Alleviating Mode Collapse in GAN via Diversity Penalty Module

Sen Pei, Richard Yi Da Xu, Shiming Xiang, Gaofeng Meng

1Institute of Automation, Chinese Academy of Sciences
2University of Technology Sydney

Abstract

The vanilla GAN (Goodfellow et al. 2014) suffers from mode collapse deeply, which usually manifests as that the images generated by generators tend to have a high similarity amongst them, even though their corresponding latent vectors have been very different. In this paper, we introduce a pluggable diversity penalty module (DPM) to alleviate mode collapse of GANs. It reduces the similarity of image pairs in feature space, i.e., if two latent vectors are different, then we enforce the generator to generate two images with different features. The normalized Gram matrix is used to measure the similarity. We compare the proposed method with Unrolled GAN (Metz et al. 2016), BourGAN (Xiao, Zhong, and Zheng 2018), PacGAN (Lin et al. 2018), VEEGAN (Srivastava et al. 2017) and ALI (Dumoulin et al. 2016) on 2D synthetic dataset, and results show that the diversity penalty module can help GAN capture much more modes of the data distribution. Further, in classification tasks, we apply this method as image data augmentation on MNIST, Fashion-MNIST and CIFAR-10, and the classification testing accuracy is improved by 0.24%, 1.34% and 0.52% compared with WGAN-GP (Gulrajani et al. 2017), respectively. In domain translation, diversity penalty module can help StarGAN (Choi et al. 2018) generate more accurate attention masks and accelerate the convergence process. Finally, we quantitatively evaluate the proposed method with IS and FID on CelebA, CIFAR-10, MNIST and Fashion-MNIST, and the results suggest GAN with diversity penalty module gets much higher IS and lower FID compared with some SOTA GAN architectures.

Introduction

A deeper convolutional neural network can bring better performance while also increasing the demand for data, but sometimes, it is burdensome to fulfill this requirement. In this paper, we mainly focus on how to generate data in high diversity based on the available, and further, apply the augmented data into some downstream tasks for better performance.

Except for the geometric transformations of images, GAN (Goodfellow et al. 2014) can be used to augment data. Given the fact that images distribute on a low dimensional manifold of higher space, we can use GAN to learn the mapping between latent vectors and images in data space. However, to the best of our knowledge, directly using GAN to augment image data dose not work well. In (Tanaka and Aranha 2019), researchers find that the GAN-based augmentation method can bring negative effects on classification task, even sometimes, it is not surprising to find that the traditional data augmentation methods can outperform the GAN-based method significantly. As in (Shorten and Khoshgoftaar 2019), low quality and lack of feature modes in fake images are blamed for these poor performance.

In general, mode collapse usually manifests as that the trained generator can only generate images in some specific classes which really harms the data diversity. Using multiple GANs can alleviate this problem to some extent. However, in our experiments (see Figure 7), we notice that even very different latent vectors may be mapped to similar images, this kind of reduction in data diversity is left to be settled.

To alleviate the effects of mode collapse and improve data diversity, we present a novel pluggable module called Diversity Penalty in this paper, hereinafter, DP. Figure 1 shows the domain translation with proposed diversity penalty module. From top to bottom are: outputs of StarGAN, attention masks of StarGAN, attention masks of StarGAN_DPM (λ=1e-3) and (λ=1e-4). We can see DPM helps GAN capture clearer features and thus accelerate the convergence process. λ is the coefficient of DPM.
the discriminator, \( z \) framework above, \( f_1 \) and \( f_2 \) are feature maps extracted from the discriminator, \( z_1 \) and \( z_2 \) are latent vectors while \( g_1 \) and \( g_2 \) are their corresponding fake images. \( S(\cdot) \) indicates the similarity measurement function. The key idea of diversity penalty is that the similarity relationship of fake images’ features should be consistent with their corresponding latent vectors.

shows the pipeline of our framework. Concretely, the more difference between latent vectors the more different their corresponding fake images should be. In latent space, we measure the similarity between latent vectors using cosine coefficient. However, in data space, each image usually has a great amount of pixels which are unnecessary for distinguishing, and in fact, \((\text{Zhou et al. 2016})\) find that the feature representations can better describe an image than pixels. Taking this into consideration, we use discriminator to extract features of images first, and then, similarly, the cosine coefficient is used to measure the similarity between feature maps. Besides, we perform nonlinear mapping to normalize the cosine coefficient to the range of 0~1. Our proposed method can be stated as the similarity of feature pairs should be consistent with that of their corresponding latent vector pairs. This paper mainly has the following contributions:

- We alleviate mode collapse from a new perspective and propose a framework to visualize this phenomenon. Based on this framework (Figure 18), we also give the convincing statistic results of mode collapse alleviation;
- Compared with other complex methods (Dumoulin et al. 2016; Lin et al. 2018; Metz et al. 2016; Srivastava et al. 2017; Xiao, Zhong, and Zheng 2018), our proposed diversity penalty module is effective yet easy to perform, and it has good generalization since it can be used as an attachment on almost all GANs. The results in Figure 8 on 2D synthetic dataset show that the diversity penalty module can help GAN capture much more modes effectively, and Figure 11 also suggests DPM has good performance in domain translation;
- In downstream tasks such as image data augmentation and image generation, our proposed method gains a considerable improvement, also, it performs well on IS (Barrett and Sharma 2018) and FID (Heusel et al. 2017). Besides, compared with MSGAN (Mao et al. 2019), our method can generate better interpolation results with less noisy points on CelebA (Liu et al. 2015).

### Related Work

**Generative Adversarial Nets**

GAN is the representative of generative models. In (Goodfellow et al. 2014), Ian J. Goodfellow et al. proposed the vanilla GAN for generating high-quality images, at that time, it is not an easy task to train GAN stably, and the imbalance between generator and discriminator can easily result in divergence. Further, in (Arjovsky, Chintala, and Bottou 2017), Martin Arjovsky et al. use Wasserstein distance to measure the similarity of two distributions instead of using KL-divergence which greatly reduces the difficulty in training GAN, and then, in (Gulrajani et al. 2017), Ishaan Gulrajani et al. propose the gradient penalty term to enforce the Lipschitz constraint instead of using weight clipping as in (Arjovsky, Chintala, and Bottou 2017), and this has made the training process more stable.

**Data Augmentation**

Image data augmentation has been proven to be effective in practice. In (Krizhevsky, Sutskever, and Hinton 2012), Alex Krizhevsky and Geoffrey E. Hinton et al. use data augmentation to reduce overfitting, and their AlexNet has made great success in ImageNet LSVRC2010(1). Also in (Shorten and Khoshgoftaar 2019), studies suggest that even some very simple techniques such as cropping, rotating and flipping can have considerable effects on reducing overfitting and improving testing accuracy. Similarly, in (Perez and Wang 2017), Jason Wang and Luis Perez systematically discuss several different branches (e.g. traditional transformations, generative adversarial nets, learning the augmentation, etc.) of image data augmentation and point out their effectiveness.

**Feature Representation of CNN**

A deep convolutional layer can extract the feature of an input image accurately. In (Zeiler and Fergus 2014), Matthew D. Zeiler et al. use deconvnet to visualize the features that a fully trained model has learned. Further, in (Zhou et al. 2016), Bolei Zhou et al. demonstrate that the convolutional neural networks are able to localize the discriminative regions of image. Based on this finding, we use the features extracted from discriminator to represent the images instead of using images directly. And in (Selvaraju et al. 2017), the Grad-CAM method proposed by R.R. Selvaraju et al. supports the results in Zhou et al. (2016). Figure 3 shows some Grad-CAM results on CelebA (Liu et al. 2015) with our trained discriminator.

**Reducing Mode Collapse**

For improving data diversity and stable training, researchers have done a lot of work. In Unrolled GAN (Metz et al. 2016), Luke Metz et al. define the generator objective with respect to an unrolled optimization of the discriminator, which makes the training more stable and leads to a better solution. In VEEGAN (Srivastava et al. 2017), Akash Srivastava et al. introduce a variational principle for estimating implicit probability distributions which can help avoid mode collapse. Further, in PC-GAN (Lin et al. 2018), Zinan Lin et al. let the discriminator to make decisions based on multiple samples from the same

---

1See more details at: [http://www.image-net.org/](http://www.image-net.org/)
class which can penalize generator with mode collapse. In BourGAN (Xiao, Zhong, and Zheng 2018), Chang Xiao et al. view modes as a geometric structure of data distribution in a metric space which also leads to a better generator. Recently, in MSGAN (Mao et al. 2019), Qi Mao et al. modify the objective function for encouraging the generators to explore more minor modes. These works have provided inspiration for the follow-up study.

Diversity Penalty

Expanding the feature space of generator will improve the diversity of fake images. To this end, we propose diversity penalty term which can be used as a constraint to make the feature distribution more discrete. For consistency, we use the same notations of GAN as Goodfellow et al. [2014].

Measurement of similarity

Suppose \( p_z(z) \) is the distribution of latent vectors which follows a standard normal distribution, we randomly sample a batch vectors \( \{ z_1, z_2, ..., z_m \} \) from \( p_z(z) \), and then, the normalized Gram matrix can be shown as:

\[
G^*_z(i, j) = \frac{z^T_i z_j}{||z_i||_2 \cdot ||z_j||_2} \tag{1}
\]

where \( || \cdot ||_2 \) represents l2-norm. It is reasonable to suppose that \( z_i \) and \( z_j \) are independent identically distributed (i.i.d.), the expectation of \( G^*_z(i, j) \) is 0 which can be derived from the following claim:

\( f(x) \) and \( g(x) \) are Gaussian PDFs with means \( \mu_f \) and \( \mu_g \) and standard deviations \( \sigma_f \) and \( \sigma_g \), then the product of \( p(x) \) and \( q(x) \) follows a scaled Gaussian distribution with

\[
\mu = \frac{\mu_f \sigma_g^2 + \mu_g \sigma_f^2}{\sigma_f^2 + \sigma_g^2} \quad \text{and} \quad \sigma = \sqrt{\frac{\sigma_f^2 \sigma_g^2}{\sigma_f^2 + \sigma_g^2}}. \tag{2}
\]

Nonlinear mapping

Using the similar approach stated above, the similarity of feature pairs can be got as follows:

\[
G^*_f(i, j) = \frac{f^T_i f_j}{||f_i||_2 \cdot ||f_j||_2} \tag{3}
\]

where \( f_i \) represents flattened feature map of the \( i \)-th fake image extracted from discriminator. Since the value of \( \frac{||z_i||_2 \cdot ||z_j||_2}{||f_i||_2 \cdot ||f_j||_2} \) can be zero or negative, performing division directly doesn’t make sense, and thus, we use \( \text{sigmoid} \) function to scale them. The scale factor is denoted by \( s \) and formula (1)(2) can be revised as:

\[
G^*_z(i, j) = \sigma(s \frac{z^T_i z_j}{||z_i||_2 \cdot ||z_j||_2}) \tag{4}
\]

\[
G^*_f(i, j) = \sigma(s \frac{f^T_i f_j}{||f_i||_2 \cdot ||f_j||_2}) \tag{5}
\]

In this paper, we set \( s \) to 5, because in this case, the range of \( \text{sigmoid} \) function can be well covered while avoiding gradient vanishment.

Loss function

For alleviating mode collapse, the diversity penalty module should pay much attention to the second situation which often results in mode collapse. Through these observations, the diversity penalty module is designed as follows:

\[
DP(z) = \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} G^*_f(i, j) \tag{6}
\]

where \( m \) represents the batch size. It is clear that the advantage of our proposed diversity penalty is its generalization which means it almost can be used in all GAN frameworks, what we have to do is just to add this diversity penalty term to the loss of generator.

Take the vanilla GAN for example, we use \( G(\cdot) \) to indicate generator while \( D(\cdot) \) to indicate discriminator. \( z \) is latent vector. As Goodfellow et al. [2014], we train \( D(\cdot) \) to maximize the probability of assigning the correct label to both training examples and samples from \( G(\cdot) \) (fake images), also, we simultaneously train \( G(\cdot) \) to get high score from \( D(\cdot) \). Thus, the basic loss function of GAN can be formulated as follows:

\[
\max_G \mathbb{E}_{z \sim p_z} L_G(z) = \mathbb{E}_{z \sim p_z} D(G(z)) \tag{7}
\]

\[
\min_D \mathbb{E}_{z \sim p_z} L_D(z, x) = \mathbb{E}_{z \sim p_z} D(G(z)) - \mathbb{E}_{x \sim p_x} D(x) \tag{8}
\]

where \( p_z \) represents the distribution of latent vector, and \( p_r \) represents the distribution of training data. To perform diversity penalty, we just need to add diversity penalty loss to generator. The loss function of GAN with DPM can be formulated as follows:

\[
\min_D \mathbb{E}_{z \sim p_z} L_D(z, x) = \mathbb{E}_{z \sim p_z} D(G(z)) - \mathbb{E}_{x \sim p_x} D(x) + \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} G^*_f(i, j) \tag{9}
\]
\[
\max_G L_G(z) = \mathbb{E}_{z \sim p_z} D(G(z)) - \lambda \mathbb{E}_{z \sim p_z} DP(z)
\]
\[
\min_D L_D(z, x) = \mathbb{E}_{z \sim p_z} D(G(z)) - \mathbb{E}_{x \sim p_X} D(x)
\]
\[
\max_D \min_G \mathbb{E}_{z \sim p_z} D(G(z)) - \mathbb{E}_{x \sim p_X} D(x) - \lambda \mathbb{E}_{z \sim p_z} DP(z)
\]
where \(\lambda\) is the scale factor of diversity penalty term. The loss function of discriminator remains unchanged. The objective function of GAN with DPM is shown in (11). According to formulas (9) and (10), the whole training process can be summarized in Algorithm 1.

Algorithm 1: GAN with DPM training via mini-batch SGD
1: for total training epochs do
2: for \(k\) times do
3: Sample a batch data from \(p_z: \{z_1, z_2, \ldots, z_m\}\); 
4: Sample a batch data from \(p_r: \{x_1, x_2, \ldots, x_m\}\); 
5: Update discriminator:
6: \(\theta_d \leftarrow \theta_d - \nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} L_D(z_i, x_i)\)
7: end for
8: Sample a batch data from \(p_z: \{z_1, z_2, \ldots, z_m\}\); 
9: Update generator:
10: \(\theta_g \leftarrow \theta_g - \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} L_G(z_i)\)
11: end for

Ablation Study and Experiments
Prior works in (Selvaraju et al. 2017) Zeiler and Fergus 2014 Zhou et al. 2016 have proven that the CNN can extract feature representations accurately. Further, in this section, we first tell the feasibility of our similarity measurement, and then, we check the convergence performance of our proposed DPM. Extended experiments in domain translation are performed, and we find that StarGAN (Choi et al. 2018) with DPM can generate better attention masks compared with its vanilla counterpart while accelerating the convergence process. Also, we compare our method with BourGAN (Xiao, Zhong, and Zheng 2018), Unrolled GAN (Metz et al. 2016), VEEGAN (Srivastava et al. 2017), PacGAN (Lin et al. 2018) and ALI (Dumoulin et al. 2016) on synthetic datasets(2D Ring and 2D Grid), the results show that the diversity penalty module can help GAN capture more modes effectively. Besides, we evaluate the proposed DPM under IS score and FID score. In downstream tasks such as image classification, GAN with DPM can also introduce a markable improvement. Finally, on CelebA (Liu et al. 2015), we compare the performance of DCGAN (Radford, Metz, and Chintala 2015), DCGAN_MS, DCGAN_DP, WGAN_GP, WGAN_GP_MS and WGAN_GP_DP, respectively. Here, MS represents mode seeking proposed in (Mao et al. 2019), and DP represents our proposed diversity penalty. Both visual and quantitative results suggest that our proposed method outperforms the others.

Feasibility Analysis of Similarity Measurement
Similarity measurement must has two basic characteristics:
- The similarity value should be higher within classes than between classes.
- Visually similar images should be close in feature space.

Picking Fashion-MNIST as samples, we extract the features using the trained discriminators and compute their similarity value among different categories. Results shown in Figure 4 tell that our adopted measurement is feasible.

Further, we perform similar operation within one specific class on Fashion-MNIST to verify the second character stated above. Results in Figure 5 confirm that visually similar images are also similar in feature space and vice versa.

Convergence of DPM and Domain Translation
For generative adversarial net, whether it can converge stably or not is vital, and thus, we evaluate our proposed diversity penalty module on MNIST, Fashion-MNIST, CIFAR-10 and CelebA, respectively. Architectures of GAN are contained in Appendix Table. Here, we just talk the results on CelebA and domain translation, the detailed results of other datasets are attached in Appendix Figure.
We set StarGAN \cite{choi2018stargan} as our baseline, two groups with DPM are set for comparison. From the results shown in Figure \ref{fig:loss}, we can see that DPM can accelerate the convergence of generator significantly.

![Loss of generator](image)

**Figure 6: Loss of generator.** Coefficient of diversity penalty module $\lambda$ is set to 1e-3 in StartGAN\_DPM 1 while 1e-4 in StartGAN\_DPM 2.

This acceleration is achieved because DPM can capture accurate feature representations which are vital in facial expression transfer. Besides, from Figure \ref{fig:loss} it is clear to find that DPM can help StarGAN \cite{choi2018stargan} generate much clearer facial attention mask (with less background) which can bring better and smoother detail changes.

**DPM Effects Visualization**

In vanilla GANs, the latent vectors even with very low similarity may be mapped to similar images, but with DPM, this phenomenon is alleviated since this situation will result in higher loss. On the other hand, DPM makes similar fake images have corresponding latent vectors with higher similarity. Using the method in Figure \ref{fig:visualization} we test the trained generator with and without diversity penalty module on MNIST \cite{lecun1998mnist} and Fashion-MNIST \cite{xiao2017fashion}. Part of these results are shown in Figure \ref{fig:visualization} and they are in line with the analysis above.

Further, to avoid occasionality, we use generators with and without diversity penalty module to generate sufficient samples, 5k each class, and then use the method shown in formula \ref{formula} to calculate the similarity between latent vector pairs. In Table \ref{table:diversity} the value indicates the similarity of latent vector pairs whose corresponding fake images are similar under MSE metrics. We can find DPM prevents the case that two different latent vectors are mapped to similar fake images.

**Quantitative evaluation with IS** \cite{barratt2018inception, szegedy2016rethinking} and FID \cite{heusel2017gans} are performed on MNIST \cite{lecun1998mnist}, Fashion-MNIST \cite{xiao2017fashion}, CelebA \cite{liu2015faceattributes} and CIFAR-10 \cite{krizhevsky2009learning}. For each class, we use the trained generator to generate 5k images. The parameter $n_{split}$ in \cite{barratt2018inception} is 10. Table \ref{table:quantitative} shows the details.

**Table 1: Statistic results of diversity penalty module.**

| Dataset   | WGAN\_GP | Ours\_1 $\lambda = 5$ | Ours\_2 $\lambda = 10$ |
|-----------|-----------|------------------------|------------------------|
| M         | ↑ IS      | ↓ FID                  | ↑ IS                   | ↓ FID                  |
|           | 4.28 ± 0.68 | 7.35 ± 0.27 | 2.18 ± 0.003 | 2.91 ± 0.005 |
| FM        | ↑ IS      | ↓ FID                  | ↑ IS                   | ↓ FID                  |
|           | 7.36 ± 0.12 | 16.97 ± 0.13 | 6.43 ± 0.009 | 25.45 ± 0.017 |
| C10       | ↑ IS      | ↓ FID                  | ↑ IS                   | ↓ FID                  |
|           | 7.35 ± 0.07 | 28.45 ± 0.015 | 7.83 ± 0.007 | 28.45 ± 0.015 |
| CelebA    | ↑ IS      | ↓ FID                  | ↑ IS                   | ↓ FID                  |
|           | 2.78 ± 0.02 | 25.45 ± 0.015 | 2.91 ± 0.005 | 24.86 ± 0.002 |

↑: Greater is better. ↓: Lower is better.

**Table 2: Some quantitative results.**

| Dataset   | WGAN\_GP | Ours\_1 | Ours\_2 |
|-----------|-----------|---------|---------|
| M         | ↑ IS      | ↓ FID   | ↑ IS    | ↓ FID   |
|           | 2.18 ± 0.003 | 2.91 ± 0.005 | 2.31 ± 0.005 |
| FM        | ↑ IS      | ↓ FID   | ↑ IS    | ↓ FID   |
|           | 4.28 ± 0.004 | 4.38 ± 0.015 | 4.36 ± 0.005 |
| C10       | ↑ IS      | ↓ FID   | ↑ IS    | ↓ FID   |
|           | 7.35 ± 0.007 | 7.52 ± 0.005 | 7.83 ± 0.007 |
| CelebA    | ↑ IS      | ↓ FID   | ↑ IS    | ↓ FID   |
|           | 2.78 ± 0.002 | 2.91 ± 0.005 | 2.94 ± 0.002 |

↑: Greater is better. ↓: Lower is better.

**Data Augmentation via DPM**

We use DPM to augment data on MNIST \cite{lecun1998mnist}, Fashion-MNIST \cite{xiao2017fashion} and CIFAR-10 \cite{krizhevsky2009learning}, respectively. The fake images are served as auxiliary training set. On MNIST and Fashion-MNIST, we generate fake images in each epoch online. On CIFAR-10, we generate 5k images to serve as auxiliary training set. ResNet20v1 proposed in \cite{he2016deep} is adopted as classification net.

The group with traditional transformations on the original training set is named DA(Data Augmentation). We do not generate images online using generators on CIFAR-10. Instead, we generate 500 images each class and merge them with the original training set at first. Adam optimizer is used with learning rate decay(start from 1e-3) in each epoch \{200, 220, 240, 260\}. Results of accuracy on testing set are shown in Table \ref{table:accuracy} training details are attached in Appendix Figure \ref{fig:training}.

When training ResNet, we find that the testing accuracy of WGAN\_GP, Ours\_1 and Ours\_2 is a little volatile. However, on CIFAR-10, this phenomenon is not such obvious. We attribute this problem to bad samples. Because on MNIST and Fashion-MNIST, we generate auxiliary images (5% of batch size) online, and thus in some cases, we may get low-quality images which will result in sharp decline of testing accuracy. On CIFAR-10, we merge the generated images with the original training data first, and then shuffle them in each epoch. This can reduce the chance of concentration of
Figure 7: Alleviation of mode collapse via DPM. (a) WGAN_GP without DPM. (b) WGAN_GP with DPM $\lambda=5$. The value above each image pair indicates the similarity value of their latent vectors. We can see that in GAN without DPM, latent vectors with low similarity value can be mapped to similar images while DPM not.

Table 3: Testing Accuracy on Several Datasets.

| Testing Acc | MNIST | Fashion-MNIST | CIFAR-10 |
|-------------|-------|---------------|----------|
| Baseline    | 0.9897 | 0.9257        | ×        |
| DA          | ×     | ×             | 0.9172   |
| WGAN_GP     | 0.9951 | 0.9394        | ×        |
| Ours_1 $\lambda = 5$ | **0.9975** | 0.9465        | **0.9239** |
| Ours_2 $\lambda = 10$ | 0.9969 | **0.9527**    | 0.9212   |

$\lambda$: DPM coefficient, see formula (11)

Table 4: Quantitative results on 2D Synthetic Dataset.

|                      | 2D Ring modes | h-q | 2D Grid modes | h-q |
|----------------------|---------------|-----|---------------|-----|
| GAN                  | 1.0           | ×   | 17.7          | 82.3|
| ALI                  | 2.8           | ×   | 12.8          | 1.6 |
| Unrolled GAN         | 7.6           | 87.97 | 14.9 | 4.89 |
| VEEGAN               | 8.0           | 86.77 | 24.4 | 77.16 |
| PacGAN               | 7.8           | 98.21 | 24.3 | 79.46 |
| BourGAN              | 8.0           | 99.76 | 25.0 | 95.91 |
| BourGAN_DP           | 8.0           | 99.89 | 25.0 | 95.99 |
| GAN_DP               | 2.0           | ×   | 21.3          | 80.8|
| Unrolled GAN_DP      | 8.0           | 99.36 | 21.7 | 75.21 |
| BourGAN_DP           | 8.0           | 99.89 | 25.0 | 95.99 |

h-q: percentage of generated samples in high-quality.

From both Figure 8 and Table 4 we can see that the diversity penalty help the vanilla GAN capture more modes of the data distribution, and it outperforms the ALI (Dumoulin et al. 2016) and Unrolled GAN (Metz et al. 2016) on 2D Grid Dataset while closer to VEEGAN (Srivastava et al. 2017) and PacGAN (Lin et al. 2018). Besides, from BourGAN and BourGAN_DP in Figure 8 we can see that the group with diversity penalty module can converge to a better solution than its blank counterpart.

2D Synthetic Datasets

On synthetic dataset, we can get the quantitative evaluation results of mode collapse accurately, because the distribution of data and its modes are known. As in (Dumoulin et al. 2016; Lin et al. 2018; Metz et al. 2016; Srivastava et al. 2017; Xiao, Zhong, and Zheng 2018), we evaluate our proposed method on 2D Ring and 2D Grid. 2D Ring dataset contains eight 2D Gaussian distributions whose centers locate on a ring equally. 2D Grid contains twenty-five 2D Gaussian distributions whose centers locate on the meshgrid of a square. We use the official code and architecture in (Xiao, Zhong, and Zheng 2018), besides, we apply our proposed diversity penalty term on the vanilla GAN (Goodfellow et al. 2014), Unrolled GAN (Metz et al. 2016) and BourGAN (Xiao, Zhong, and Zheng 2018). We use the number of modes captured by generator and the percentage of points generated by generator in high-quality as metrics. As in (Srivastava et al. 2017), we count a sample as high quality, if it is within three standard deviations of the nearest mode, and the number of modes captured by generator is the number of Gaussian centers which are nearest to at least one high quality sample. Our visual results are shown in Figure 8 and the quantitative metrics results are shown in Table 4.

Comparison of different GANs on CelebA

We compare the quality and diversity of images generated by generator in different GANs on CelebA (Liu et al. 2015). The GANs are split into two groups which are DCGAN series with {DCGAN, DCGAN_MS, DCGAN_DP} and WGAN_GP series with {WGAN_GP, WGAN_GP_MS, WGAN_GP_DP}. Here we use suffix MS to represent the mode seeking regularization proposed in (Mao et al. 2019), and DP to indicate our proposed diversity penalty module. The coefficient $\lambda_{ms}$ of MS used in (Mao et al. 2019) is set to

https://github.com/a554b554/BourGAN
In 2D Ring GAN DP, the black points indicate the centers of mixture gaussian distribution. From the first two columns we can see diversity penalty term can help the vanilla GAN capture more modes, especially in 2D Grid, the GAN with diversity penalty captures four more modes than the vanilla counterpart. Also the results of BourGAN DP is smoother than the original architecture. The results of Unrolled GAN (Metz et al. 2016) and VEEGAN (Srivastava et al. 2017) are from paper (Xiao, Zhong, and Zheng 2018).

In 2D Ring GAN DP, the black points indicate the centers of mixture gaussian distribution. From the first two columns we can see diversity penalty term can help the vanilla GAN capture more modes, especially in 2D Grid, the GAN with diversity penalty captures four more modes than the vanilla counterpart. Also the results of BourGAN DP is smoother than the original architecture. The results of Unrolled GAN (Metz et al. 2016) and VEEGAN (Srivastava et al. 2017) are from paper (Xiao, Zhong, and Zheng 2018).

1, and the penalty coefficient $\lambda$ shown in formula (11) is set to 10. We use an Adam optimizer with $\beta_1=0.5$ and $\beta_2=0.9$, and the learning rate is set to $1e^{-4}$. All GANs are trained with a batch size of 128 and 100 epochs in total. The details of architecture are attached in Appendix Table 7.

In Figure 9, we randomly sample two images from CelebA, and then use the back propagation method to optimize the input noise so that we can get the corresponding latent vectors of these two images. From the results, we can see that the WGAN GP series performs better than DCGAN series visually, moreover, from Figure 9 (2), we find that the MS group tends to generate some noisy points because it measures similarity using L1-norm between pixels in data space, and in some cases, this will result in mutation of pixel values for improving the difference among fake images. Thinking if we move images with a very small step, and this will result in great difference in data space while little change in feature representations. By comparison, our proposed diversity penalty tends to reduce the similarity of images in feature space which is more reasonable and robust. Moreover, in Figure 9 the transition between images in MS group is not smooth since the man with glasses appears only in the last two images.

The quantitative results are evaluated with IS and FID. When we calculate the FID between fake images and the original, all images ($\approx 203k$) in CelebA (Liu et al. 2015) are used. $n_{splits}$ in IS is set to 10. Results are shown in Table 5. Both IS and FID suggest that diversity penalty module do have positive effects on improving data diversity while ensure the quality of images.

Table 5: IS and FID results on CelebA.

|               | DCGAN   | DCGAN_MS | DCGAN_DP |
|---------------|---------|----------|----------|
| ↑IS           | 2.113±0.014 | 2.360±0.006 | 2.379±0.013 |
| ↓FID          | 24.23±0.150 | 23.51±0.090 | 21.76±0.110 |
|               | WGAN_GP | WGAN_GP_MS | WGAN_GP_DP |
| ↑IS           | 2.775±0.018 | 2.927±0.016 | 2.941±0.021 |
| ↓FID          | 33.48±0.011 | 24.86±0.020 | 24.18±0.031 |

Conclusions

In this paper, we present a pluggable block called diversity penalty module to alleviate mode collapse in GAN. We use this penalty term to enforce the similarity between feature pairs to be consistent with that between latent vector pairs, therefore, we can expand the feature space of generator, and get more different images by changing latent vectors greatly. The advantage of our proposed method is its generalization, it almost can be combined with all GANs in different architectures. Results of experiment suggest that our penalty term is effective and has good generalization performance. In domain translation, DPM can help StarGAN generate much more accurate attention masks, and on 2D synthetic dataset, both visual and quantitative results suggest that DPM can help different GANs capture more modes. Moreover, in downstream tasks such as image data augmentation, diversity penalty method also makes a considerable improvement on GAN-based data augmentation. More generated samples and details are attached in Appendix.
References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein gan. arXiv preprint arXiv:1701.07875.

Barratt, S.; and Sharma, R. 2018. A note on the inception score. arXiv preprint arXiv:1801.01973.

Choi, Y.; Choi, M.; Kim, M.; Ha, J.-W.; Kim, S.; and Choo, J. 2018. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In Proceedings of the IEEE conference on computer vision and pattern recognition, 8789–8797.

Dumoulin, V.; Belghazi, I.; Poole, B.; Mastropietro, O.; Lamb, A.; Arjovsky, M.; and Courville, A. 2016. Adversarially learned inference. arXiv preprint arXiv:1606.00704.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In Advances in neural information processing systems, 2672–2680.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of wasserstein gans. In Advances in neural information processing systems, 5767–5777.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Advances in neural information processing systems, 6626–6637.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, 1097–1105.

LeCun, Y. 1998. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/.

Lin, Z.; Khetan, A.; Fant, G.; and Oh, S. 2018. Pacgan: The power of two samples in generative adversarial networks. Advances in neural information processing systems.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep Learning Face Attributes in the Wild. In Proceedings of International Conference on Computer Vision (ICCV).

Mao, Q.; Lee, H.-Y.; Tseng, H.-Y.; Ma, S.; and Yang, M.-H. 2019. Mode seeking generative adversarial networks for diverse image synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1429–1437.

Metz, L.; Poole, B.; Pfau, D.; and Sohl-Dickstein, J. 2016. Unrolled generative adversarial networks. arXiv preprint arXiv:1611.02163.

Perez, L.; and Wang, J. 2017. The effectiveness of data augmentation in image classification using deep learning. arXiv preprint arXiv:1712.04621.
Appendices

Figure 10: Image Classification using GAN-based Data Augmentation on Several Datasets.

Table 6: Architectures of GAN in Keras-like style on MNIST / Fashion-MNIST / CIFAR-10.

| Names | MNIST / Fashion-MNIST | CIFAR-10 |
|-------|-----------------------|----------|
| Layer_1 | Input Size | Dense | 100x1 | Dense | 128x1 |
| | Output Size | BN ReLU Reshape | 7x7x24 | ReLU Reshape | 4x4x512 |
| Layer_2 | Input Size | Conv2DTranspose | 14x14x12 | Conv2DTranspose | 8x8x256 |
| | Output Size | BN ReLU | 14x14x6 | ReLU | 16x16x128 |
| Layer_3 | Input Size | Conv2DTranspose | 14x14x6 | Conv2DTranspose | 8x8x256 |
| | Output Size | BN ReLU | 28x28x1 | Tanh | 32x32x3 |
| Trainable Params | 124477 | 5162755 |

| Layer_1 | Input Size | Conv2D | 28x28x1 | Conv2D | 32x32x3 |
| | Output Size | LeakyReLU | 14x14x6 | LeakyReLU | 16x16x128 |
| Layer_2 | Input Size | Conv2D | 14x14x6 | Conv2D | 16x16x128 |
| | Output Size | BN LeakyReLU | 7x7x12 | LeakyReLU | 8x8x256 |
| Layer_3 | Input Size | Conv2D | 7x7x12 | Conv2D | 8x8x256 |
| | Output Size | LeakyReLU | 4x4x24 | LeakyReLU | 4x4x512 |
| Layer_4 | Input Size | Flatten | 4x4x512 | Flatten | 4x4x512 |
| | Output Size | Dense | 1 | Dense | 1 |
| Trainable Params | 9649 | 4114689 |
Table 7: Architectures of GAN in Keras-like style on CelebA.

| Names         | DCGAN Series(GP,DP,MS) | WGAN_GP Series(GP,DP,MS) |
|---------------|------------------------|--------------------------|
| Layer_1       |                         |                          |
| Input Size    | Dense Reshape           | 100x1                    |
| Output Size   | BN ReLU                 | 4x4x1024                 |
|               |                        |                          |
| Layer_2       | Conv2DTranspose         | 4x4x1024                 |
| Input Size    | Conv2DTranspose         | 4x4x1024                 |
| Output Size   | ReLU                    | 8x8x512                  |
|               |                        |                          |
| Layer_3       | Conv2DTranspose         | 8x8x512                  |
| Input Size    | Conv2DTranspose         | 8x8x512                  |
| Output Size   | BN ReLU                 | 16x16x256               |
|               |                        |                          |
| Layer_4       | Conv2DTranspose         | 16x16x256               |
| Input Size    | Conv2DTranspose         | 16x16x256               |
| Output Size   | ReLU                    | 32x32x128               |
|               |                        |                          |
| Layer_5       | Conv2DTranspose         | 32x32x128               |
| Input Size    | Conv2DTranspose         | 32x32x128               |
| Output Size   | Tanh                    | 64x64x3                 |
| Trainable Params |                        | 12679171              |

Discriminator

| Layer_1       |                         |                          |
| Input Size    | Conv2D                 | 64x64x3                  |
| Output Size   | LeakyReLU              | 32x32x128               |
|               |                        |                          |
| Layer_2       | Conv2D BN              | 32x32x128               |
| Input Size    | Conv2D                | 32x32x128               |
| Output Size   | LeakyReLU             | 16x16x256               |
|               |                        |                          |
| Layer_3       | Conv2D BN              | 16x16x256               |
| Input Size    | Conv2D                | 16x16x256               |
| Output Size   | LeakyReLU             | 8x8x512                 |
|               |                        |                          |
| Layer_4       | Conv2D BN              | 8x8x512                 |
| Input Size    | Conv2D                | 8x8x512                 |
| Output Size   | LeakyReLU             | 4x4x1024                |
|               |                        |                          |
| Layer_5       | Flatten                | 4x4x1024                |
| Input Size    | Dense Sigmoid         | 4x4x1024                |
| Output Size   | Dense                 | 8x8x512                 |
| Trainable Params |                        | 11038081              |

Figure 11: Convergence on Several Datasets. In images above, the lines indicate the averaging loss of 10 generators, the band in light color marks out the boundaries of all losses. Since the loss of discriminator is almost the same in each groups (WGAN_GP, Ours_1 and Ours_2), we just give the results on generator here.
Figure 12: Alleviation of mode collapse via DPM on several datasets.

Figure 13: Generated Images on MNIST.

Figure 14: Generated Images on Fashion-MNIST.
Figure 15: Generated Images on CIFAR-10.

Figure 16: Generated Images on CelebA.
**Figure 17:** Training loss of generator and discriminator on CelebA. Diversity penalty coefficient $\lambda$ in WGAN$_{GP}$DP and DCGAN$_{DP}$ is set to 10. $\lambda_{ms}$ in WGAN$_{MS}$ and DCGAN$_{MS}$ is set to 1.

**Figure 18:** Reduction of data diversity detected in our experiments. We use the adversarial learning method to get two similar images under the MSE metrics. Given random noise $z_1$ and its corresponding generated image $img_1$, we optimize random noise $z_2$ using back propagation to minimize the difference between its corresponding image $img_2$ and $img_1$. Our results show that $z_2$ may be very different from $z_1$, even though their corresponding generated images are similar. (a) WGAN$_{GP}$ (Gulrajani et al. 2017) on MNIST (LeCun, 1998): some examples of very different noise vectors which are mapped to similar images. The value above each image pair indicates the similarity of their corresponding noise vectors. (b) WGAN$_{GP}$ with diversity penalty on CelebA (Liu et al. 2015): results show that the similar images also have corresponding latent vectors with higher similarity compared with (b). Calculation of similarity is shown in formula (3).