Neurocomputing 323 (2019) 363–372

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

# Improved generative adversarial networks with reconstruction loss

Yanchun Li[a,*], Nanfeng Xiao[a], Wanli Ouyang[b]

[a] *School of Computer Science and Engineering, South China University of Technology, Guangzhou, China*
[b] *School of Electrical and Information Engineering, University of Sydney, Sydney, Australia*

ARTICLE INFO

*Article history:*
Received 28 May 2018
Revised 27 September 2018
Accepted 4 October 2018
Available online 11 October 2018

Communicated by Dr. Shen Jianbing Shen

*Keywords:*
Generative adversarial networks (GAN)
Image generation
Reconstruction loss
Deep generative model

ABSTRACT

We propose a simple regularization scheme to handle the problem of mode missing and unstable training in the generative adversarial networks (GAN). The key idea is to utilize the visual features learned by the discriminator. We reconstruct the real data by feeding the generator with the real data features extracted by the discriminator. A reconstruction loss is added to the GAN's objective function to enforce the generator can reconstruct from the features of the discriminator, which helps to explicitly guide the generator towards to near the probable configurations of real data. The proposed reconstruction loss improves the performance of GAN, produces higher quality images on different dataset, and can be easily combined with other regularization loss functions such as gradient penalty to improve the performance of various GANs. We conducted experiments on the widespread adopted architecture DCGAN and the complicated ResNet architecture across different datasets, the results of which show the effectiveness and robustness of our proposed method.

## 1. Introduction

Generative model is about learning the unknown distribution from which the observed data are sampled and producing new different samples based on the learned distribution. Generative model can be used for supervised learning, such as classification, and unsupervised learning, such as independent sample generation. Generative adversarial networks (GAN) [1] is a powerful generative model and has shown success on many specific tasks: image synthesis [2–5], image super resolution [6], image to image translation [7,8], video prediction [9] and so on.

Typically, a GAN includes two deep networks: a generator and a discriminator. The generator maps data from a simple distribution to the real data space, and the discriminator distinguishes real date from generated data. The discriminator and the generator are optimized by the adversarial training: the generator tries to produce data as realistic as possible to confuse the discriminator, while the discriminator tries to discriminate the real data from the generated one. Ideally the optimization ends when the Nash equilibrium is reached.

GAN is a very flexible model, which directly maps the random noise into plausible data samples. But this flexibility accompanies with training instability and mode missing, which is a common failure pattern where large volumes of probability mass are collapsed onto a few modes in the generator. In order to handle the problems above, it is desirable to explicitly guide the generator towards to the real data manifold. Meanwhile, we also hope this method do not introduce significant computational overhead or influence the GAN's powerful generation ability. Another motivation is that since the discriminator has learned very useful information about the real data [3], it is reasonable to use this information to guide the generator learning.

Based on the observation above, we add reconstruction loss to the objective cost function of GAN. We utilize the features learned by the discriminator to reconstruct real data through the generator, and then add the reconstruction loss to objective function of both the generator and discriminator. As the features come from the discriminator not the extra networks, the proposed method adopts two networks as in the original GAN. To reconstruct real data, the discriminator needs to provide informative features to the generator, which can implicitly improve the ability of the discriminator to differentiate real data from generated data. On the other hand, the generator needs to reconstruct every real data, thus all modes of real data pass through the generator, which can guide the generator towards to near the manifold of real data in all modes. In addition, the generator's optimization signal comes from both the discriminator's classification result and the reconstruction loss, thereby avoiding the no gradient problem and improving the training stability.

[*] Corresponding author. Yanchun Li.
*E-mail addresses:* aserwer@163.com, 201510105258@mail.scut.edu.cn (Y. Li).

https://doi.org/10.1016/j.neucom.2018.10.014
0925-2312/© 2018 Elsevier B.V. All rights reserved.

In summary, the main contributions are as follows:

- We propose a simple, effective and robust method, denoted as “GAN-RL”, to improve the performance of GAN.
- We evaluate “GAN-RL” with metrics Inception Score (IS) and Fréchet Inception Distance (FID). By implementing and comparing the proposed method with other approaches across different datasets with various resolution, we demonstrate that our method outperforms other methods or at least can compete with other methods.
- The proposed method can be easily combined with other methods to improve the performance of various GANs, and a series of experiments were conducted to prove this.

## 2. Related works

Driven by the rapid development of deep learning, a variety of network architectures and their applications have been put forward, such as object detection [10,11], object tracking [12], photo cropping [13,14], visual attention prediction [15], visual fashion understanding [16,17], generative models [1,18,19] and so on. Most of these models are based on the fully convolutional network and generally give very impressive results. Next, we first outline the improved methods of GAN from different aspects, and then summarize several very related methods.

### 2.1. Overview of improvement methods

Many recently works have been proposed to solve the problem of GAN’s training instability and mode missing, which can be roughly classified into three categories: adding supervision condition such as classification information, hybrid with additional networks and optimizing the networks with different objective functions. The first category can be viewed as supervision GAN, including Semi-GAN, C-GAN, Info-GAN [20–22] and so on. The second category is always associated with other networks such as Auto-encoder, including energy-based GAN, BEGAN, DFM, VAE-GAN, MRGAN, $\alpha$-GAN [23–28] and so on. The third category neither requires additional information nor changes the network architecture, but adopts different loss function, including LSGAN, McGAN, WGAN, WGAN-GP, AGE, DRAGAN, etc. [29–34], which can be seen as the GAN’s variants. These methods have more or less disadvantages. Supervision methods required difficult and expensive classification information. Hybrid methods required multiple networks (more than two) to be optimized simultaneously, resulting in time-consuming and high computational complex. While for GAN’s variants, such as WGAN underused the ability of the discriminator, WGAN-GP introduced extra computational overhead, and AGE was weakness on produce high quality images. This paper focuses on fully unsupervised GAN.

### 2.2. Summary of relevant methods

The most related work is AGE [33], which also employed two networks and combined adversarial loss with reconstruction loss of real data. The main difference is that the adversarial loss of AGE was in the feature space: the encoder attempted to make the feature space of real data draw from the prior distribution and the feature space of generated data do not match the prior distribution, meanwhile the generator attempted to produce data whose feature space matched the prior distribution. Another difference with our method is that AGE augmented the objective function with reconstruction loss of the latent codes. The adversarial loss of AGE needed to compute the mean and variation of every sample, which leaded to slow training. Another main shortage of AGE was the generated data of low quality. We conjecture that this may be because the adversarial loss in AGE is between the posterior distribution of real data/generated data and the prior distribution, rather than directly between the real data distribution and the generated data distribution like in original GAN.

From the perspective of adopting generator to reconstruct real data, the related works include MR-GAN [27] and $\alpha$-GAN [28]. Both MR-GAN and $\alpha$-GAN contained a generator, an encoder and two discriminators. Their generators had the same goals as our generator, which were to produce real-like data and decode training data. The difference between MR-GAN and $\alpha$-GAN was that the other three networks had different objective functions. Specifically, the encoder optimization signal of MR-GAN only came from the real data reconstruction loss, while the encoder optimization signal of $\alpha$-GAN included reconstruction loss of real data and judgment loss of latent space discriminator. In MR-GAN, one discriminator separated real data and their reconstruction, and the other discriminator separated the reconstruction of real data and the generated data. In $\alpha$-GAN, the data space discriminator differentiated training data, reconstruction data and the generation, while the latent space discriminator differentiated whether the codes came from posterior or prior. Due to the need to optimize four networks simultaneously, both MR-GAN and $\alpha$-GAN were time-consuming with high computational complexity.

The adversarial training in the above mentioned approaches are not directly between the generated data and training data like in standard GAN. The adversary in our method is between training distribution and generated distribution, and improves the performance with an additional regularization. From this viewpoint, the related works are DFM, WGAN-GP, DRAGAN [25,32,34]. DFM augmented the objective function of the generator with a denoising feature loss, and the denoising signal came from training an extra network on the feature space of real samples. WGAN-GP improved the performance by adding a gradient penalty of linear interpolation between real and generated samples to the objective function of the discriminator. DRAGAN proposed to add regularization based on local perturbations of the input real samples to the objective function of the discriminator. In experimental Section 6.4, we will illustrate that our method can improve the performance of GAN-GP (similar to WGAN-GP) and DRAGAN.

## 3. Background

In this section, we briefly introduce the Adversarial Generative Networks and analyze its disadvantages.

### 3.1. Adversarial generative networks

Let $D^{\phi}$ denote the discriminator with parameters $\phi$, and $G^{\theta}$ denote the generator with parameters $\theta$. The generator $G^{\theta}$ maps the random noise $z$ to the data space, while the discriminator $D^{\phi}$ identifies whether its input is training or generated data. The original GAN [1] adopted Jensen–Shannon divergence (JSD) to measure the distance between data distribution and generation distribution. The cost function for the discriminator is defined as:

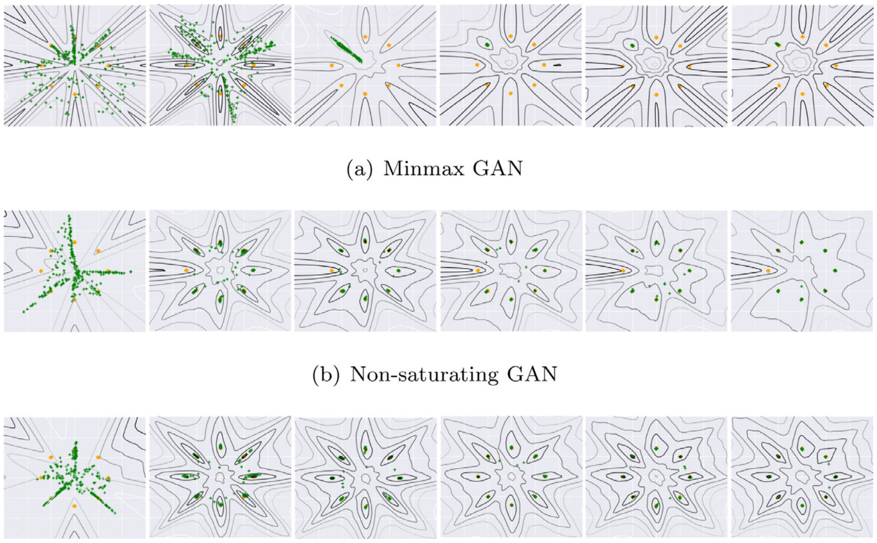
$$L_{D}^{\phi} = \mathbb{E}_{x\sim p_x}[-log(D^{\phi}(x))] + \mathbb{E}_{z\sim p_z}[-log(1-D^{\phi}(G^{\theta}(z)))] \tag{1}$$

The cost function for the generator has two forms:

$$Minimax: L_{G}^{\theta} = \mathbb{E}_{z\sim p_z}[log(1-D^{\phi}(G^{\theta}(z)))] \quad or \tag{2a}$$

$$Non-saturating: L_{G}^{\theta} = \mathbb{E}_{z\sim p_z}[-log(D^{\phi}(G^{\theta}(z)))] \tag{2b}$$

where $p_x$ and $p_z$ are the training distribution and a simple noise distribution, respectively. The theoretical analysis in [1] was based on a zero-sum game in which the generator adopted Minimax formulation (Eq. (2a)) as cost function. But in practice, Goodfellow



(a) Minmax GAN

(b) Non-saturating GAN

(c) GAN-RL(Ours)

**Fig. 1.** 8 Gaussian mode experiment (different phases of training: each column represents the state of the three methods per 5000 training iterations). (a) Minimax GAN (top), (b) Non-saturating GAN (middle), and (c) GAN-RL (bottom, ours). Real samples are marked orange and generated samples are green. Value surfaces of $D^{\phi}(x)$ are shown in the background. Note that we add the reconstruction loss to Non-saturating GAN in (c) GAN-RL. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

et al. [1] recommend Non-saturating formulation (Eq. (2b)) as an alternative cost function for the Non-saturating formulation providing more stable gradient. We adopt Non-saturating objective (Eq. (2b)) for the generator in all our experiments if without specified.

### 3.2. Disadvantages of original GAN

According to the objective function Eq. (1), optimizing the discriminator equals to training a characteristic function in data space, which attempts to assign value 1 to real samples and 0 to fake samples [1–3]. We train the generator via the generated samples' classification signals provided by the discriminator, and hope the generation distribution move towards the real data distribution. But when alternately training these two networks, the generation distribution varies with training, therefore we have no control over the shape of the discriminator in between two fixed distributions like in supervised binary classification function. In practice, if the distributions do not have substantial overlap, the gradients may point to random directions, thus resulting in training instability or no gradient for generator optimizing [31].

Another serious disadvantage of training GAN is the mode missing. In order to illustrate this phenomenon, we conducted experiments on 8 normal distribution modes. The results are shown in Fig. 1. From the Fig. 1(a) and (b), we observe that the Minimax GAN (the generator employs Minimax form cost function) only learned one mode and the Non-saturating GAN (the generator employs Non-saturating form cost function) miss one mode. In Minimax GAN, the generator tries to produce data that confuse the discriminator to judge them not fake, therefore the generator only needs to produce a few modes can achieve this target. Like in Fig. 1(a), the generator produced one mode and met the requirement of not fake samples. In Non-saturating GAN, the generator tries to produce data that confuse the discriminator to judge them real, and may produce most of (not all) modes to achieve this target. Like in Fig. 1(b), the generator produced 7 modes in 8 modes. Both the Minimax GAN and the Non-saturating GAN will not be penalized for missing modes.

## 4. Generative adversarial networks with reconstruction loss

### 4.1. Reconstruction loss

Our method is based on two foundations: the discriminator can learn the hierarchy of features about the training data [3]; the architecture of the generator adopted by GAN doesn't significantly from other deep generative methods such as VAE [19], which is based on approximating the likelihood distribution of real data. Thus a natural idea is to utilize the features learned by the discriminator to reconstruct real data through the generator, and then add the reconstruction loss to the GAN loss to improve the training stability and suppress the mode missing. Following is the formulation of reconstruction loss:

$$L_X^{\phi,\theta} = \mathbb{E}_{x\sim p_x}[\|G^{\theta}(D_F^{\phi}(x)) - x\|_1] \tag{3}$$

Where $D_F^{\phi}$ is part of the discriminator (except the last layer) and transforms the data to feature space. $G^{\theta}(D_F^{\phi}(x))$ in Eq. (3) means that $D_F^{\phi}$ encodes training data to features and the generator $G^{\theta}$ decodes the features to the training data. Augment the GAN objective function with $L_1$ loss, the loss function of our model "GAN-RL" are specified as:

$$Discriminator\ loss : L_D^{\phi} + \lambda * L_X^{\phi} \tag{4a}$$

$$Generator\ loss : L_G^{\theta} + \lambda * L_X^{\theta} \tag{4b}$$

Where $\phi$, $\theta$ denote the parameters of the discriminator and the generator, respectively, and $\lambda$ controls the weight of reconstruction loss when training the GAN. From Eq. (4), both the generator and the discriminator have two targets: the discriminator differentiates real data from generated data and provides informative features for reducing the reconstruction loss simultaneously; the generator reconstructs real data and produces samples as real as possible. The optimization of GAN is performed alternatively. When optimizing $D^{\phi}$, $G^{\theta}$ is fixed, so we only mark out $\phi$ in Eq. (4a), and it is the same reason in Eq. (4b).

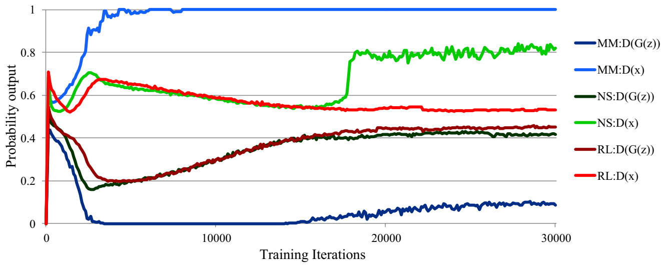


**Fig. 2.** In 8 Gaussian mode experiment, the probability outputs of the discriminator to fake samples and real samples. MM, NS and RL are abbreviations of Minimax GAN, Non-saturating GAN and GAN-RL(Ours), respectively. $x$ and $G(z)$ represent real and fake samples, respectively. Note that we omit the parameters. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

### 4.2. Effectiveness of the proposed regularization

One question is why the reconstruction loss can improve the performance of GAN. Let $x_f = D_F^{\phi}(x)$, $p_{x_f}$, respectively, denote the features captured by the discriminator and their distribution. Let $\hat{x} = G^{\theta}(x_f)$ denote the reconstruction of real data by the generator. According to [28] a zero-mean Laplace distribution with the scale parameter $\gamma$ can be chosen as a likelihood of $p(x|x_f)$, and the mathematical formulation is defined as follows:

$$p(x|x_f) \propto exp(-\gamma\|x-\hat{x}\|_1) \tag{5}$$

Based on Jensen's inequality, we have

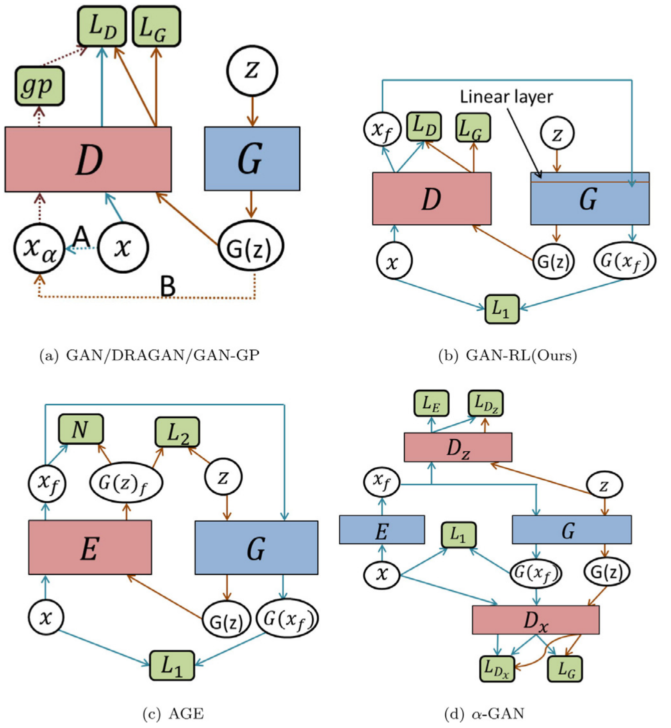
$$log(p(x)) = log(\mathbb{E}_{p_{x_f}}[p(x|x_f)]) \geq \mathbb{E}_{p_{x_f}}[log(p(x|x_f))] \tag{6a}$$

$$\mathbb{E}_{p_{x_f}}[log(p(x|x_f))] \propto \mathbb{E}_{p_{x_f}}[-\gamma\|x-\hat{x}\|_1] \tag{6b}$$

Thereby, according to the Eqs. (5), (6a) and (6b), the $L_1$ reconstruction loss $\|x-\hat{x}\|_1$ is an estimation of $log(p(x))$. This is a highly popular choice and used in many GANs, such as AGE, BEGAN, cycle GAN, PPGN and $\alpha$-GAN [8,24,28,33,35].

Augment GAN objective function with reconstruction loss means that the whole networks have an explicit estimation of log-density of real-data, thus can improve the training stability and discourage mode missing. The main cause of training hard is the discriminator perfectly apart from the training data and the generated data, so the generator receives no gradient to optimize itself [31]. As show in Eq. (4b), the generator in our method receives two kinds of optimizing signals, one comes from the classification results of discriminator like in original GAN, and the other is the $L_1$ reconstruction loss of real data, thus can effectively prevent the no gradient problem and training instability. In addition, as the reconstruction is based on the features provided by the discriminator, the discriminator needs to capture informative features from real data, which further improves the ability of the discriminator to apart real data from generated data.

In 8 Gaussian mode experiment, we plot the probability outputs of the discriminator to both generated samples and real samples over the course of training in Fig. 2. For Minimax GAN (two blue lines), the discriminator judges the real samples as having probability close to 1 to be real and judges the fake samples as having probability close to 0 to be real, which is correct for both cases and results in no gradient problem. For Non-saturating GAN (two green lines), the value assigned to real sample by the discriminator are oscillating (bright green line), which means the training is unstable. For GAN-RL (two red lines), the probabilities of real samples and the probabilities of fake samples are getting closer and closer with the number of training iterations, and tend to be stable, which indicates that the GAN-RL can provide stable and effective gradient signals for both the discriminator and the generator.



(a) GAN/DRAGAN/GAN-GP

(b) GAN-RL(Ours)

(c) AGE

(d) $\alpha$-GAN

**Fig. 3.** Data flow charts for various models. Note that, the graphic (a) includes the data flow of GAN, DRAGAN and GAN-GP three models: GAN does not includes dot line, while DRAGAN includes dot line except B, and GAN-GP includes dot lines except line A. Data flows are distinguished by the colored lines.

Another advantage of our method is to discourage mode missing. As show in Eq. (3) $G^{\theta}(D_F^{\phi}(x))$, the generator needs to reconstruct every training data, thus low probability and high probability modes have the same chance to pass through the generator, which encourages the generator to produce all modes of real data manifold [27]. In the 8 Gaussian mode experiment, the Fig. 1(c) is the results of our method, which shows that all modes have been learned. In the experimental section, we will demonstrate that our method generates high quality and diversity images, and obtains excellent quantitative score.

Easy to implement and can be added to various GANs are third advantages of our method. We utilize the features (the penultimate layer) learned by the discriminator as the generator's extra inputs and calculate the reconstruction loss, which only increase a bit of computational overhead and do not require changing the original model architectures. In Section 6.4, we prove that our method can improve the performance of various GANs.

Fig. 3 summarizes the data flow charts of our model and other models that we will compare in our experiments. In Fig. 3 $G(z)$, $G(x_f)$, $x_f$ each denotes the generation, the reconstruction and the features of real data. In Fig. 3(a), $gp$ represents the gradient penalty, and $x_\alpha$ represents the linear interpolation between real data and its perturbation for DRAGAN or the line interpolation between real data and fake data for GAN-GP. In Fig. 3(b), when reconstructing real data, $x_f$ skips the linear layer and is the input of the second layer of the generator. In Fig. 3(c), $N$, $G(z)_f$ each denotes the normal distribution, the features of generated data. $D_x$ and $D_z$ of Fig. 3(d) represent the data space discriminator and the latent space discriminator, respectively.

## 5. Quantify metrics

Evaluating generative models is challenging [36]. As GAN directly produces data without likelihood, evaluating GAN is to

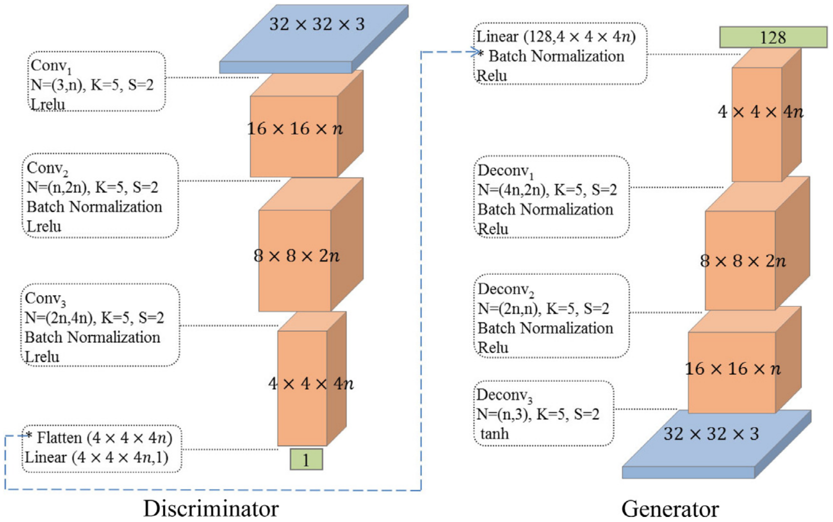


**Fig. 4.** The DCGAN architecture of the generator and the discriminator for generating $32 \times 32 \times 3$ resolution images. "Conv/Deconv $N = (n_1, n_2), K = k, S = s$" denote a Convolution/Deconvolution layer with $k \times k$ kernel, $n_1$ input filters, $n_2$ output filters and stride $= s$. "Batch Normalization" means that the layer is followed by a batch normalization layer. "Relu and Lrelu" denote the activation functions are Relu and LeakyRelu respectively. "Linear" means that the layer performs linear operation, and "Flatten" means that the inputs are reshaped to one dimensional tensors. The light blue dotted line indicators that the output of flatten layer of the discriminator is the input of the batch normalization layer of the generator when reconstructing the real data. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

evaluate the generated data. Multiple proxy metrics have been proposed, and we introduce two of them in this section. In next experimental section, we will employ these two metrics to assess the image quality and diversity.

*Inception score (IS).* Proposed by Salimans et al. [37]. The Inception Score usually adopts an Inception Net [38], which was pretrained on ImageNet dataset, to capture the meaningfulness and diversity of the generated samples. The generated data hope to have a low entropy for condition label distribution and a high entropy for the marginal distribution $\int_z p(y|x = G(z))d_z$, and the Inception Score is the exponent of the average KL divergence between these two distributions:

$$IS(G) = exp(\mathbb{E}_{x \sim G(z)}[KL(p(y|x) \| p(y))]) \tag{7}$$

The IS has two major drawbacks, one is that it is only suitable for labeled data, and the other is that it cannot detect the mode dropping in a class, i.e. the models that generate different images and generate the same images in a class have the same IS. As IS is widely adopted, to compare with other methods, we still use IS to evaluate the generated images on CIFAR-10 [39].

*Fréchet Inception Distance (FID).* Proposed by Heusel et al. [40]. FID calculates the distance between the feature space of training data and that of generated data, where the feature space is a specific layer of a pre-trained Inception Net. Specifically, the training data and generated data are feed into the Inception Net to obtain their feature space (a specific layer) respectively. Then, the specific layer is regard as a continuous multivariate Gaussian, and the FID is the distance between these two Gaussian distributions, which can be calculated through their means and covariances. The mathematical formulation is:

$$\mathrm{FID}(x, x^*) = \|\mu_x - \mu_{x^*}\|_2^2 + \mathrm{Tr}(C_x + C_{x^*} - 2(C_x C_{x^*})^{\frac{1}{2}}) \tag{8}$$

where $(\mu_x, C_x)$, and $(\mu_{x^*}, C_{x^*})$ are the mean and covariance of the sample embeddings from the data distribution and model distribution, respectively. In [40], the authors showed that the FID was correlated well with human judgment and was very sensitive to various distortions of image. Unlike IS, FID can detect intra-class mode dropping, i.e. the model that generates very similar images has a bad FID. FID is the main metric that we will adopt to evaluate the quality and diversity of the generation across different datasets in experiments.

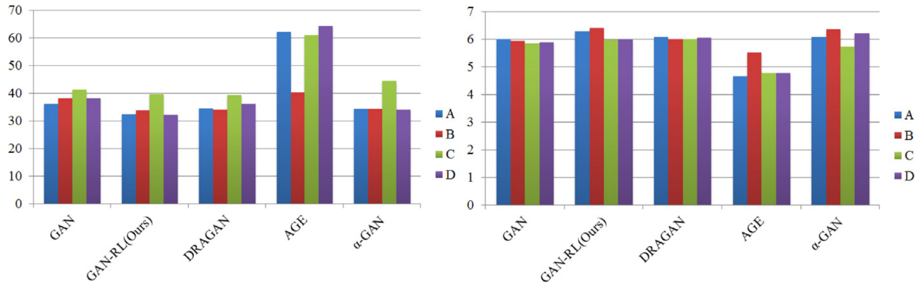
## 6. Experiments

In this section, we will conduct a series of experiments to assess the performance of our proposed method. In Section 6.1, on CIFAR-10 [39], we test the optimal hyperparameters settings for the compared five methods and then compare their performance under their respectively optimal setting. Next in Section 6.2, we choose the optimal value for $\lambda$ and test the architecture robustness of our method. Then we demonstrate the performance of our model on SVHN and LSUN dataset [41,42] in Section 6.3. In Section 6.4, we implement contrast experiments across three datasets to show our method on improving the performance of various approaches.

To reconstruct the real data, the features (the penultimate layer of the discriminator) learned from the discriminator are the input for the second layer of the generator. In other words, the generator has two kinds of inputs, one is the random noise, which is used to produce real-like samples as in typical GAN, and the other comes from the real data features, which is used for reconstruction. To clearly illustrate this process, we draw the framework of the DCGAN [3] in Fig. 4, in which the light blue dotted line expresses that when reconstructing real data, the output of flatten layer of the discriminator is the input of the generator, which also means that when the generator reconstructs the real data, it skips the linear layer. Please note that we do not make any changes to the network structure.

*Hyperparameters.* We used the Adam optimizer [43] with $\beta_1 = 0.5$. For low resolution images, the batch size is 64, and for images with resolution of $128 \times 128 \times 3$, the batch size is 16. All models were trained for 200k iterations. The coefficient of gradient penalty for DRAGAN and GAN-GP equals to 10, and the coefficient of reconstruction loss $\lambda = 5$ if not specified. Like in AGE and $\alpha$-GAN, we update the generator twice for each discriminator update. For AGE, we implement the program provided by the authors in open resource Github[1]. Fréchet Inception Distance (FID) is calculated over

[1] https://github.com/DmitryUlyanov/AGE.



(a) FIDs (lower is better)

(b) ISs (higher is better)

**Fig. 5.** FIDs (left) and ISs (right) on CIFAR-10 for different methods under different hyperparameter settings. Note: for IS, higher is better, and for FID, lower is better.

**Table 1**
Hyperparameter settings for five methods: GAN, GAN-RL(Ours), DRAGAN, $\alpha$-GAN and AGE. Note that: LR is the abbreviation of learning rate, $\beta_1$ and $\beta_2$ are the first and second order momentum parameters of Adam optimizer.

| Settings | LR | $\beta_1$ | $\beta_2$ |
|---|---|---|---|
| A | 0.0002 | 0.5 | 0.9 |
| B | 0.0002 | 0.5 | 0.999 |
| C | 0.0001 | 0.5 | 0.9/0.999 |
| D | 0.0005 | 0.5 | 0.9/0.999 |

**Table 2**
The optimal hyperparameter settings for five methods: GAN, GAN-RL(Ours), DRAGAN, $\alpha$-GAN and AGE.

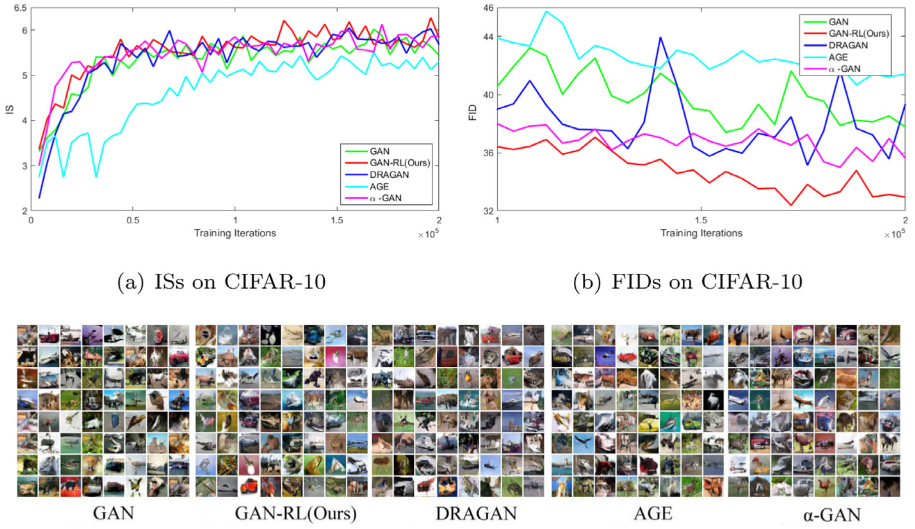
| Methods | GAN | GAN-RL(Ours) | DRAGAN | AGE | $\alpha$-GAN |
|---|---|---|---|---|---|
| lr | 0.0002 | 0.0002 | 0.0002 | 0.0002 | 0.0005 |
| $\beta_2$ | 0.9 | 0.9 | 0.9 | 0.999 | 0.999 |

20k generated samples, and Inception Score (IS) is calculate over 1k generated samples in Fig. 6(a).

### 6.1. Hyperparameter tests on CIFAR-10

On CIFAR-10, we tested 4 settings to evaluate the robustness of each method to learning rate and second order momentum parameter $\beta_2$ of Adam. First, we fixed the learning rate and selected the better one from $\beta_2$ equal to 0.9 or 0.999, and then based on the previous selection, we changed the learning rate. The details of the four hyperparameter settings are listed in Table 1, where A and C ($\beta_2 = 0.9$) are the settings used in [32] for CIFAR-10 and LSUN bedrooms [42] respectively, and B is the settings used in the paper [3]. For $\alpha$-GAN, because the authors suggested a learning rate of 0.0005, the learning rates for A and B are 0.0005, while the learning rates for C and D are 0.0001 and 0.001, respectively. For AGE, the learning rate in Table 1 is the initial value, and decays during training as in the original paper.

In Fig. 5(a) and (b), we show the best ISs and FIDs of each method on CIFAR-10 with the settings A-D. We can see that reducing the learning rate decreases the performance of all methods, while increasing the learning rate has no significant effect on the performance of all methods (Compared the better of A and B with D). AGE is very sensitive to the changes of hyperparameters and fails to generate plausible samples unless using the hyperparameter settings suggested by the authors. In Table 2, we list the best hyperparameter settings for each method. The learning rate is 0.0002 for GAN, GAN-RL(Ours), DRAGAN and AGE like in many GAN's paper such as [3,32,33,44]. For $\alpha$-GAN, the model training with $\beta_2$ of 0.999 performed better than training with $\beta_2$ of 0.9, which was different from the original settings [28]. In

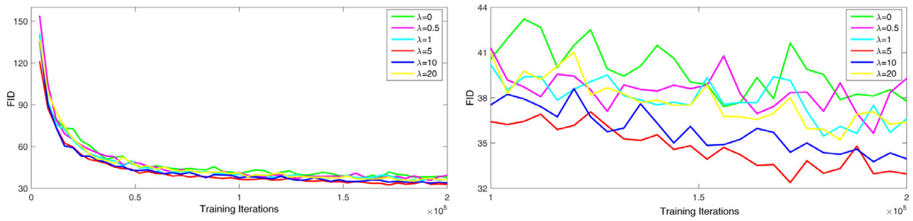


(a) ISs on CIFAR-10

(b) FIDs on CIFAR-10

(c) CIFAR-10 random samples

**Fig. 6.** The ISs (up left), the FIDs (up right) and random samples (down) for different methods with DCGAN architecture on CIFAR-10.



(a) The FID training curves of different $\lambda$ (b) Zoom-in of (a) for iterations $1\times10^5$ to $2\times10^5$

**Fig. 7.** Comparison of different λ on CIFAR-10. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

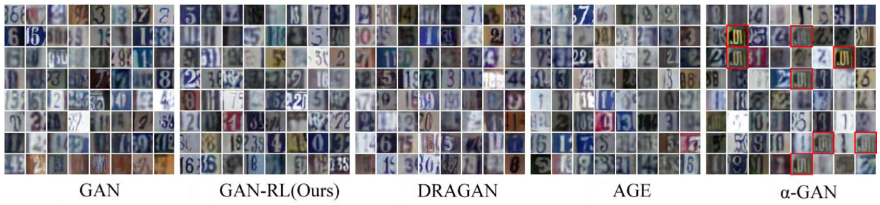


**Fig. 8.** SVHN random samples. Note that: some messy samples of $\alpha$-GAN were highlighted with red boxes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Table 3**
The best FIDs on CIFAR-10, SVHN and LSUN bedrooms for five models: GAN, GAN-RL(Ours), DRAGAN, AGE and $\alpha$-GAN.

| Datasets FIDs methods | GAN | GAN-RL (Ours) | DRAGAN | AGE | $\alpha$-GAN |
|---|---|---|---|---|---|
| CIFAR-10 | 37.4 | **32.4** | 35.1 | 40.7 | 35 |
| SVHN | 28.3 | **17.2** | 24.4 | 30.2 | 58.2 |
| LSUN | 34.6 | **25.4** | 28 | 39.5 | 45 |

this subsection, we only manually adjust the hyperparameters for different methods. Dong et al. [45] provided a novel method to automatically select hyperparameters for tracking. How to apply their method to GAN's hyperparameter selection is the future research direction.

In Fig. 6(a) and (b), we plot the ISs and the FIDs over the training iterations under the optimal hyperparameter settings, where we find that GAN-RL is more stable and performs better than other methods. In Fig. 6(c), we show the samples generated by different methods on CIFAR-10, but it is difficult to judge which one is better based on visual evaluation. The best FIDs obtained by different methods on CIFAR-10 are displayed in the second row of Table 3, in which GAN-RL gets a lowest FID, and AGE gets a highest FID. We also list the total training time for different comparison methods in Table 4, where $\alpha$-GAN is the slowest, and GAN-RL is the second fastest, only slower than GAN.

### 6.2. λ test and network architecture robustness

*Comparison different λ.* In this subsection, we implemented a set of experiments to test the effect of reconstruction loss and choose the optimal value for λ. Under the optimal settings selected in the previous subsection, we tried $\lambda\in\{0, 0.5, 1, 5, 10, 20\}$, the FID results of which were shown in Fig. 7. $\lambda=0$ (green curve) performs the worst, and its FID curve fluctuates greatly and is basically higher than other curves. $\lambda=0.5$ (magenta curve) improves the performance, and $\lambda=1$ (cyan curve) further improves the training stability. $\lambda=5$ (red curve) performs best in terms of training stability and image quality, and is almost continuously lower than other curves. All in all, from 0 to 5, increasing λ improves the training stability and the sample quality, but when λ exceeds 5, the performance of the model decreases, the larger the worse. Note that $\lambda=0$ represents the model is the original GAN.

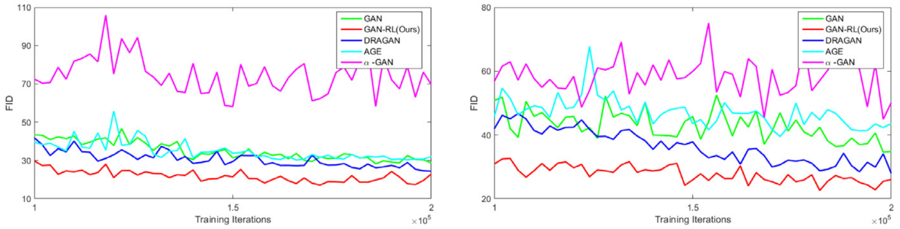
*Network architecture robustness.* To illustrate the robustness of the proposed method on network architecture, we implemented experiments with ResNet architecture on CIFAR-10. In Table 5 we list the FIDs and ISs for three methods with ResNet architecture[2]. The IS of GAN-RL is 7.83, which is slightly lower than WAGN-GP (7.86), but the FID of GAN-RL is 17.4, which is better than WAGN-GP (29.3) and SNGAN (21.7) [46]. This suggests that our approach at least can be competitive with other state-of-the-art methods with ResNet architecture. Please note that, for fair comparison, in this experiment, the ISs and the FIDs were calculated over 50k generated images like in SNGAN and WGAN-GP. SNGAN is a new technique for weight normalization, and is orthogonal to our work. The best reported ImageNet-based IS on CIFAR-10 for unsupervised models is 8.80 by PGGAN [4], which also produced the images at resolution of $1024^2$, however this is trained on different architecture and may not be directly compared.

### 6.3. Performance on SVHN and LSUN bedrooms

*Results on SVHN.* The best generated samples and the graphic of FIDs over the iterations for different methods on SVHN [41] are shown in Fig. 8 and Fig. 9(a) respectively. Most of the samples generated by GAN-RL, GAN, DRAGAN and AGE are plausible, while some samples produced by $\alpha$-GAN are messy (marked with red boxes). It can be seen from Fig. 9(a) that the proposed method GAN-RL outperforms other methods in training stability, image quality and diversity. The best FIDs for the five methods are reported in the third row of Table 3, where the FID of GAN-RL is 17.2, which is much lower than other methods. The best FIDs of $\alpha$-GAN is 58.2, which is almost twice that of other methods. The total training time for each method on SVHN is reported on the third row of Table 4, where GAN-RL takes 8430 s, slightly faster than DRAGAN (8853 s), much faster than AGE (12850 s) and $\alpha$-GAN (13580 s).

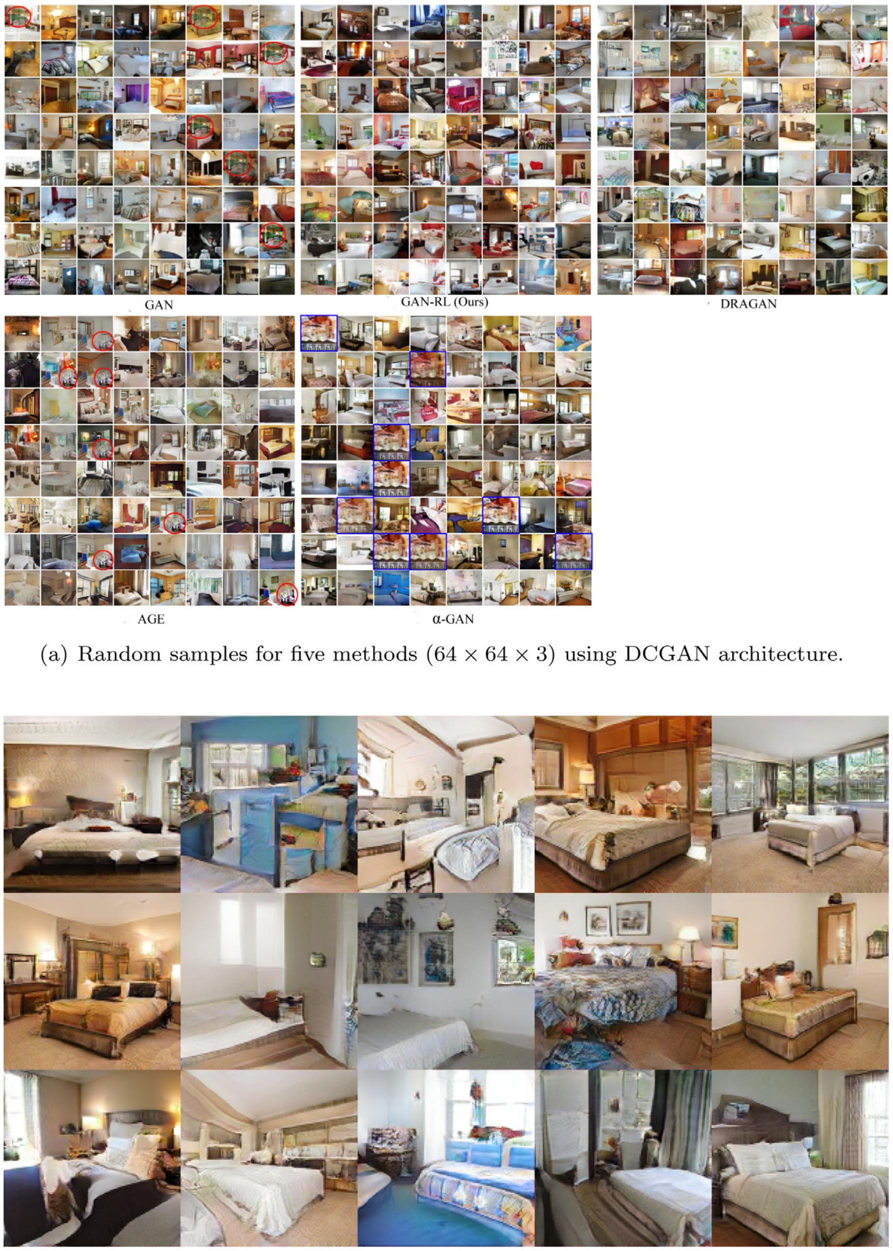
*Results on LSUN bedrooms.* LSUN bedrooms dataset consists of more than 3000k images at resolution 256 × 256, and we resized the images to 64 × 64 and 128 × 128. The FIDs over the iterations are shown in Fig. 9(b), in which the proposed method GAN-RL significantly outperforms than the other methods and its FIDs

[2] The ResNet module of our method and SNGAN [46] are similar to the ones used in WGAN-GP [32], but SNGAN adopts more filters in generator, and our method adopts LeakyRelu activation function instead of Relu for the discriminator.



(a) FIDs over iterations on SVHN. (b) FIDs over iterations on LSUN bedrooms.

**Fig. 9.** FIDs over iterations on SVHN (left) and LSUN bedrooms (right) for different methods. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Random samples for five methods ($64 \times 64 \times 3$) using DCGAN architecture.

(b) Random samples of GAN-RL (Ours, $128 \times 128 \times 3$) using ResNet architecture.

**Fig. 10.** LSUN bedrooms random samples with resolution of $64 \times 64 \times 3$ (up) and $128 \times 128 \times 3$ (down). Note that: partially similar samples of GAN and AGE were highlighted with red circles, and messy samples of $\alpha$-GAN were highlighted with blue boxes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Table 4**
The total training time on CIFAR-10, SVHN and LSUN bedrooms for five models: GAN, GAN-RL(Ours), DRAGAN, AGE and $\alpha$-GAN. Please note that all programs were performed on ubuntu 16.04.2 systems with a single Nvidia GeForce GTX 1080 GPU.

| Datasets time(s) methods | GAN | GAN-RL (Ours) | DRAGAN | AGE | $\alpha$-GAN |
|---|---|---|---|---|---|
| CIFAR-10 | **4158** | **6050** | 7961 | 10177 | 10281 |
| SVHN | **5149** | **8430** | 8853 | 12850 | 13580 |
| LSUN | **23829** | **41319** | 47845 | 52423 | 62727 |

**Table 5**
The FIDs and ISs for different methods with Resnet architecture on CIFAR-10.

| Methods | GAN-RL(Ours) | WGAN-GP [32] | SN-GAN [46] |
|---|---|---|---|
| FIDs | 17.4 ± .18 | 29.3(In [40]) | 21.7 ± .21 |
| ISs | 7.83 ± .09 | 7.86 ± .08 | 8.22 ± .05 |

**Table 6**
The best FIDs and the iterations on SVHN, CelebA and LSUN bedrooms. GAN, DRAGAN, and GAN-GP are the three baselines, GAN-RL, DRAGAN-RL, and GAN-GP-RL are augmented with reconstruction loss to the objective of these three baselines.

| Dataset | Method | Iterations | FID | Method | Iterations | FID |
|---|---|---|---|---|---|---|
| SVHN | GAN-RL | 172k | 17.2 | GAN | 180k | 28.3 |
| | DRAGAN-RL | 194k | 18.9 | DRAGAN | 171k | 24.4 |
| | GAN-GP-RL | 194k | **15.0** | GAN-GP | 198k | 16.1 |
| CelebA | GAN-RL | 178k | 20.5 | GAN | 190k | 24.4 |
| | DRAGAN-RL | 188k | **10.1** | DRAGAN | 176k | 11.8 |
| | GAN-GP-RL | 194k | 10.3 | GAN-GP | 200k | 11.9 |
| LSUN | GAN-RL | 145k | 25.4 | GAN | 195k | 34.6 |
| | DRAGAN-RL | 191k | **24.5** | DRAGAN | 189k | 28.0 |
| | GAN-GP-RL | 191k | 26.1 | GAN-GP | 195k | 32.4 |

(red line) are consistently lower than other methods. The generated images for different methods are show in Fig. 10(a). We observe that the samples generated by GAN-RL(Ours) and DRAGAN are of higher quality and more diversification, while some samples generated by GAN and AGE have very similar scenes at the same location (highlighted by red circles). $\alpha$-GAN suffers from mode dropping, and produces some meaningless and similar samples (highlighted with blue boxes). From the generated samples Fig. 10(a) and the FIDs Fig. 9(b), we confirm that FID is correlated well with human evaluation. The bad samples have the highest FIDs ($\alpha$-GAN), some partial similar samples have the medium FIDs (GAN, AGE), and the good samples have the lowest FIDs (GAN-RL, DRAGAN). The best FIDs are reported in the fourth row of Table 3, and GAN-RL obtains the lowest. We also generated images with resolution of $128 \times 128$ using ResNet network architecture, and some random generated samples are displayed in Fig. 10(b). The generated high resolution images are clear and plausible.

### 6.4. Improve the performance of various methods

In this section, we performed three pairs comparison tests across three dataset under the optimal hyperparameter settings of each method. The difference between each pair is whether the objective function includes the reconstruction loss. The best FID of each method is listed in Table 6, where we observe that reconstruction loss improves the performance of all methods on every dataset. From the lowest FID (highlighted with bold type) of each dataset, we also get a conclusion that combining adversarial loss, reconstruction loss and gradient penalty can produce higher quality images.

On LSUN bedrooms dataset, the reconstruction loss has considerable influence on the three baseline, indicating that as the train data increase, the reconstruction loss catch more data characteristics (LSUN bedrooms includes 3033k images, much larger than SVHN and CelebA dataset, which include more than 60k and 200k image respectively). On CelebA, the reconstruction loss only slightly improve the performance of the three methods, possibly because the background of face image is clear and simple (For CelebA images, we crop the images according to the method in [4], thus the whole image only includes the face). Note that in this section, we added the gradient penalty proposed in [32] to the standard GAN (denote as GAN-GP) not WGAN as in original paper, because GAN-GP is much faster and performs similarly to the WGAN-GP model [47].

## 7. Conclusions

In this paper, we develop GAN-RL, which uses the reconstruction loss to improve the performance of adversarial generative networks on training stability and mode diversity. The generator utilizes the features learned by the discriminator to reconstruct real data, which encourages the discriminator to capture informative features and directs the generator towards to near the manifold of real data.

As it is difficult to evaluate the sample quality only according to the visual judgment, we use quantitative metrics Inception Score and Fréchet Inception Distance to assess the quality and the diversity of the generation. By comparing our method with GAN's varieties and hybrids methods across CIFAR-10, SVHN and LSUN bedrooms datasets, we illustrate that our method outperforms over other methods. We also combine our method with gradient penalty based methods, and the experimental results show that our method can improve the performance of various GANs. As our proposed method is easily implemented, although we only combine our method with gradient penalty methods, we believe it can be generalized to various GAN based methods.

## Acknowledgments

This research is funded by the National Natural Science Foundation of China under Grant 61573145, the Public Research and Capacity Building of Guangdong Province of China under Grant 2014B010104001 and the Basic and Applied Basic Research of Guangdong Province of China under Grant 2015A030308018, the authors are greatly thanks to these grants.

## References

[1] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Proceedings of the Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.
[2] E. Denton, S. Chintala, A. Szlam, R. Fergus, Deep generative image models using a laplacian pyramid of adversarial networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2015, pp. 1486–1494.
[3] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, Proceedings of the International Conference on Learning Representations (2016).
[4] T. Karras, T. Aila, S. Laine, J. Lehtinen, Progressive growing of gans for improved quality, stability, and variation, arXiv:1710.10196 (2017).
[5] Y. Li, N. Xiao, W. Ouyang, Improved boundary equilibrium generative adversarial networks, IEEE Access 6 (2018) 11342–11348.
[6] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the

IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 105–114.

[7] P. Isola, J.Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5967–5976.

[8] J.Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2242–2251.

[9] M. Mathieu, C. Couprie, Y. Lecun, Deep multi-scale video prediction beyond mean square error, arXiv:1511.05440 (2015).

[10] S. Ren, K. He, R. Girshick, J. Sun, Faster r-CNN: Towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2015) 1137–1149.

[11] W. Wang, J. Shen, L. Shao, Video salient object detection via fully convolutional networks, IEEE Trans. Image Process. 27 (1) (2017) 38–49.

[12] X. Dong, J. Shen, Triplet loss in siamese network for object tracking, in: Proceedings of the European Conference on Computer Vision, 2018.

[13] W. Wang, J. Shen, H. Ling, A deep network solution for attention and aesthetics aware photo cropping, IEEE Trans. Pattern Anal. Mach. Intell. (2018).

[14] W. Wang, J. Shen, Deep cropping via attention box prediction and aesthetics assessment, Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2205–2213.

[15] W. Wang, J. Shen, Deep visual attention prediction, IEEE Trans. Image Process. 27 (5) (2018) 2368–2378.

[16] W. Wang, Y. Xu, J. Shen, S. Zhu, Attentive fashion grammar network for fashion landmark detection and clothing category classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[17] Z. Liu, S. Yan, P. Luo, X. Wang, X. Tang, Fashion landmark detection in the wild, Proceedings of the European Conference on Computer Vision, 2016, pp. 229–245.

[18] A.V.D. Oord, N. Kalchbrenner, K. Kavukcuoglu, Pixel recurrent neural networks, Proceedings of the International Conference on Machine Learning, vol. 48, 2016, pp. 1747–1756.

[19] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, Proceedings of the International Conference on Learning Representations, 2014.

[20] A. Odena, Semi-supervised learning with generative adversarial networks, arXiv:1606.01583 (2016).

[21] M. Mirza, S. Osindero, Conditional generative adversarial nets, Comput. Sci. (2014) 2672–2680.

[22] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, P. Abbeel, InfoGAN: interpretable representation learning by information maximizing generative adversarial nets, Proceedings of the Advances in Neural Information Processing Systems, 2016, pp. 2172–2180.

[23] J. Zhao, M. Mathieu, Y. Lecun, Energy-based generative adversarial network, arXiv:1609.03126 (2016).

[24] D. Berthelot, T. Schumm, L. Metz, Began: Boundary equilibrium generative adversarial networks, arXiv:1703.10717 (2017).

[25] D. Wardefarley, Y. Bengio, Improving generative adversarial networks with denoising feature matching, Proceedings of the International Conference on Learning Representations, 2017.

[26] A.B.L. Larsen, S.K. Snderby, H. Larochelle, O. Winther, Autoencoding beyond pixels using a learned similarity metric, Proceedings of the International Conference on Machine Learning, 2016, pp. 1558–1566.

[27] T. Che, Y. Li, A.P. Jacob, Y. Bengio, W. Li, Mode regularized generative adversarial networks, Proceedings of the International Conference on Learning Representations, 2017.

[28] M. Rosca, B. Lakshminarayanan, D. Warde-Farley, S. Mohamed, Variational approaches for auto-encoding generative adversarial networks, arXiv:1706.04987 (2017).

[29] X. Mao, Q. Li, H. Xie, R.Y.K. Lau, Z. Wang, S.P. Smolley, Least squares generative adversarial networks, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2813–2821.

[30] Y. Mroueh, T. Sercu, V. Goel, Mcgan: mean and covariance feature matching GAN, Proceedings of the International Conference on Machine Learning, 2017, pp. 2527–2535.

[31] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein GAN, arXiv:1701.07875 (2017).

[32] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, Improved training of Wasserstein GANs, Proceedings of the Advances in Neural Information Processing Systems, 2017, pp. 5769–5779.

[33] D. Ulyanov, A. Vedaldi, V. Lempitsky, Adversarial generator-encoder networks, arXiv:1704.02304 (2017).

[34] N. Kodali, J. Abernethy, J. Hays, Z. Kira, On convergence and stability of GANs, arXiv:1705.07215 (2017).

[35] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, J. Yosinski, Plug and play generative networks: Conditional iterative generation of images in latent space, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3510–3520.

[36] L. Theis, A.V.D. Oord, M. Bethge, A note on the evaluation of generative models, arXiv:1511.01844 (2015).

[37] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training gans, Proceedings of the Advances in Neural Information Processing Systems, 2016, pp. 2234–2242.

[38] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018) 2818–2826.

[39] A. Krizhevsky, G. Hinton, Learning Multiple Layers of Features from Tiny Images, Technical Report, Citeseer, 2009.

[40] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, Gans trained by a two time-scale update rule converge to a local nash equilibrium, Proceedings of the Advances in Neural Information Processing Systems, 2017, pp. 6626–6637.

[41] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A.Y. Ng, Reading digits in natural images with unsupervised feature learning, Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011, p. 5.

[42] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, J. Xiao, Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop, arXiv:1506.03365 (2015).

[43] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, Proceedings of the International Conference on Learning Representations (2015).

[44] M. Lucic, K. Kurach, M. Michalski, S. Gelly, O. Bousquet, Are GANs created equal? A large-scale study, arXiv:1711.10337 (2017).

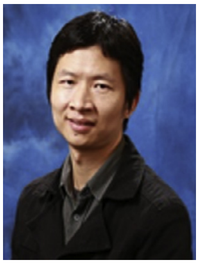
[45] X. Dong, J. Shen, W. Wang, Y. Liu, L. Shao, F. Porikli, Hyperparameter optimization for tracking with continuous deep q-learning, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[46] T. Miyato, T. Kataoka, M. Koyama, Y. Yoshida, Spectral normalization for generative adversarial networks, arXiv:1802.05957 (2018).

[47] W. Fedus, M. Rosca, B. Lakshminarayanan, A.M. Dai, S. Mohamed, I. Goodfellow, Many paths to equilibrium: GANs do not need to decrease a divergence at every step, arXiv:1701.08446 (2017).

**Yanchun Li** obtained the B.E.E., M.S.E.E. degrees in computer science all from the College of Information Engineering of Xiangtan University, Hunan, China. She is now a Ph.D. student in the School of Computer Science and Engineering, South China University of Technology. Her research interests include deep learning, computer vision and image processing.

**Nanfeng Xiao** obtained the B.E.E. degree from the Department of automatic control and computer of Huazhong University of Science and Technology, received the M.S.E.E. and Ph.D. degrees in engineering from Northeastern University, China and Yokohama National University, Japan respectively. He is now a professor and Ph.D. supervisor of South China University of Technology. His research interests include deep learning and artificial intelligence.



**Wanli Ouyang** obtained Ph.D. degrees from the Department of Electronic Engineering, the Chinese University of Hong Kong. He is now a senior lecturer in the School of Electrical and Information Engineering, University of Sydney. His research interests include deep learning and its application to computer vision and pattern recognition, image and video processing.