DMGAN: Discriminative Metric-based Generative Adversarial Networks

Zhangling Chen[a], Ce Wang[b,*], Huaming Wu[a], Kun Shang[c], Jun Wang[d]

[a] *Center for Applied Mathematics, Tianjin University, Tianjin 300072, P.R. China*
[b] *Center for Combinatorics, Nankai University, Tianjin 300071, P.R. China*
[c] *College of Mathematics and Econometrics, Hunan University, Changsha, Hunan 410082, P.R. China*
[d] *School of Mathematics, Tianjin University, Tianjin 300072, P.R. China*

Abstract

With the proposed of Generative Adversarial Networks (GANs), the generative adversarial models have been extensively studied in recent years. Although probability-based methods have achieved remarkable results in image synthesis tasks, there are still some unsolved challenges that are difficult to overcome. In this paper, we propose a novel model, called Discriminative Metric-based Generative Adversarial Networks (DMGANs), for generating real-like samples from the perspective of deep metric learning. To be specific, the generator is trained to generate realistic samples by reducing the distance between real and generated samples. Instead of outputting probability, the discriminator in our model is conducted as a feature extractor, which is well constrained by introducing a combination of identity preserving loss and discriminative loss. Meanwhile, to reduce the identity preserving loss, we calculate the distance between samples and their corresponding center and update these centers during training to improve the stability of our model. In addition, a data-dependent strategy of weight adaption is proposed to further improve the quality of generated samples. Experiments on several datasets illustrate the potential of our model.

Keywords: Generative Adversarial Networks, Deep Metric Learning, Image Generation, Weight Adaption

[*] Corresponding author.
Email address: `1120150001@mail.nankai.edu.cn` (Ce Wang)

November 26, 2019

1. Introduction

Generative Adversarial Networks (GANs) [1] as a convincing branch of deep generative models have attracted tremendous attention. Specifically, the emergence of GANs has brought significant improvements in many tasks, such as image generation [2, 3], image super-resolution [4], image-to-image translation [5, 6], and other related applications [7, 8, 9, 10]. Compared to deep Boltzmann machines [11] or generative stochastic networks [12], which require intractable probabilistic computations explicitly, GANs avoid these computations by deriving back-propagation signals through a competitive process involving a pair of networks. Nevertheless, vanilla GAN [1] could only generate low-resolution gray-scale samples, yet the training process of vanilla GAN is notoriously difficult and often suffers from mode collapse. To alleviate these problems, researchers have explored various aspects of GANs, such as the choice of the architectures [13, 14, 15], regularization and normalization schemes [16, 17], and the design of loss functions [18, 19, 20]. Even though tremendous improvements [21, 15] have been achieved, these models still pay little attention to deep metric learning, which is widely applied in supervised classification tasks.

As a popular method for extracting more discriminative features, deep metric learning has witnessed its success in classification tasks, such as face recognition [22, 23] and objective recognition [24, 25]. By designing appropriate objective functions, deep metric learning approaches [26, 27, 28] can obtain intra-class compact and inter-class separable features and achieve state-of-the-art results on many tasks. The success of deep metric learning in achieving classification tasks has motivated researchers to investigate the use of deep metric learning in other relevant tasks such as image generation. Recently, MBGAN [29] and MLGAN [30] apply deep metric learning to GAN models to generation tasks. They view the discriminator as a feature extractor that maps samples into a feature space, where the distances between real samples are minimized as well as the distances between real and fake samples are maximized. At the same time, the generator is trained to generate samples that are close to real data under the learned metric. Furthermore, by adding a term called "center penalty", which punishes the discriminator if it learns inappropriate features for images away from their predefined center vectors, MLGAN improves the quality of generated images. However, it is still limited because hand-engineered centers are inflexible, i.e., they cannot suit the distribution of data during training. On the other hand, MBGAN adopts a data-dependent margin and needs a triplet of samples in each iteration. However, they only calculate the distance between samples, which means they can not effectively utilize the information on the distribution of data and are sensitive to samples with noise during training. It is difficult to get enough representative features only by calculating the distance between samples in transformed space.

Inspired by the works mentioned above, we propose a novel generative adversarial model from the perspective of deep metric learning, named Discriminative Metric-based Generative Adversarial Network (DMGAN). Different from traditional GANs, the generator in our model aims to capture the distribution of real data by reducing the distance between synthesized images and real ones in feature space. Simultaneously, we conduct the discriminator as a feature extractor that maps samples into a latent feature space to measure whether a given sample belongs to real data or not. Similar to [31, 32, 33] which optimize their model with group decision making (GDM) method, our discriminator is trained under the joint supervision of discriminative loss and identity preserving loss. On the one hand, we maximize the distance between real and fake samples using discriminative loss. On the other hand, the identity preserving loss is optimized to minimize the distance between samples and their corresponding centers in feature space. It should be noted that centers of samples utilized in our model are constantly updated during the training process following the strategy in center loss [28]. Thus the discriminator can extract illustrative features to distinguish real samples from false ones as well as faithfully preserve the local structure of samples in feature space. To further improve the quality of samples generated by our model, we introduce a data-dependent weight adaptive strategy for the discriminative loss. That is to say, if the distance between generated samples and real samples in features is large, the corresponding weight will be small, otherwise, the weight will be large. With the adaptive strategy, our model can focus more attention on improving those poorly-produced samples instead of wasting energy on well-produced samples.

The main contributions of our work lie in four folds:

- We propose the Discriminative Metric-based Generative Adversarial Network (DMGAN) with a simple and robust training procedure from the perspective of deep metric learning.

- We combine the discriminative loss and identity preserving loss to exactly recover the implicit distribution of real data. Furthermore, we integrate identity preserving loss and discriminative loss using an adaptive weight dependent on data to drive the model to pay more attention to improving those poorly-produced samples.

- We calculate the center of samples according to the labels of samples and then minimize the distance between samples and their data-dependent centers, so that our model can learn representative features in transformed space.
- We adopt the point that we can generate samples with the same distribution as real samples by using a deep metric learning method. Experimental results demonstrate that our model outperforms state-of-the-art results on several datasets.

2. Related works

We briefly review prior works related to our proposed approach in this section. For clarity, we group them into two aspects: deep metric learning and generative adversarial networks.

2.1. Deep metric learning

Facing with large amounts of data and complex deep models, researchers put forward deep metric learning methods, which adopt conventional metric learning approaches on the top of deep features. Generally, deep metric learning methods are utilized to learn powerful deep nonlinear transformations into a feature space whose metric is in correspondence with a predefined similarity. As a typical deep metric learning method, contrastive loss [26] learns a globally coherent nonlinear function that minimizes intra-class distance and forces inter-class distance to be larger than a fixed margin. On the other hand, rather than a pair of samples, triplet loss [27] requires a triplet of training samples as input and minimizes the distance between an anchor sample and a positive sample while maximizes the distance between the anchor sample and a negative sample, which is to make the inter-class gap distance larger than the intra-class gap by a margin relatively. However, the applications of contrastive loss and triplet loss are limited because penalizing pairs or triplets of samples suffer from dramatic data expansion. To alleviate this problem, center loss [28] targets more directly on the learning objective of the intra-class variations by constraining the distance between samples and their corresponding centers, which is very beneficial to the discriminative feature learning. Actually, through the joint supervision of center loss and softmax loss, the discriminative power of deep features can be highly enhanced. Furthermore, each class of samples in magnet loss [34] are further grouped into several clusters and local discrimination is achieved by adaptively penalizing the distance between samples and their clustering centers. In summary, as an essential statistic of samples, the center plays a crucial role in many deep metric learning algorithms, and the success of utilizing deep metric learning algorithms on classification tasks motivates us to devote more efforts to improving generative adversarial models from the perspective of deep metric learning.

2.2. Generative adversarial networks

GAN [1] is a machine learning technique that learns to generate fake samples indistinguishable from real ones via a competitive game. The architectures of GAN are composed of two neural networks, a discriminator and a generator. The discriminator D is equipped to maximize the probability of assigning correct labels to both real samples and generated samples while the generator G is trained to fool the discriminator with synthesized data. During the last few years, a large amount of GANs [35, 36, 37] have been proposed in two categories: unconditional GANs and conditional GANs.

As the primitive generative adversarial model, vanilla GAN [1] always encounters training instability and mode collapse during the process of achieving the Nash equilibrium of the generator and the discriminator. To alleviate the problem of mode collapse and increase the stability of the model, CatGAN [38] puts forward an unconditional categorical generative adversarial model by utilizing mutual information between real and generated samples. Besides, MLGAN [30] pays its attention to the way to measure the similarity between the distribution of real data and synthesized samples and proposes a novel model based on distance metric learning without condition. Recently, KM-GAN [39] presents an unconditional generative adversarial model by incorporating the idea of updating centers in K-means into GANs. Although the quality of samples generated by these unconditional GANs has exceedingly improved, they always suffer from problems during training.

To solve the problems mentioned above and further improve the performance of generative adversarial models, researchers start to pay more attention to conditional GANs [13, 18, 29, 40]. Indeed, CGAN [40] has greatly improved the model stability and quality of synthesized samples by fusing one-hot labels into the adversarial learning process. Subsequently, DCGAN [13] designs a stable architecture utilizing convolutional neural networks and provides several tricks to stabilize the adversarial training of conditional GANs. Based on these efforts, a growing number of conditional GANs [2, 6, 14, 18, 19, 20, 21, 41] are proposed. Among them, some conditional GANs [2, 6, 14, 21] dedicate to redesigning the architecture of models while some models [18, 19, 20, 41] adopt different criteria to distinguish between real and fake samples. For instance, EBGAN [19] regards the discriminator as an energy function, and LSGAN [20] adopts the least square loss for the discriminator. Inspired by the successful utilization of deep metric learning in the tasks of supervised classification, MBGAN further [29] extends the framework of GAN from the perspective of deep metric learning. To be specific, the discriminator adopts a triplet of inputs and learns a nonlinear transformation to map these samples from the original space into a feature space. However, only penalizing triplets of samples can not employ sufficient insights of data structure, which would hinder the performance of the model. The key challenge for generating high-quality images is whether the discriminator can learn representative features for metric-based generative models. Therefore, it is desirable to tell the algorithm to concentrate on the statistics of features in representation space for extracting illustrative features as well as generating more realistic images.

3. Proposed method

In this section, we introduce our generative adversarial model, Discriminative Metric-based Generative Adversarial Network (DMGAN), which borrows the idea from deep metric learning. Firstly, we give a detailed description of our model in regular. Then a strategy of weight adaption is introduced to improve the performance of DMGAN.

3.1. Regular DMGAN

The diagram in Figure 1 shows the framework of our model. Give a random vector $\boldsymbol{z} \sim p_{\boldsymbol{z}}$, the generator G directly learns a mapping that maps the latent variable $\boldsymbol{z}$ to a real-like fake sample $G(\boldsymbol{z})$. The discriminator, as a feature extractor, utilizes the proposed metric function to distinguish real samples from synthesized ones. To be specific, the discriminator embeds the real sample $\boldsymbol{x}$ or generated sample $G(\boldsymbol{z})$ into a feature space where samples are measured by Euclidean distance. Indeed, many different distance metrics can be selected for DMGAN, and we focus on Euclidean distance for ease of

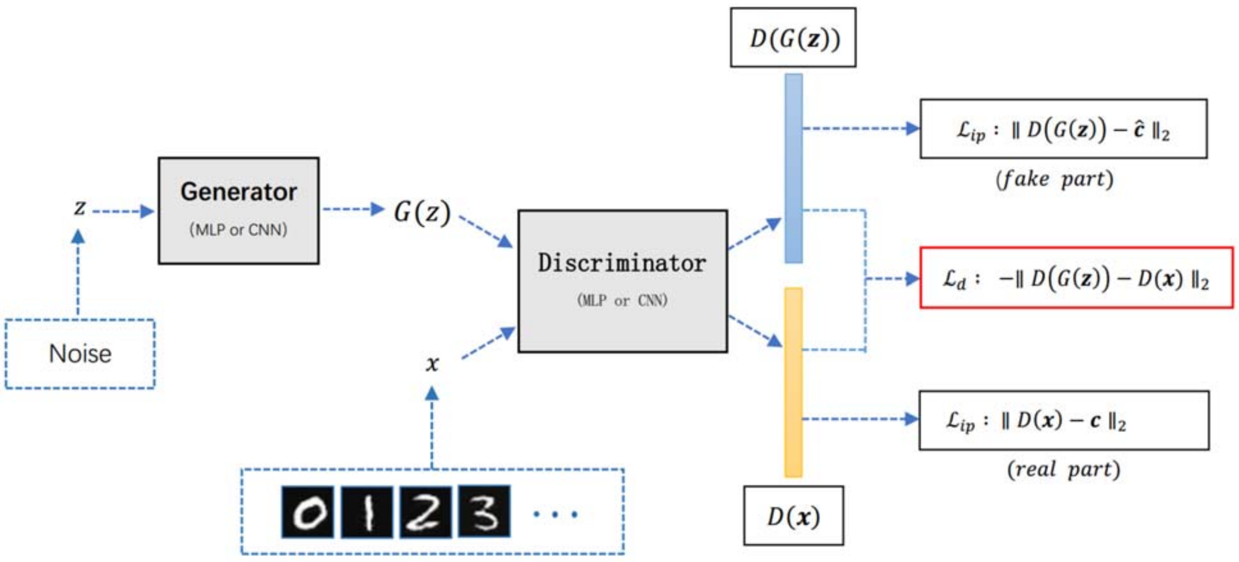

$D(G(\boldsymbol{z}))$
$\mathcal{L}_{ip}$: $\| D(G(\boldsymbol{z})) - \hat{\boldsymbol{c}} \|_2$
$(fake\ \ part)$
z
Generator
(MLP or CNN)
$G(z)$
Discriminator
(MLP or CNN)
$\mathcal{L}_d$: $-\| D(G(\boldsymbol{z})) - D(\boldsymbol{x}) \|_2$
Noise
x
$\mathcal{L}_{ip}$: $\| D(\boldsymbol{x}) - \boldsymbol{c} \|_2$
$(real\ \ part)$
0 1 2 3 ...
$D(\boldsymbol{x})$

Figure 1: Architectures of the proposed DMGAN. Similar to regular GANs, the generator and discriminator in DMGAN can be realized by multi-layer perceptron (MLP) or convolutional neural network (CNN). The generator aims to synthesize realistic images while the discriminator aims to extract representative features through joint supervision of identity preserving loss $\mathcal{L}_{ip}$ and discriminative loss $\mathcal{L}_d$. $\widehat{\boldsymbol{c}}$ and $\boldsymbol{c}$ in the objective of $\mathcal{L}_{ip}$ represent the centers corresponding to samples $G(\boldsymbol{z})$ and $\boldsymbol{x}$.

presentation. The discussion and analysis can easily be extended to other types of metrics. To accurately measure the distance between generated samples and real samples, we adopt a group decision making method and introduce an objective for the discriminator that contains two parts, i.e., discriminative loss and identity preserving loss. The discriminative loss is used to enlarge the distance between real samples and fake ones so that real and fake samples can be distinguished, and the specific objective function is listed as follows:

$$\mathcal{L}_d = - \parallel D(G(\boldsymbol{z})) - D(\boldsymbol{x}) \parallel_2 \quad (discriminative\ \ loss) \tag{1}$$

where $D(\boldsymbol{x})$ and $D(G(\boldsymbol{z}))$ are the output features of real and generated samples of the discriminator, respectively.

Since $\mathcal{L}_d$ is a part of the loss function of our discriminator for a pair of dissimilar samples, we can separate real samples and synthetic samples in feature space by minimizing the objective function $\mathcal{L}_d$. Nevertheless, only optimizing $\mathcal{L}_d$ during training cannot guarantee that the features learned by the discriminator are representative. Hence, we introduce identity preserving loss to the discriminator to learn robust features. To be specific, as another part of the objective function of the discriminator, the identity preserving loss tries to push each sample close to its corresponding center. Let $\boldsymbol{c}$ and $\widehat{\boldsymbol{c}}$ denote centers of deep features $D(\boldsymbol{x})$ and $D(G(\boldsymbol{z}))$, then the objective can be formulated as follows:

$$\mathcal{L}_{ip} = \parallel D(\boldsymbol{x}) - \boldsymbol{c} \parallel_2 + \parallel D(G(\boldsymbol{z})) - \widehat{\boldsymbol{c}} \parallel_2 \quad (\textit{identity preserving loss}) \tag{2}$$

L_{ip} is an objective to minimize the intra-class variations by enforcing $D(\boldsymbol{x})$ (or $D(G(\boldsymbol{z}))$) to have small distance with its corresponding center $\boldsymbol{c}$ (or $\widehat{\boldsymbol{c}}$) in feature space. Centers in our model share the same dimension with the output of the discriminator and are initialized to $(0, 0, \cdots, 0)$. Furthermore, to alleviate the limitation of hand-engineered centers on training, we constantly update them during training as deep features of samples are changed. The updated criteria is computed as:

$$\triangle \, \boldsymbol{c}_i = \frac{\sum_{j=1}^{m} \delta(y_j = i)(\boldsymbol{c}_i - \boldsymbol{x}_j)}{1 + \sum_{j=1}^{m} \delta(y_j = i)} \tag{3}$$

$$\boldsymbol{c}_i^{new} = \boldsymbol{c}_i - \gamma \cdot \triangle \boldsymbol{c}_i \tag{4}$$

where m is the size of mini-batch. δ is an indicator function that means $\delta(condition) = 1$ if the *condition* is satisfied, and $\delta(condition) = 0$ if not. $\boldsymbol{c}_i$ represents the center of deep features of real samples in category i. If there is no label in given data, the model treats all the training data as the same class, that means $\mid i \mid= 1$ ($\mid i \mid$ represents the number of different i). On the contrary, $\mid i \mid = k$ ($k > 1$), where k refers to the number of classes of data. $\widehat{c}$ has the same update rule as c, but the difference is that updating $\widehat{c}$ depends on generated samples. γ is a hyper-parameter introduced for controlling the updated ratio of data-dependent centers. When the value of γ is large, the new data-dependent center depends more heavily on the features of the current stage and has less memory of the previous features. On the other hand, when γ is small, it will depend heavily on the center of the last step. So we can see that fixing centers in MLGAN can be considered as a special case when γ is set to 0.

To learn more discriminative features and accurately distinguish real samples from generated ones, we adopt the joint supervision of $\mathcal{L}_d$ and $\mathcal{L}_{ip}$ to train the discriminator, and the final objective is a weighted sum of $\mathcal{L}_d$ and $\mathcal{L}_{ip}$:

$$\min_D \mathcal{L}_D = \mathcal{L}_{ip} + \lambda \cdot \mathcal{L}_d \tag{5}$$

Algorithm 1 Training algorithm for DMGAN

```
Input: Training set $\boldsymbol{X}$, random noise $\boldsymbol{z} \sim P_{\boldsymbol{z}}$, batch size $m$, hyper-parameters $\lambda, \gamma$, number of epochs $T$, Adam hyper-parameters $\alpha, \beta_1, \beta_2$
Output: Generated samples $G(\boldsymbol{z})$
  Initialize parameters of $D$ and $G$
  Initialize centers $\boldsymbol{c} = \widehat{\boldsymbol{c}} = (0, 0, \cdots, 0)$
  for $t = 1 : T$ do
    Sample $m$ samples $\{\boldsymbol{x}_i\}_{i=1}^m$ from real data $\boldsymbol{X}$
    Sample $m$ noise samples $\{\boldsymbol{z}_i\}_{i=1}^m$ from random noise distribution $P_{\boldsymbol{z}}$
    $\mathcal{L}_D = \mathcal{L}_{ip} + \lambda \cdot \mathcal{L}_d$
    $grad_{\theta_d} = \nabla_{\theta_d}\mathcal{L}_{ip} + \lambda \cdot \nabla_{\theta_d}\mathcal{L}_d$
    $\theta_d = Adam(grad_\theta, \theta_d, \alpha, \beta_1, \beta_2)$
    $\mathcal{L}_G = -\mathcal{L}_d$
    $grad_{\theta_g} = \nabla_{\theta_g}\mathcal{L}_G$
    $\theta_g = Adam(grad_\theta, \theta_g, \alpha, \beta_1, \beta_2)$
    Update centers $\boldsymbol{c}$ and $\widehat{\boldsymbol{c}}$ by $\boldsymbol{c}^{t+1} = \boldsymbol{c}^t - \gamma \cdot \triangle\boldsymbol{c}^t$
  end for
```

where λ is a predefined hyper-parameter to govern the relative importance of discriminative loss compared with the identity preserving loss. On the other hand, the generator attempts to synthesize real-like samples by minimizing the distance between real samples and generated samples in feature space, and the objective of the generator is listed as:

$$\min_G \mathcal{L}_G = -\mathcal{L}_d \tag{6}$$

In DMGAN, the generator and the discriminator can be trained with stochastic gradient descent (SGD) [42] by backpropagation. The details of the learning algorithm are given in Algorithm 1.

3.2. waDMGAN

In regular DMGAN, the discriminative loss L_d assigns the same weight for different generated samples, although some synthesized samples are of good quality while others are not. This way of setting weights limits the convergence of our model due to the lack of considering the difference between samples. In this section, instead of using a fixed weight of $\mathcal{L}_d$ as in

Algorithm 2 Training algorithm for waDMGAN

```
Input: Training set $\boldsymbol{X}$, random noise $\boldsymbol{z} \sim P_{\boldsymbol{z}}$, batch size $m$, hyper-parameters $\lambda, \gamma$, number of epochs $T$, Adam hyper-parameters $\alpha, \beta_1, \beta_2$
Output: Generated samples $G(\boldsymbol{z})$
  Initialize parameters of $D$ and $G$
  Initialize centers $\boldsymbol{c} = \widehat{\boldsymbol{c}} = (0, 0, \cdots, 0)$
  for $t = 1 : T$ do
    Sample $m$ samples $\{\boldsymbol{x}_i\}_{i=1}^m$ from real data $\boldsymbol{X}$
    Sample $m$ noise samples $\{\boldsymbol{z}_i\}_{i=1}^m$ from random noise distribution $P_{\boldsymbol{z}}$
    for $i = 1 : m$ do
      $weight_i = \exp\{\frac{\frac{1}{m}\sum_{i=1}^m \|G(\boldsymbol{z}_i)-\boldsymbol{x}_i\|_1}{\|G(\boldsymbol{z}_i)-\boldsymbol{x}_i\|_1}\}$
    end for
    $weight = (weight_1, weight_2, \cdots, weight_m)$
    $\mathcal{L}_D = \mathcal{L}_{ip} + \lambda \cdot weight \cdot \mathcal{L}_d$
    $grad_{\theta_d} = \nabla_{\theta_d}\mathcal{L}_{ip} + \lambda \cdot weight \cdot \nabla_{\theta_d}\mathcal{L}_d$
    $\theta_d = Adam(grad_\theta, \theta_d, \alpha, \beta_1, \beta_2)$
    $\mathcal{L}_G = -\mathcal{L}_d$
    $grad_{\theta_g} = \nabla_{\theta_g}\mathcal{L}_G$
    $\theta_g = Adam(grad_\theta, \theta_g, \alpha, \beta_1, \beta_2)$
    Update centers $\boldsymbol{c}$ and $\widehat{\boldsymbol{c}}$ by $\boldsymbol{c}^{t+1} = \boldsymbol{c}^t - \gamma \cdot \triangle\boldsymbol{c}^t$
  end for
```

Eq. 5, we improve regular DMGAN by providing a data-dependent weight adaptive strategy. That is to say, we assign different weights to different samples according to the quality of samples generated during the training process. By adding the data-dependent weights, the model can automatically adapt the weights to guide the discriminator to extract more robust and representative features and make the generator pay more attention to improving poor-produced samples. The ith adaptive weight is defined as follows:

$$weight_i = \exp\left\{\frac{\frac{1}{m}\sum_{i=1}^m \parallel G(\boldsymbol{z}_i) - \boldsymbol{x}_i \parallel_1}{\parallel G(\boldsymbol{z}_i) - \boldsymbol{x}_i \parallel_1}\right\} \quad (i = 1, 2, \cdots, m) \tag{7}$$

Given a mini-batch samples, we calculate the pixel-wise gap between each real and generated sample and then count the average distance of batch samples in each step of training. When the distance between real samples and synthesized samples is smaller than the average value, the weight is larger. Similarly, the weight will be a lower value if the distance between the real and generated samples is larger than the average distance. Besides, we add an exponential term to change the degree of the variation of weights and the objective function of the discriminator with weight adaption is as follows:

$$\min_{D} \mathcal{L}_D = \mathcal{L}_{ip} + \lambda \cdot weight \cdot \mathcal{L}_d \tag{8}$$

With this more relaxed condition, the discriminator in our model can obtain more robust and discriminative features, thus the strategy of weight adaption is an efficient way for generating more realistic images. For convenience, we call DMGAN with weight adaption waDMGAN and summarize the learning details of waDMGAN in Algorithm 2.

4. Experiments

We implement our experiments on various datasets, including MNIST [42], SVHN [43] and CIFAR-10 [44]. In the following sections, we first describe experimental details and then show results on different datasets.

Generator	Discriminator
Input 100-D random noise	Input 64 × 32 × 32 × 3 images
5c-2s-512o UpConv, BN, LReLU	5c-2s-64o Conv, BN, LReLU
5c-2s-256o UpConv, BN, LReLU	5c-2s-128o Conv, BN, LReLU
5c-2s-128o UpConv, BN, LReLU	5c-2s-256o Conv, BN, LReLU
5c-2s-64o UpConv, BN, LReLU	5c-2s-512o Conv, BN, LReLU
5c-2s-3o UpConv, BN, LReLU	500o FC
Tanh	
Output 64 × 32 × 32 × 3	Output 500-D feature vector

Table 1: The structures of the generator and discriminator. "5c-2s-512o" denotes a 5 × 5 kernel with stride 2 and 512 outputs. "UpConv" stands for a fractionally-strided convolution layer, "FC" is the abbreviation of a fully connected layer. "BN" and "LReLU" imply batch normalization and leaky ReLU, respectively. The dimensionality of the output vector of discriminator is set to 500.

Experimental details and hyper-parameters Before presenting experiments, we briefly introduce some experimental details. We use the TensorFlow [45] library (version 1.3.0) to implement our experiments. Meanwhile, a speed-up computing technique by TitanX GPU is exploited. The exact architectures of the discriminator and the generator are typically implemented as MBGAN, which are described in Table 1. Besides, we constraint our model with Lipschitz restriction, which is realized by adding weight clipping in the discriminator. For the clipping threshold, we experimentally set it to [-0.1, 0.1]. In our experiments, the model requires techniques such as batch normalization [46] and leaky ReLU [47] to achieve convergence. We use Adam optimization [48] for training and set the learning rate to 0.0002, momentum parameters α to 0.5, β_1 and β_2 to 0.9 and 0.99, respectively. All models used in the following experiments are trained with mini-batch size of 64. Without a special explanation, these hyper-parameters are fixed for all the visualization experiments.

Datasets We implement our experiments on various datasets, including MNIST [42], SVHN [43] and CIFAR-10 [44]. We conduct experiments on these datasets for the following reasons. Firstly, they are all labeled databases, which meet the requirements of our algorithm. It is suitable for us to learn faithful data-dependent centers during training due to the little difference in the number of samples of different categories in these three datasets. Secondly, many generative adversarial models conduct their experiments on these datasets, and the complexity of samples in MNIST, SVHN, and CIFAR-10 is gradually increasing. Experiments on them can illustrate that our model not only performs very well on simple images but also can deal with complex datasets. Figure 2 shows some examples of these datasets, and details of them are described as follows:

- MNIST [42] contains 60,000 training images and 10,000 test images of digits 0 to 9, and the images in MNIST are grayscale with size 28.

- SVHN [43] is a real-world dataset that is obtained from house numbers in Google Street View pictures. As a dataset composed of digital images, SVHN contains RGB samples with more complication than MNIST.

- CIFAR-10 [44] contains 32 × 32 RGB images belong to 10 different classes, with 5000 training images and 1000 test images per class. Both training images and test images are utilized to train our model.

Evaluation metrics Quantitatively estimating GAN models remains a challenging task because likelihood cannot be efficiently evaluated. An

(a) MNIST (b) SVHN (c) CIFAR-10

Figure 2: Some real samples of MNIST, SVHN and CIFAR-10.

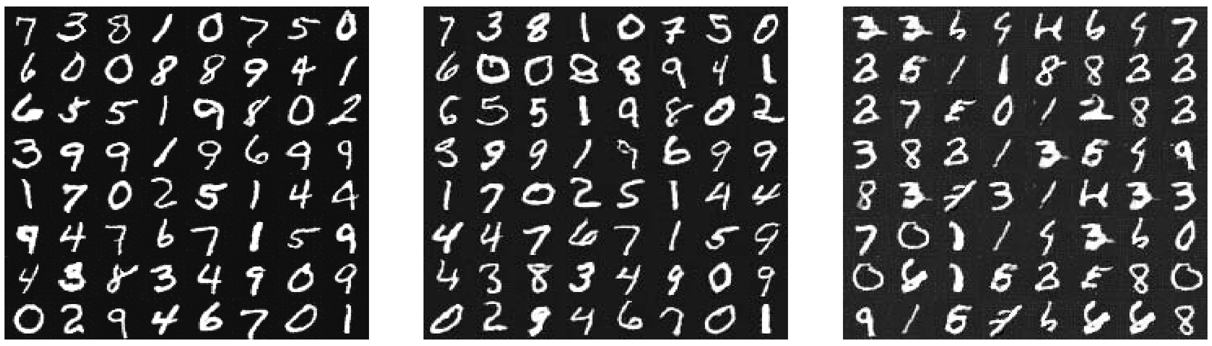
intuitive metric can be obtained by having human annotators judge the visual quality of samples [2]. However, using human annotators always suffers from a problem that the metric varies depending on the setup of the task and the motivation of annotators.

As a substitution to human annotators, *Inception Score* (IS) [3] is proposed to evaluate samples automatically. In particular, generated samples are fed into the Inception model [49] to get a conditional distribution. IS reveals the exponential result of the entropy of samples, which corresponds to a higher value when generated samples are of high quality and diversity, and a lower value if the quality of generated images is poor.

As another evaluating criterion, *Frechet Inception Distance* (FID) [50] measures the difference between real samples and generated samples by Frechet distance. It should be noted that if the distributions of generated images and real images are more similar, the value of the corresponding FID is smaller. Both IS and FID are well-performing approaches to measure the performance of GANs and correlate well with human judgment. We use both of them to quantify the diversity and quality of generated samples in our experiments.

4.1. Experiments on MNIST

In this experiment, we use the network architectures listed in Table 1 but reset the output dimensionality of the generator and the input dimensionality of the discriminator to $64 \times 28 \times 28 \times 1$. For a fair comparison, all GAN models use the same network architectures. In regular DMGAN, we set the update ratio of centers to 1.0 and the hyper-parameter λ in Eq. 5 to 1.2. In MBGAN, there are two additional hyper-parameters α and K, where α is used to control the magnitude of the data-dependent margin and K denotes the dimensionality of the output features. According to the descriptions in MBGAN, we set α to 200 and K to 500.



(a) DCGAN (b) regular DMGAN (c) MBGAN

Figure 3: The generated samples on MNIST. By comparing subfigure (a) and (b), we find that our regular DMGAN can synthesize samples with comparable quality over DCGAN. Besides, our model is capable to generate samples with clearer backgrounds than MBGAN by comparing subfigure (b) and (c).

We compare our regular DMGAN with popular DCGAN and MBGAN, which are also from the perspective of deep metric learning. From the results shown in Figure 3, we can see that DMGAN generates real-like samples similar to DCGAN, although it is trained without implicit calculations of probability. Meanwhile, the generated images are more realistic than images synthesized by MBGAN. Indeed, due to the lack of restrictions on the distribution of features of the whole samples, MBGAN just generates images with blurred backgrounds. And we own the superiority of DMGAN to implicitly constraint the distribution of data by utilizing data-dependent centers.

4.2. Experiments on SVHN

In our experiments, we use the training set of SVHN, which consists of 73,527 RGB digits with all images having been resized to a fixed resolution 32×32. We use the same architectures as MBGAN shown in Table 1, and the metric criteria FID is utilized to evaluate the quality of synthesized samples.

In our model, we introduce data-dependent centers to the objective of the discriminator to extract representative features, which are essential for generative models based on deep metric learning. To demonstrate data-dependent centers can help DMGAN to generate more realistic images, we first conduct experiments to investigate the performance of our model related to the update ratio of centers. For ease of exposition, DMGAN with centers' update ratio $\gamma = 0$ is called DMGAN-f, in which centers utilized in the discriminative objective are fixed as initial vectors during training. Then we compare DMGAN-f with regular DMGAN models where centers are updated in the training process as in Algorithm 1.

λ	1.0	1.4	1.8	2.2	2.6	3.0
Regular DMGAN	191.25	64.76	46.79	60.87	52.76	51.28

Table 2: The FID of regular DMGAN with different values for hyper-parameter λ. The lower the score corresponding to the better the model.

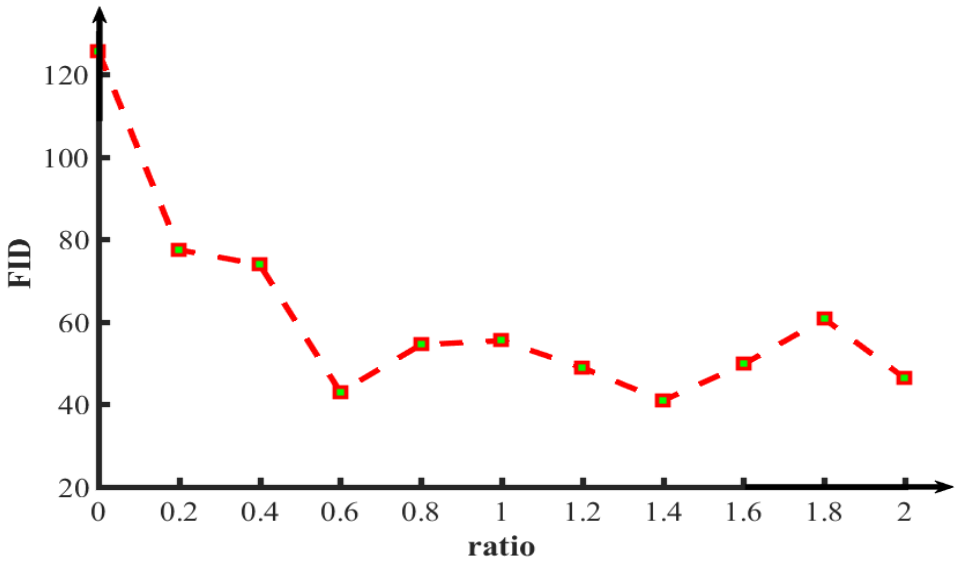


Figure 4: The curve shows the FID scores of regular DMGAN for different update ratio of centers tested on SVHN. We find that DMGAN with data-dependent centers ($ratio > 0$) could achieve superior performance compared with models with fixed centers ($ratio = 0$). This demonstrates that utilizing data-dependent centers significantly enhances the performance of DMGAN.

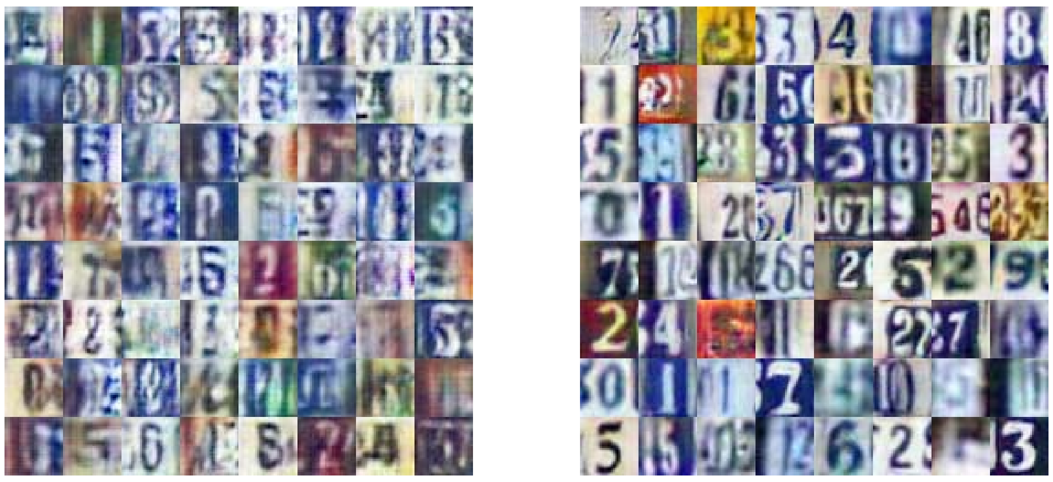
The hyper-parameter λ is introduced to balance the identity preserving loss and discriminative loss in the objective of the discriminator. Specifically, the identity preserving loss can make the discriminator learn more robust features, while the discriminative loss is used to increase the distance between real samples and generated ones. Therefore, it is very important to select an appropriate λ before investigating the impact of data-dependent centers. To select the most suitable λ, we fix the update ratio γ of centers to 0.5 and vary λ from 1.0 to 3.0 to learn different models. The FID of these models on SVHN listed in Table 2 shows that the quality of generated samples is the best when λ is selected to be 1.8.

After fixing λ, we vary the update ratio of centers from 0 to 3.0 to explore the effect of different update ratios on the performance of our model.

However, DMGAN encounters mode collapse problem when γ is larger than 2.0, which may due to the instability caused by too fast changes of centers during training. Figure 4 shows the results of FID on different models with the update ratio of centers from 0 to 2.0. From these results, we can make several observations:

- DMGAN with data-dependent centers could achieve superior performance compared with models with fixed centers. This result shows that utilizing data-dependent centers significantly enhances the performance of DMGAN.
- The experimental results show that the performance of our model remains stable across a wide range of γ, which illustrates the robustness of DMGAN.

In addition, we give a comparison of the synthesized samples showed in Figure 5 to demonstrate the advantage of data-dependent centers in DMGAN. Specifically, DMGAN-f produces poorer images, while DMGAN with centers updated improves the quality of generated images.



(a) DMGAN-f

(b) DMGAN

Figure 5: Subfigure (a) shows samples synthesized by DMGAN with fixed centers (DMGAN-f) and subfigure (b) exhibts samples generated by DMGAN with data-dependent centers. The results of DMGAN-f and DMGAN on SVHN illustrate the advantage of data-dependent centers in DMGAN.

4.3. Experiments on CIFAR-10

Experimental results in section 4.1 demonstrate that the generating tasks can be achieved by GAN models from the perspective of deep metric learning. Meanwhile, we also validate the importance of data-dependent centers in DMGAN via experiments on SVHN in section 4.2. In this part, we first present a comparison between our proposed regular DMGAN and waDMGAN to verify the crucial role of the strategy of weight adaption in DMGAN. At the same time, we compare our model with state-of-the-art GAN models and illustrate that waDMGAN can generate samples with similar quality to other models on CIFAR-10 dataset.

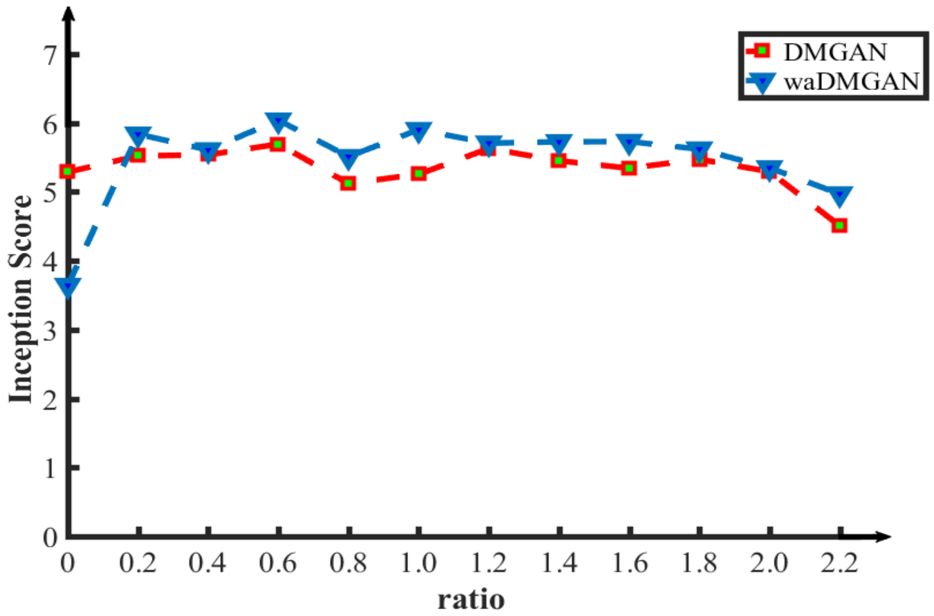
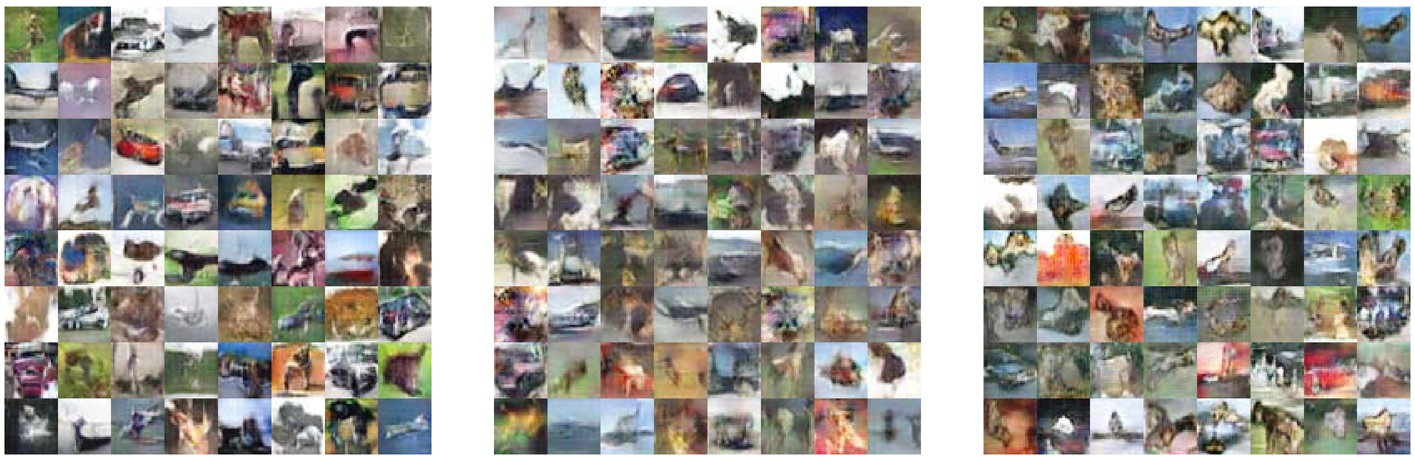


Figure 6: The comparison of IS scores of samples synthesized by waDMGAN and regular DMGAN with different update ratios of centers. According to the results, we find that the scores of waDMGAN are regularly higher than that of DMGAN except for the case that the centers are fixed, which verifies that equipping DMGAN with the strategy of weight adaption improve the performance of our model.

4.3.1. DMGAN vs. waDMGAN

In our experiments, both training images and test images are utilized to train DMGAN models. The architectures of the discriminator and generator are the same as MBGAN with weight clipping as shown in Table 1. To present the influence of the strategy of weight adaption on DMGAN, we compare regular DMGAN with waDMGAN. Before the comparison, we first select appropriate hyper-parameter λ following the same procedure as in experiments on SVHN in section 4.2 and obtain the optimal λ in Equation 8 at 1.2. Differently, we use IS to evaluate the quality of synthesized samples in this section.

In order to highlight the advantages of adaptive weights over fixed weights, we compare the quality of generated samples of waDMGAN and regular DMGAN with different update ratios of centers in our experiment. Quantitative results are shown in Figure 6. According to the results, equipping DMGAN with the strategy of weight adaption increases the performance within the whole range of update ratios of centers, especially in the case of update ratio at 1.0, which shows that waDMGAN has the desired effect of improving the quality of generated samples. Indeed, waDMGAN's superiority is that it pays more attention to poor synthesized samples by automatically adjusting updated gradients. To further demonstrate the advantages of waDMGAN, we visualize the results of waDMGAN and DMGAN in Figure 7 (a) and (b). According to the results, images generated by waDMGAN are clearer and containing more details than images generated by DMGAN.



(a) waDMGAN (b) DMGAN (c) MBGAN

Figure 7: The visualization results of waDMGAN, DMGAN, and MBGAN on CIFAR-10. By comparing results of subfigures (a) and (b), we find that synthesized samples by waDMGAN contain more clear background details than those of regular DMGAN. Compared with results of waDMGAN and DMGAN, samples generated of MBGAN in subfigure (c) suffer from a serious lack of details.

4.3.2. DMGAN vs. other GAN models

To further verify the effectiveness of our proposed approach, we conduct experiments to compare our model with state-of-the-art GAN models. The quantitative results of different models are shown in Table 3. Compared with two popular models, DCGAN and WGAN, waDMGAN achieves the IS of 6.04 that outperforms 5.88 and 5.92 gained by WGAN and DCGAN, respectively. These results illustrate that GAN models, which is from the perspective of deep metric learning, can generate similar or better samples than probability-based GANs. In addition, compared with metric-based models such as MBGAN and KMGAN, both DMGAN and waDMGAN outperform them with a large margin. These results demonstrate the effectiveness of our models. However, the results of our models are lower than WGAN-GP due to lack of gradient penalty, which motivates us to introduce gradient penalty into our model in further work.

On the other hand, we record the time of each iteration of models during the learning process and show them in Table 3. According to the results shown in Table 3, we find that our models are slower than DCGAN due to the need of metric learning. Same as metric-based generative adversarial models, our models achieve comparable speed with MBGAN, although our models need to calculate data-dependent centers and adaptive weights additionally. Meanwhile, our models have an apparent advantage over KM-GAN and SAGAN, which needs complex self-attention calculations. This result illustrates that our models are more relaxed to be optimized than other models.

Model	Inception Score	Time (ms/per iteration)
DCGAN	5.92 ± 0.17	237
WGAN [51]	5.88 ± 0.07	397
MBGAN	5.07 ± 0.06	274
MLGAN-clipping [30]	5.23 ± 290	-
WGAN-GP	6.46 ± 0.03	413
KM-GAN	5.61 ± 0.09	650
SAGAN	5.72 ± 0.06	3563
DMGAN	**5.69 ± 0.08**	263
waDMGAN	**6.04 ± 0.04**	259

Table 3: Inception Scores and the time of each iteration on CIFAR-10. Among unconditional models, our models achieve state-of-the-art performance. With the addition of condition information, waDMGAN outperforms all other supervised algorithms except WGAN-GP. Besides, the time of per iteration of models illustrate that our models are more relaxed to be optimized than other state-of-the-art models.

Finally, Figure 7 shows the images generated by DMGAN, waDMGAN and MBGAN on CIFAR-10 dataset. As we can see from the figure, samples synthesized by waDMGAN have more details and clearer backgrounds than those of regular DMGAN. On the other hand, details of images generated by MBGAN degrade more heavily. Through convincing visualization results and quantitative evaluations, we demonstrate the performance of our method.

5. Conclusion and future work

In this paper, we proposed a novel GAN model, referred to DMGAN, from the perspective of deep metric learning. Instead of outputting probability, the discriminator in DMGAN is conducted as a feature extractor whose outputs are multi-dimensional features. In addition, identity preserving loss and discriminative loss are introduced to constrain the discriminator for representative features. Moreover, we introduce data-dependent centers in the identity preserving loss to learn robust discriminative features. Meanwhile, a strategy of weight adaption is proposed to make the discriminator pay more attention to poor samples and improve the quality of generated images. On the other hand, the generator synthesizes realistic samples by minimizing the distance between real samples and generated samples. Extensive experiments on several datasets demonstrate the effectiveness of our proposed approach.

Unfortunately, our proposed model is conditioned on labels of samples, and the acquisition of class labels is expensive and time-consuming in practice. Therefore, we will improve our model to fit more unlabeled datasets in future works.

Acknowledgments

This work was supported by the National Key R & D Program of China (2018YFC0809800), the National Natural Science Foundation of China (61801325), and the Huawei Innovation Research Program (HO2018085347). The authors would like to thank Dong Wang for his helpful advices.

References

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

[2] E. L. Denton, S. Chintala, R. Fergus, et al., Deep generative image models using a laplacian pyramid of adversarial networks, in: Advances in neural information processing systems, 2015, pp. 1486–1494.

[3] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training gans, in: Advances in neural information processing systems, 2016, pp. 2234–2242.

[4] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al., Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4681–4690.

[5] 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. 1125–1134.

[6] 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. 2223–2232.

[7] Z. Zhang, Y. Xie, L. Yang, Photographic text-to-image synthesis with a hierarchically-nested adversarial network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6199–6208.

[8] Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, A. Elgammal, A generative adversarial approach for zero-shot learning from noisy texts, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 1004–1013.

[9] X. Peng, Z. Tang, F. Yang, R. S. Feris, D. Metaxas, Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2226–2234.

[10] A. Jahanian, L. Chai, P. Isola, On the"steerability" of generative adversarial networks, arXiv preprint arXiv:1907.07171 (2019).

[11] R. Salakhutdinov, H. Larochelle, Efficient learning of deep boltzmann machines, in: Proceedings of the thirteenth international conference on artificial intelligence and statistics, 2010, pp. 693–700.

[12] Y. Bengio, E. Laufer, G. Alain, J. Yosinski, Deep generative stochastic networks trainable by backprop, in: International Conference on Machine Learning, 2014, pp. 226–234.

[13] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv:1511.06434 (2015).

[14] T. Karras, T. Aila, S. Laine, J. Lehtinen, Progressive growing of gans for improved quality, stability, and variation, arXiv preprint arXiv:1710.10196 (2017).

[15] T. Karras, S. Laine, T. Aila, A style-based generator architecture for generative adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4401–4410.

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

[17] K. Kurach, M. Lučić, X. Zhai, M. Michalski, S. Gelly, A large-scale study on regularization and normalization in gans, in: International Conference on Machine Learning, 2019, pp. 3581–3590.

[18] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein gan, arXiv preprint arXiv:1701.07875 (2017).

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

[20] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, S. P. Smolley, Least squares generative adversarial networks, in: Computer Vision (ICCV), 2017 IEEE International Conference on, IEEE, 2017, pp. 2813–2821.

[21] A. Brock, J. Donahue, K. Simonyan, Large scale gan training for high fidelity natural image synthesis, arXiv preprint arXiv:1809.11096 (2018).

[22] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: Closing the gap to human-level performance in face verification, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 1701–1708.

[23] Y. Sun, X. Wang, X. Tang, Deep learning face representation from predicting 10,000 classes, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 1891–1898.

[24] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[26] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 2, IEEE, 2006, pp. 1735–1742.

[27] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 815–823.

[28] Y. Wen, K. Zhang, Z. Li, Y. Qiao, A discriminative feature learning approach for deep face recognition, in: European conference on computer vision, Springer, 2016, pp. 499–515.

[29] G. Dai, J. Xie, Y. Fang, Metric-based generative adversarial network, in: Proceedings of the 2017 ACM on Multimedia Conference, ACM, 2017, pp. 672–680.

[30] Z.-Y. Dou, Metric learning-based generative adversarial network, arXiv preprint arXiv:1711.02792 (2017).

[31] G. Li, G. Kou, Y. Peng, A group decision making model for integrating heterogeneous information, IEEE Transactions on Systems, Man, and Cybernetics: Systems 48 (2016) 982–992.

[32] H. Zhang, G. Kou, Y. Peng, Soft consensus cost models for group decision making and economic interpretations, European Journal of Operational Research 277 (2019) 964–980.

[33] G. Kou, Y. Peng, G. Wang, Evaluation of clustering algorithms for financial risk analysis using mcdm methods, Information Sciences 275 (2014) 1–12.

[34] O. Rippel, M. Paluri, P. Dollar, L. Bourdev, Metric learning with adaptive density discrimination, arXiv preprint arXiv:1511.05939 (2015).

[35] C. Wang, C. Xu, X. Yao, D. Tao, Evolutionary generative adversarial networks, IEEE Transactions on Evolutionary Computation (2019).

[36] Y. Hong, U. Hwang, J. Yoo, S. Yoon, How generative adversarial networks and their variants work: An overview, ACM Computing Surveys (CSUR) 52 (2019) 10.

[37] Y.-J. Cao, L.-L. Jia, Y.-X. Chen, N. Lin, C. Yang, B. Zhang, Z. Liu, X.-X. Li, H.-H. Dai, Recent advances of generative adversarial networks in computer vision, IEEE Access 7 (2019) 14985–15006.

[38] J. T. Springenberg, Unsupervised and semi-supervised learning with categorical generative adversarial networks, stat 1050 (2016) 30.

[39] C. Wang, Z. Chen, K. Shang, H. Wu, Label-removed generative adversarial networks incorporating with k-means, Neurocomputing 361 (2019) 126–136.

[40] M. Mirza, S. Osindero, Conditional generative adversarial nets, arXiv preprint arXiv:1411.1784 (2014).

[41] R. Wang, A. Cully, H. J. Chang, Y. Demiris, Magan: Margin adaptation for generative adversarial networks, arXiv preprint arXiv:1704.03817 (2017).

[42] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (1998) 2278–2324.

[43] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, Reading digits in natural images with unsupervised feature learning (2011).

[44] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, Technical Report, Citeseer, 2009.

[45] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., Tensorflow: Large-scale machine learning on heterogeneous distributed systems, arXiv preprint arXiv:1603.04467 (2016).

[46] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167 (2015).

[47] A. L. Maas, A. Y. Hannun, A. Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in: Proc. icml, volume 30, 2013, p. 3.

[48] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[49] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.

[50] 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, in: Advances in Neural Information Processing Systems, 2017, pp. 6626–6637.

[51] Y. Li, K. Swersky, R. Zemel, Generative moment matching networks, in: International Conference on Machine Learning, 2015, pp. 1718–1727.