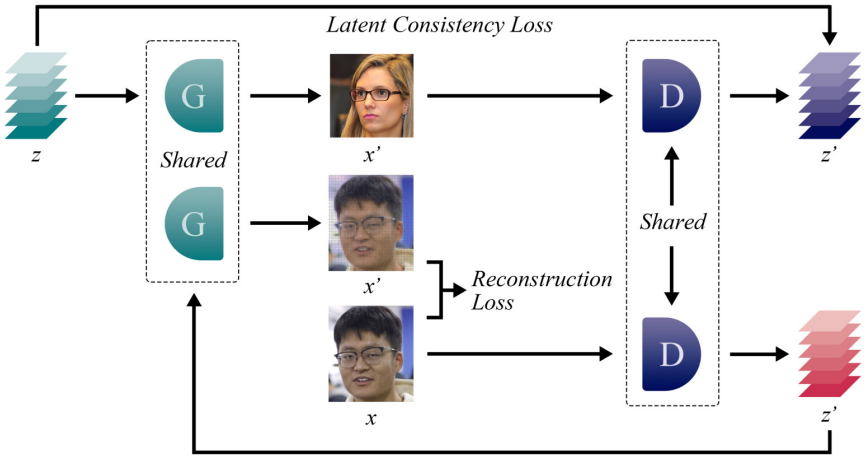

Fig. 1: Sketch map of the proposed method, we proposed two additional learning objectives to consider D and G as inverse process. The top is latent consistency loss where D tries to rebuild the latent distribution of G. The bottom is reconstruction loss where G tries to restore the original real images.

models, Generative Adversarial Networks(GANs) have received considerable attention due to its remarkable performance in generating realistic images [12]. A conventional GAN comprises of two individual networks named generator(G) and discriminator(D). While training a GAN, the generator aims to produce indistinguishable images, and the discriminator tries to figure out the synthesized images from the input images. By the competition between D and G, ideally, we can eventually train a generator that can recover the real distribution and generate high quality samples.

However, it is well known that training a GAN is hard and unstable [26,37]. In general, the instability of training GANs can be attributed to the inequal set up of generator(G) and discriminator(D). Specifically, in the traditional paradigm, the training process is formulated as a two-player game. But in fact, the discriminator seems to dominant the game [2,47]. In practice, the discriminator can easily identify the fake samples from a relatively early stage of the training and maintain this advantage during the whole training, making the generator hard to converge [51]. Many attempts have been done to improve the stability of training GANs. Previous studies attempted to find new metric functions [1, 14, 32, 45], enhance the generator's ability to generate [4,21,40], or use data augment methods [19,30]. These solutions have shown some degree of improvement on training stability, but they did not fundamentally change the core relationship between the generator and discriminator, while ignoring the essence of dynamic balance within GANs architecture. The competition between the generator and discriminator is still unfair. In addition, existing methods have not fully utilized the structural information of the latent space, which limits the further improvement

of GANs generation ability. Therefore, improving the training of GANs still remains an unsolved and challenging problem.

In this paper, contrast to existing methods, we proposed a novel perspective of training GANs. By rethinking the training process and network architecture of GANs. We found that the generator and the discriminator got optimized in a considerably large latent space, resulting unstable training and inconsistency between the generator and the discriminator. While traditional methods primarily focus on adjusting network architectures or tuning hyperparameters to stabilize training [19, 40]. Our method is to view the generator and the discriminator as two inverse components. As shown in Fig. 1. In particular, the discriminator can be regarded as outputting a realness score and extracting samples to the latent representation space at the same time. And the generator not only generates images but restores source images from the extracted latent representation. By constraining the generator and discriminator with these additional tasks, we make the discriminator's representation latent space closer to the generator's latent space, thus making the generator and the discriminator more consistent, leading a fairer game between the generator and the discriminator. This not only provides a more robust framework for training GANs but also opens up new possibilities for improving GANs training from different aspects.

2 Related Work

2.1 Improving GANs through discriminators

As we mentioned above, due to the limitations of original training objective [1, 12], the discriminator tends to easily take over the competition during the early stage of training. Prior works have been devoted to improving the inequality using modified discriminators. Dist-GAN [45] decelerate the convergence of the discriminator by constructing a reconstruction constrain using a auxiliary autoencoder. Wang et al. [47] combined Grad-CAM [42] with GANs to improve the spatial awareness of the generator using the discriminator as a regularizer, lessening the information gap between D and G. DynamicGAN [51] controls the strength of the discriminator by gradually increasing D's learning capacity, thereby preventing the discriminator from being overly dominant in the early stages of training. AdaptiveMix [32] shrinks the feature space of the discriminator by training using constructed hard samples, reducing the feature space of the discriminator and making training easier. There are also some data augmentation methods that enhance model training by providing more data to the model [19, 46]. However, the above methods have not changed the inner properties of training. All gradient during the optimization process is provided by D, so D always occupies a more dominant position during the training process, which will prevent further optimization of the model. While our model, based on limiting D, also constrains D to optimize based on the gradient provided by G. This allows both G and D to optimize themselves using the gradients provided by each other, making the game fairer.

2.2 Latent space of GANs

The latent space of GANs has always been a popular research area due to its rich semantic information. Multi-Code GAN [13] combines multiple latent feature maps with adaptive channels to improve the quality of image reconstruction. ClusterGAN [38] combines latent space encoding and one-hot encoding to obtain a better clustering result. StyleMapGAN [23] utilizes latent space as an accurate feature embedding, improving the performance of image editing and interpolation tasks. Although latent space plays an important role in various downstream tasks. But to our best knowledge, there has been no previous work that directly uses the latent space of GANs as a component to optimize the training of GANs. In our proposed method, we have directly incorporated latent space into the training process of GANs as an additional constrain. The experimental results show that applying this method can effectively improve the quality and diversity of generated images.

2.3 Differences to Reconstruction-based Methods

Prior works have explored the application of reconstruction-based methods within GANs. HoloGAN [39] and PiGAN [6] utilize latent reconstruction to learn the 3D-aware image representations. GLeaD [2] takes advantage of image reconstruction to improve the training dynamics of GANs. However, these methods either lack a specific focus on stabilizing GANs training or solely concentrate on improving the generator without considering the discriminator. Specifically, in the GLeaD paradigm, given an image, we expect D to extract representative features that can be adequately decoded by G to reconstruct the input, however, this paradigm only improves the generator, the discriminator still controls the game. To address these limitations, we use the latent spatial code extracted by the discriminator and reconstruction loss to achieve stable GAN training and generate higher quality and more diverse images. Thus making GANs training more efficient and practical.

3 Method

As we mentioned before, we try to augment the training fairness of GANs via considering the generative process of the generator (G) and the discriminative process of the discriminator (D) as inverse to each other, thereby enhancing the stability and quality of generated samples. To better understand this new perspective, we first review the conventional formulation of GANs in Sec. 3.1. To obtain a fairer training, Sec. 3.2 introduces our new training paradigm named Consistent Latent Representation that considers G and D as an inverse process. Furthermore, in Sec. 3.3, we also propose a Real Image Reconstruction strategy to achieve mutual connection between generator and the discriminator. Then Sec. 3.4 explained our method from a regularization perspective. Finally, Sec. 3.5 gives the full optimization objective and shows the pseudo code of the new training paradigm.

3.1 Preliminary

A GAN usually consists of two components: a generator $G(\cdot)$ and a discriminator $D(\cdot)$. The generator aims to map a latent code z into an image, while the discriminator tries to distinguish the generated image $G(z)$ from the real one x. The conventional GAN trains via a two-player game by optimizing the learning objective as follows:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}\left[logD(x)\right] + \mathbb{E}_{z\sim p_z(z)}\left[log(1-D(G(z)))\right] \quad (1)$$

where $p_z(z)$ and $p_{data}(x)$ represent a random latent representation and real data distribution respectively.

Goodfellow *et al.* [12] have proved that the optimal solution to this objective is that G can eventually recover the distribution of source data and D can not separate the generated images out from real images. However, the game between two players is not fair since in Eq. 1 that the gradient for optimizing all comes from D, which makes the discriminator a natural dominant [2]. Thus the ideal solution is hard to reach in practice. [11, 47]

3.2 Consistent Latent Space

We aim to improve the stability of GANs by introducing a novel perspective, that is to consider the generate and discriminate as an inverse process. Recall that in a regular GAN paradigm, the generator and discriminator are treated as two individual component. While in our proposed view, we have refined the concepts of generator and discriminator and consider them as an inverse process. In our perspective, the generator maps a latent representation to the real data distribution, and the discriminator not only outputs a realness score, but it also transform the high-dimensional distribution into a trainable latent space. By measuring the distance between these two spaces, we can improve GANs training in the following aspects: First, we can to some extent make the generator and discriminator behave more consistent in the mapping process. Second, since the discriminator needs to align with the generator, it can be seen as constraining the discriminator using generator, which makes the game not completely dominated by discriminator. The detailed implementation is shown in Fig. 2.

Extracting Latent Representation through D. In the training of GANs, the discriminator plays a role of downsample high-dimensional data. For any discriminator, it can be divided into two parts, the first part is the feature extractor denoted as $f(\cdot)$ which extracts features from high-dimensional data samples, another serves as a affine from extracted features to realness score to tell whether the input is true(real samples) or false(generated samples). For any input data x and latent code z, the whole discriminate process can be denoted as follows:

$$D(x) = R(f(x)) \quad (2)$$

$$D(z) = R(f(G(z))) \quad (3)$$

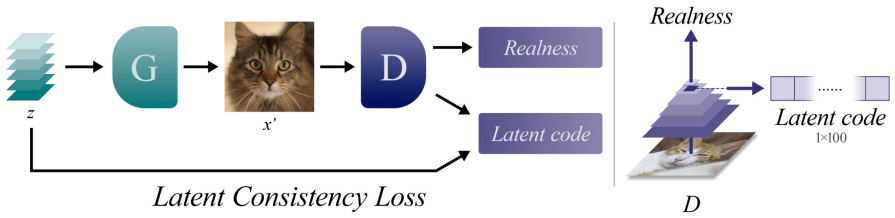

Fig. 2: Detailed implementation of Consistent Latent Representation. Given a latent code z, the generator maps z to an image. The synthesised image was then got discriminated by the discriminator, outputting a realness score and a reconstructed latent code z'. The Consistent Latent loss is calculated between the real latent z and the reconstructed latent z'. The right part specifically shows the architecture of D.

where $f(\cdot)$ is the feature extractor module of discriminator and $R(\cdot)$ is the affine module to transform feature space to the realness score.

Through this discriminate process, we can transform structural high-dimensional input into feature map that contains abundant semantic information [31], thus judging the realness of the input data. We discovered both latent code in the generator and feature map in the discriminator contain lots semantic features, making it possible to align the generator and discriminator using a mapping function. The mapping function converts discriminator's feature space to latent space. Finally we can get a reconstructed latent code $\phi(x)$ from discriminator as follows:

$$\phi(x) = \sigma(f(x)) \tag{4}$$

where x is the real input, $f(\cdot)$ is the feature extractor of D and $\sigma(\cdot)$ is a mapping function that transform feature to latent space, we use an identical affine in this paper for simplicity.

Consistent Latent Representation Loss. As we mentioned above, we now have a real latent code(z) from the generator and a reconstructed latent code(Eq.4) from the discriminator. The latent representation loss can be defined by measuring the distance between these two latent spaces:

$$\mathcal{L}_{CLR} = Dist(\phi(x), z) \tag{5}$$

where $Dist(\cdot)$ is distance measurement for any two distributions. In this paper, we use L1-Distance similar to [55] for all distance measurement, so the $\mathcal{L}_{clr}$ can be rewritten as:

$$\mathcal{L}_{CLR} = \frac{1}{n}\sum_{i=1}^{n} \|z_i - \phi(x_i)\|_1 \tag{6}$$

3.3 Real Image Reconstruction

As mentioned in Sec. 2, in a regular GAN, the generator optimized through the gradient offered by discriminator. Also during the training, the generator has no access to real data distributions, it can only update its parameters implicitly, while the discriminator can get optimized according to real data distribution. This unfair situation helps the discriminator converge faster at an early stage of the training [47], thus the discriminator can not provide useful gradient in the later stage of training, making the generator not perform well.

To solve this problem, we proposed a new strategy called Real Image Reconstruction, which enables the generator to also get gradient from the real data distribution. Specifically, since we have strengthened the discriminator to output both realness and latent code in the last section, we can then use latent code as a bridge to indirectly connect the generator with real distribution. Subsequently, we can use the generated images for a image reconstruction task. The generator then can utilize this gradient that comes from real distribution for better synthesis, making the whole process more efficient. In practice, we can use Eq. 4 to get the reconstructed latent code of real distributions, then we use generator for image restoration.

$$I_{rec} = G(\phi(x)) \tag{7}$$

After reconstructing the image, we can calculate the distance between real images and reconstructed images. And for the reconstruction loss, we use a regular from [55] as follows:

$$\mathcal{L}_{rec} = \frac{1}{n}\sum_{i=1}^{n} \|I(x) - I_{rec}\|_1 \tag{8}$$

where $I(x)$ is the source image data and I_{rec} is the reconstructed image using generator and reconstructed latent code.

3.4 Enhancing Training Stability via Regularization Techniques

To better understand the superiority of our method, we turn to investigate the above training strategies from the perspective of regularization in this section.

With our method, the objective of training GANs can be rewritten as:

$$\begin{aligned}\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}} log D(x) + \mathbb{E}_{z\sim p_z} log[1 - D(G(z))] \\ + \mathbb{E}_{x\sim p_{data}} \parallel G(D(x)) - x\parallel - \mathbb{E}_{x,z} \parallel D(x) - z\parallel\end{aligned} \tag{9}$$

The first two terms of Eq. 9 are the default training objectives for GANs, while the third and fourth terms correspond to Latent Consistency Loss and Real Image Reconstruction Loss, respectively. Note that these two objectives actually appear in the form of L1 norm, and we will consider them as two types of regularization in this section.

When optimizing networks, we actually calculate the gradients of objective function. For the training objective we propose, the gradients can be expressed

as:

$$\nabla_{\theta_d,\theta_g} V(G,D) = \mathbb{E}_{x\sim p_{data}}[\nabla_{\theta_d} logD(x)] + \mathbb{E}_{z\sim p_z}[\nabla_{\theta_d,\theta_g} logD(1-D(G(z)))] + \mathbb{E}_{x\sim p_{data}}\text{sign}(G(D(x))-x) - \mathbb{E}_{x,z}\text{sign}(D(x)-z) \tag{10}$$

where $\text{sign}(G(D(x))-x)$ and $\text{sign}(D(x)-z)$ is simply the sign of both constrains applied element-wise.

Next, we will discuss the role of our regularization in two different scenarios:

Training the Discriminator with a fixed Generator. When the generator is fixed, we can consider the gradients related to the generator to be 0, so we can obtain the gradients as follows:

$$\nabla_{\theta_d} V(D) = \mathbb{E}_{x\sim p_{data}}[\nabla_{\theta_d} logD(x)] - \mathbb{E}_{x,z}\text{sign}(D(x)-z) \tag{11}$$

Considering the general training situation of GANs, when the discriminator is too strong during training. The first term in the gradient tends to be large, resulting in unstable gradient descent. However with our regularization term. Due to the negative sign in the front, we can reduce the gradient of D using this term, making the gradient smaller and can stable the training.

Training the Generator with a fixed Discriminator. As in the previous case, we first write the representation for the gradient:

$$\nabla_{\theta_g} V(G) = \mathbb{E}_{z\sim p_z}[\nabla_{\theta_d,\theta_g} logD(1-D(G(z)))] + \mathbb{E}_{x\sim p_{data}}\text{sign}(G(D(x))-x) \tag{12}$$

When the discriminator is dominating the game, it can make the first term to become very small, making it difficult for the generator to effectively optimize and generate real images. After adding a regularization term, we found an additional training objective was given to the generator, allowing the generator to optimize through this training objective, and continue to compete with the discriminator instead of collapsing.

Finally, we can find that by adding two additional training objective, our model can penalize the discriminator when it is too strong. It can optimize the generator by providing more information through additional tasks, making the game fairer and improving the effectiveness of image generation.

3.5 Full Objective

With the enhanced discriminator and the reconstruction guided generator, the overall training objective can be rewritten as:

$$\mathcal{L}_D{}' = \mathcal{L}_D + \lambda_1 \mathcal{L}_{CLR} \tag{13}$$

$$\mathcal{L}_G{}' = \mathcal{L}_G + \lambda_2 \mathcal{L}_{rec} \tag{14}$$

Algorithm 1 Training a GAN with Consistent Latent Representation and Reconstruction

```
Require: G and our D that are initialized with random parameters.
  Training data {x_i}, latent variable {z}.
  for t = 1 until converge  do
     Sample z ~ P(Z), x ~ P(X)
     real_score, real_latent= D(x)
     fake_score, fake_latent= D(G(z))
     reconstruct_images = G(real_latent)
     Calculate Consistent Latent Loss with Eq.(6)
     Calculate Reconstruction Loss with Eq. (8)
     Update D and G with Eq. (9), Eq. (10)
  end for
  Output: G with best training set FID.
```

where $\mathcal{L}_G$ and $\mathcal{L}_D$ are the original loss function of generator and discriminator, $\mathcal{L}_{rec}$ and $\mathcal{L}_{CLR}$ are the proposed consistent latent space distance and real image reconstruction distance calculated through Eq. 6 and Eq. 8, λ_1 and λ_2 are weighting coefficient of the additional loss.

With this updated objective, we can stabilize GANs training in the following two aspects: First, we align the generator and discriminator during training phase by assigning a consistent latent representation task for the discriminator, which makes the game between generator and discriminator fairer. Second, we enable the generator to get access to the real data through a reconstruction task. This helps generator make full use of alternative information, leading to a more realistic output. The pseudo code of our method is shown in Algorithm 1. Extensive experiments result in Sec 4. showed the effectiveness of our method.

4 Experiments

To validate the effectiveness of the proposed method, we conducted extensive experiments on various GAN architectures and datasets. Sec. 4.1 first introduces the detailed settings of our experiments. In Sec. 4.2, we proved that our method achieves better quality and stability through quantitative evaluation on various image generation datasets. Sec. 4.3 includes ablation study to the proposed Consistent Latent loss and Reconstruction loss to better understand the designed components. In Sec. 4.4 we prove that we can get a better feature extractor by unsupervised classification. At last, we visualize the realness curve of D to validate the improved fairness of GANs in Sec. 4.5. The generated samples can be found in the Supplementary Material.

4.1 Experimental Setup

Datasets. For a more comprehensive evaluation of the proposed method, the model was tested on low resolution images and high resolution images separately. We used DCGAN [40] as the basic architecture for low resolution image

Table 1: FIDs of DCGAN [40] using different learning objectives on CelebA [33] and CIFAR-10 [25] dataset. The bold numbers indicate the best result for each dataset.

Learning Objective	CelebA	CIFAR-10
WGAN [1]	36.47	55.96
HingeGAN [54]	25.57	42.4
LSGAN [36]	30.76	42.01
DCGAN [40]	27.02	38.56
WGAN-GP [14]	70.28	41.86
Realness GAN-Obj.1 [49]	-	36.73
Realness GAN-Obj.2 [49]	23.51	34.59
Realness GAN-Obj.3 [49]	-	36.21
AdaptiveMix [32]	**12.43**	30.85
CLR-GAN(Ours)	13.63	**23.3**

generation, with CIFAR-10 [25] dataset and Celeba64×64 [33] dataset. For high-resolution image generation, we used StyleGAN-V2 [22] as the basic architecture, and test the generation ability under AFHQ Cat [8], LSUN Church [53], and FFHQ [21] dataset corresponding to animal generation, natural scene generation, and face generation respectively.

Evaluation. During evaluation phase, we mainly used the most common Frechet Inception Distance(FID) [15] for quantitatively represent the image generation quality. Besides, Precision & Recall [29] is also adopted to provide more details of the realness and diversity of the proposed model. In practice, we calculate FID between 20K generated images and all real samples on CIFAR-10 and CelebA dataset. While for AFHQ-Cat, LSUN-Church and FFHQ, we calculate the FID and P&R between 50K generated images and all real samples.

Other settings. For all baselines, our experimental hyperparameters were set according to the original paper. We used pre-trained Inception-V3 [43] as the feature extractor for FID and P&R calculation. For the weights of the additional loss, we set $\lambda_1 = 5$ and $\lambda_2 = 0.5$.

4.2 Performance on Image Generation

Low-resolution Image Generation. We first conducted experiments on low resolution image datasets to verify the effectiveness of the proposed method. We used DCGAN [40] as the baseline architecture and compared our method with some well-known learning objectives on CIFAR10 and CelebA datasets.

Tab. 1 presents the results. From the perspective of FID, we can directly see a significant margin between the proposed method and previous learning objectives. Furthermore, compared with the recently proposed learning objective [32], our method outperforms 24.5% on the CIFAR-10 dataset while remains a competitive result on CIFAR-10 dataset(8% behind), which further demonstrates the superiority of our model.

Table 2: Comparisons on AFHQ-Cat [8], FFHQ [21] and LSUN-Church [53] with different high-resolution image generative models. P and R denote precision and recall. The bold numbers indicate the best result for each dataset. The blue numbers indicate the improvements.

Method	AFHQ-Cat			FFHQ			LSUN Church		
	FID ↓	P ↑	R ↑	FID ↓	P ↑	R ↑	FID ↓	P ↑	R ↑
StyleGAN-V2 [22]	7.92	0.68	0.27	3.86	0.68	0.25	4.04	0.58	0.40
LC-Reg [46]	6.70	-	-	3.93	-	-	4.07	-	-
StyleGAN-V2-ADA [19]	6.05	0.66	0.25	4.01	0.66	0.26	4.01	0.61	0.43
StyleGAN-V2-APA [18]	4.88	0.65	0.30	3.75	0.67	0.29	3.92	0.60	0.43
StyleGAN-V2 + Ours	4.79(-3.13)	**0.76**	0.28	3.44(-0.42)	0.70	0.41	3.52(-0.52)	**0.63**	0.46
StyleGAN-V2-ADA + Ours	**4.45(-0.43)**	0.74	**0.33**	**3.37(-0.64)**	**0.71**	**0.44**	**3.43(-0.58)**	0.61	**0.48**

High-resolution Image Generation. To further demonstrate the effectiveness of the proposed method, we also considered the recently proposed high-resolution image synthesis architecture StyleGAN-V2 [22]. We compared our method with other StyleGAN-V2 variants methods on AFHQ-Cat, LSUN Church and FFHQ datasets, respectively. We also combined our method with recent proposed works that aims to improve GANs training called StyleGAN-V2-ADA [19] to show the compatibility of our method.

As shown in Tab. 2, there is a substantially improvement with the proposed method on StyleGAN-V2 for both datasets, outperforming other methods with a clear margin. We also found that combined with StyleGAN-V2-ADA, our method can get a further improvement, achieving a better results on various datasets. Therefore, our proposed method can improve the synthesis quality of high-resolution images while also integrating well with previous works.

As for the Precision and Recall. The improvements among multiple datasets are clear, indicating that our model can not only learn to generate more realistic images, but also enhance the diversity of generated images, to some extent solving the problem of insufficient diversity of conventional GANs. This may suggest that CLR-GAN can benefit from the proposed two constrains that leads to learning a wider distribution, thereby improving the diversity of generated samples. More validation metrics and experimental results on the more complex ImageNet [41] dataset are shown in the Supplementary Material.

4.3 Ablation Study

In the ablation experiment, we primarily focus on the impact of the two new training constrains we added during training. To ensure the generality, we selected both the low-resolution CIFAR-10 dataset and the high-resolution FFHQ dataset as two baseline datasets. Recall that we can use λ_1 and λ_2 to control the strength of the proposed constrains in Eq. 13. Therefore we can modify different λ values in the ablation study to figure out the optimal values of two constrains. We first set $\lambda_1 = \lambda_2 = 0$ to get the baseline values.

Table 3: Ablation studies of different weights λ_1 and λ_2 on low-resolution CIFAR-10 dataset and high-resolution FFHQ dataset. The best FID of each dataset are marked in bold.

λ_1	λ_2	FID
0	0	38.56(baseline)
1	0	25.15
0	1	26.09
5	1	25.20
10	1	25.17
15	1	26.30
5	5	25.55
5	0.5	**23.3**

(a) CIFAR-10 @ 32×32

λ_1	λ_2	FID
0	0	3.86(baseline)
1	0	3.75
0	1	3.77
5	1	3.62
10	1	3.84
15	1	4.01
5	5	3.80
5	0.5	**3.44**

(b) FFHQ @ 256×256

The FID scores are shown in Tab. 3. We can observe that when we set $\lambda_1 = 0$ or $\lambda_2 = 0$, the performance of the network significantly drops 9.96% on CIFAR-10 dataset. The performance even falls below the baseline model on high resolution dataset. Only when both the latent consistent loss and the reconstruction loss are combined together does the model gets its best performance. This proves that the two constrains we added are indispensable with each other.

Next we will find the optimal strength of the proposed constrains. We first fix $\lambda_2 = 1$ and change the value of λ_1 to find the ideal value of λ_1. During the experiments, we found that a relatively high λ_1 makes the model unable to converge, so we chose 15 as the maximum value. As shown in Tab. 3, $\lambda_1 = 5$ turns out to be the best. Consequently, we then fix $\lambda_1 = 5$ and search for the best λ_2. Ultimately, $\lambda_1 = 5$ and $\lambda_2 = 0.5$ turns out to be the best strategy.

During the search for optimal parameters, we observed some interesting phenomena. As shown in Fig. 3. Let $\eta = \frac{\lambda_1}{\lambda_2}$. We found a higher value of η leads to a better result in the early stage of training. However, an excessively high η might lead to poor convergence at the final stages of model training. We infer that when η is large, our model that constrained by the latent distance, can quickly converge to a distribution similar to the real distribution, hence leads to an early convergence. But overly tight constraints might prevent the model's ability to generalize effectively, resulting in a lack of diversity in the generated data thus making the final performance poor. The underlying causes of this phenomenon require further investigation in future research.

4.4 Performance on Unsupervised Visual Recognition

GANs are often regarded as an unsupervised training method. To further demonstrate the effectiveness of the proposed method, we did experiment to show our proposed method leads to a better feature extractor. We take the pre-trained discriminator as a feature extractor and add a linear classification head to perform

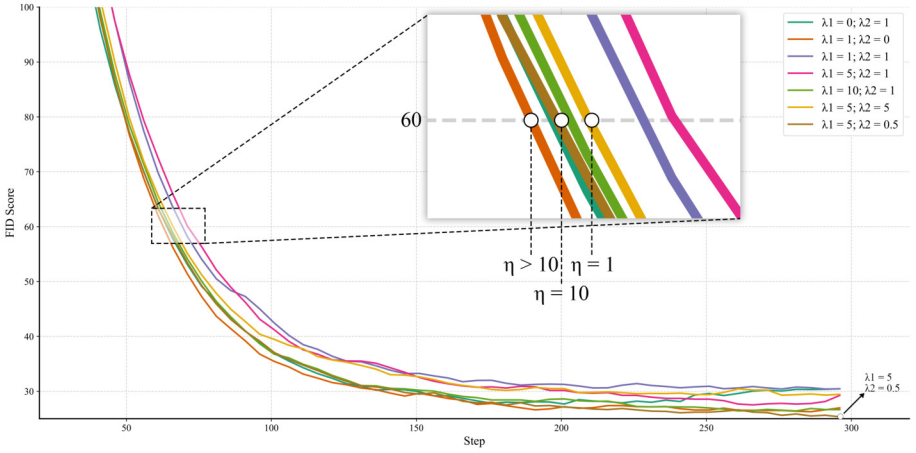

Fig. 3: Training curves at different η

Table 4: Classification accuracy on CIFAR-10 dataset [25] with various methods with linear classifier.

Method	#Parameters	Accuracy
ViT-H [10]	632M	**99.5%**
EfficientNetV2-M [44]	55M	99%
PyramidNet [28]	26M	98.6%
NAT-M4 [34]	6.2M	98.4%
HCGNet [52]	3.1M	97.7%
CLR-GAN(Ours)	**3.02M**	98.8%

classification task on the CIFAR-10 dataset. We compared our feature extractor with other classification models. The results are shown in Tab. 4.

As listed in Tab. 4. Our proposed pre-train feature extractor can reach a relatively high accuracy after simple fine-tuning. Compared with models which have similar parameter sizes ($\leq 10M$) [34,52], our model is 0.4% and 1.1% higher respectively. In comparison with models several orders of magnitude higher than our parameter size [10], our model is also very competitive, only 0.7% lower than the best model. This indicates that our strategy can not only improve the quality and diversity of generated images in generative adversarial networks, but also serve as an effective unsupervised training method, benefiting downstream tasks.

4.5 Towards a fairer game

As we mentioned in the introduction part, the fundamental purpose of our method is to produce more realistic and diverse images by making the training of GANs fairer. By the proposed consistent latent representation and reconstruction, we can shrink the distance between the generator and discriminator.

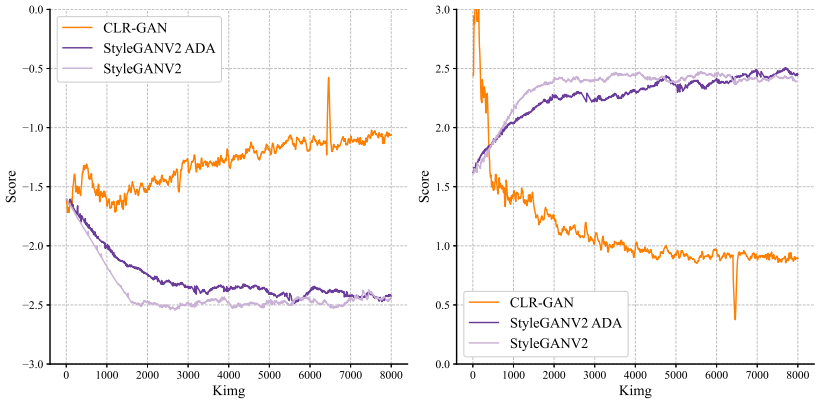

Fig. 4: Realness scores of the real samples and synthesis samples by various discriminators during training. We visualize the realness score of StyleGAN-V2 [22], StyleGAN-V2-ADA [19] and the proposed CLR-GAN.

In this section, we demonstrate the enhanced fairness of our method through experiments.

In order to better quantitatively judge the fairness of our model, we followed the previous two works [2, 47], using the score of the discriminator to represent the realness of the image. We visualize the realness score the model trained on LSUN-Church. We used exponentially weighted averages [17] on the original data for better visualization. The left part of Fig. 4 shows the score of the synthesis images, while the right part shows the score of the real images. We can clearly see that using our CLR-GAN, the realness gap between the generated image and the real image is much smaller than the baseline. While for StyleGAN-V2-ADA, the gap between scores is similar to the baseline, so it only improves FID and can not make the game fairer. We can then draw a conclusion that with the aid of the proposed auxiliary objective functions, the realness scores of real images and generated images become closer, which means that CLR-GAN can improve the realness and diversity of generated images by improving training fairness.

5 Conclusion

In this paper, we proposed a simple, effective and plug and play new objective function for GANs training. The new objective function simultaneously constrains the output distribution of the generator and the discriminator to make them more consistent. Thereby making the competition between the generator and the discriminator more fair. Eventually improving the quality and diversity of generated images. In addition to image generator, we also demonstrated that the discriminator with constrain is an efficient unsupervised feature extractor. Experimental results demonstrate that our proposed method improves the performance of baseline models over various datasets.

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International conference on machine learning. pp. 214–223. PMLR (2017)
2. Bai, Q., Yang, C., Xu, Y., Liu, X., Yang, Y., Shen, Y.: Glead: Improving gans with a generator-leading task. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12094–12104 (June 2023)
3. Bar-Tal, O., Ofri-Amar, D., Fridman, R., Kasten, Y., Dekel, T.: Text2live: Text-driven layered image and video editing. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision – ECCV 2022. pp. 707–723. Springer Nature Switzerland, Cham (2022)
4. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis (2019)
5. Ceylan, D., Huang, C.H.P., Mitra, N.J.: Pix2video: Video editing using image diffusion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 23206–23217 (October 2023)
6. Chan, E.R., Monteiro, M., Kellnhofer, P., Wu, J., Wetzstein, G.: pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5799–5809 (2021)
7. Chenthamarakshan, V., Das, P., Hoffman, S., Strobelt, H., Padhi, I., Lim, K.W., Hoover, B., Manica, M., Born, J., Laino, T., Mojsilovic, A.: Cogmol: Target-specific and selective drug design for covid-19 using deep generative models. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 4320–4332. Curran Associates, Inc. (2020)
8. Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: Stargan v2: Diverse image synthesis for multiple domains. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8188–8197 (2020)
9. Deng, Y., Tang, F., Dong, W., Ma, C., Pan, X., Wang, L., Xu, C.: Stytr2: Image style transfer with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11326–11336 (June 2022)
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale (2021)
11. Farnia, F., Ozdaglar, A.: Do GANs always have Nash equilibria? In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 3029–3039. PMLR (13–18 Jul 2020)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS. vol. 27 (2014)
13. Gu, J., Shen, Y., Zhou, B.: Image processing using multi-code gan prior. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
14. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. Advances in neural information processing systems **30** (2017)
15. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)

16. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 6840–6851. Curran Associates, Inc. (2020)
17. Hunter, J.S.: The exponentially weighted moving average. Journal of Quality Technology **18**(4), 203–210 (1986)
18. Jiang, L., Dai, B., Wu, W., Loy, C.C.: Deceive d: Adaptive pseudo augmentation for gan training with limited data. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems (NeurIPS). vol. 34, pp. 21655–21667. Curran Associates, Inc. (2021)
19. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data (2020)
20. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems. vol. 34, pp. 852–863. Curran Associates, Inc. (2021)
21. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4401–4410 (2019)
22. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8110–8119 (2020)
23. Kim, H., Choi, Y., Kim, J., Yoo, S., Uh, Y.: Exploiting spatial dimensions of latent in gan for real-time image editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 852–861 (June 2021)
24. Kingma, D.P., Welling, M.: Auto-encoding variational bayes (2022)
25. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
26. Kurach, K., Lučić, M., Zhai, X., Michalski, M., Gelly, S.: A large-scale study on regularization and normalization in GANs. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 3581–3590. PMLR (09–15 Jun 2019)
27. Kwon, G., Ye, J.C.: Clipstyler: Image style transfer with a single text condition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 18062–18071 (June 2022)
28. Kwon, J., Kim, J., Park, H., Choi, I.K.: Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks (2021)
29. Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., Aila, T.: Improved precision and recall metric for assessing generative models. Advances in Neural Information Processing Systems **32** (2019)
30. Lee, G., Kim, H., Kim, J., Kim, S., Ha, J.W., Choi, Y.: Generator knows what discriminator should learn in unconditional gans. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision – ECCV 2022. pp. 406–422. Springer Nature Switzerland, Cham (2022)
31. Liao, W., Hu, K., Yang, M.Y., Rosenhahn, B.: Text to image generation with semantic-spatial aware gan. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 18187–18196 (June 2022)
32. Liu, H., Zhang, W., Li, B., Wu, H., He, N., Huang, Y., Li, Y., Ghanem, B., Zheng, Y.: Adaptivemix: Improving gan training via feature space shrinkage. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16219–16229 (June 2023)

33. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of the IEEE international conference on computer vision. pp. 3730–3738 (2015)
34. Lu, Z., Sreekumar, G., Goodman, E., Banzhaf, W., Deb, K., Boddeti, V.N.: Neural architecture transfer. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(9), 2971–2989 (Sep 2021)
35. Luo, S., Guan, J., Ma, J., Peng, J.: A 3d generative model for structure-based drug design. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems. vol. 34, pp. 6229–6239. Curran Associates, Inc. (2021)
36. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2794–2802 (2017)
37. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks (2018)
38. Mukherjee, S., Asnani, H., Lin, E., Kannan, S.: Clustergan: Latent space clustering in generative adversarial networks. Proceedings of the AAAI Conference on Artificial Intelligence **33**(01), 4610–4617 (Jul 2019)
39. Nguyen-Phuoc, T., Li, C., Theis, L., Richardt, C., Yang, Y.L.: Hologan: Unsupervised learning of 3d representations from natural images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7588–7597 (2019)
40. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
41. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**, 211–252 (2015)
42. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
43. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision (2015)
44. Tan, M., Le, Q.: Efficientnetv2: Smaller models and faster training. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 10096–10106. PMLR (18–24 Jul 2021)
45. Tran, N.T., Bui, T.A., Cheung, N.M.: Dist-gan: An improved gan using distance constraints. In: Proceedings of the European Conference on Computer Vision (ECCV) (September 2018)
46. Tseng, H.Y., Jiang, L., Liu, C., Yang, M.H., Yang, W.: Regularizing generative adversarial networks under limited data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7921–7931 (June 2021)
47. Wang, J., Yang, C., Xu, Y., Shen, Y., Li, H., Zhou, B.: Improving gan equilibrium by raising spatial awareness. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11285–11293 (June 2022)
48. Wu, J.Z., Ge, Y., Wang, X., Lei, S.W., Gu, Y., Shi, Y., Hsu, W., Shan, Y., Qie, X., Shou, M.Z.: Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7623–7633 (October 2023)

49. Xiangli, Y., Deng, Y., Dai, B., Loy, C.C., Lin, D.: Real or not real, that is the question. arXiv preprint arXiv:2002.05512 (2020)
50. Xu, Y., AlBahar, B., Huang, J.B.: Temporally consistent semantic video editing. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision – ECCV 2022. pp. 357–374. Springer Nature Switzerland, Cham (2022)
51. Yang, C., Shen, Y., Xu, Y., Zhao, D., Dai, B., Zhou, B.: Improving gans with a dynamic discriminator. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems. vol. 35, pp. 15093–15104. Curran Associates, Inc. (2022)
52. Yang, C., An, Z., Zhu, H., Hu, X., Zhang, K., Xu, K., Li, C., Xu, Y.: Gated convolutional networks with hybrid connectivity for image classification. Proceedings of the AAAI Conference on Artificial Intelligence **34**(07), 12581–12588 (Apr 2020)
53. Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015)
54. Zhao, J., Mathieu, M., LeCun, Y.: Energy-based generative adversarial network. arXiv preprint arXiv:1609.03126 (2016)
55. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (Oct 2017)