MGGAN: Solving Mode Collapse Using Manifold-Guided Training

Duhyeon Bang[1,2*]
[1]SK Telecom
SK T Tower, 65 Eulji-ro, Jung-gu,
Seoul, South Korea
duhyeonbang@sktbrain.com

Hyunjung Shim[2]
[2]Yonsei university
School of Integrated Technology,
Yonsei University, 85, Songdogwakak-ro,
Yeonsu-gu, Incheon, South Korea
kateshim@yonsei.ac.kr

Abstract

Mode collapse is a critical problem in training generative adversarial networks. To alleviate mode collapse, several recent studies have introduced new objective functions, network architectures, or alternative training schemes. However, their achievement is often the result of sacrificing the image quality. In this paper, we propose a new algorithm, namely, the manifold-guided generative adversarial network (MGGAN), which leverages a guidance network on existing GAN architecture to induce the generator to learn the overall modes of a data distribution. The guidance network transforms an image into a learned manifold space, which is effective in representing the coverage of the overall modes. The characteristics of this guidance network helps penalize mode imbalance. Results of the experimental comparisons using various baseline GANs showed that MGGAN can be easily extended to existing GANs and resolve mode collapse without losing the image quality. Moreover, we extend the idea of manifold-guided GAN training to increase the original diversity of a data distribution. From the experiment, we confirmed that a GAN model guided by a joint manifold can sample data distribution with greater diversity. Results of the experimental analysis confirmed that MGGAN is an effective and efficient tool for improving the diversity of GANs.

1. Introduction

Generative adversarial networks (GANs) [11] are a family of generative models that implicitly learn the data distribution in an unsupervised manner. This is accomplished by learning to generate new data samples instead of explicitly constructing a density function. Since GANs do not rely on strong statistical assumptions on distributions, there are no performance limitations on modeling complex manifolds of a data distribution. Owing to this attractive nature, GANs have been successful in image-generation tasks.

Despite their promising achievements, GANs are notoriously difficult to train because of the training instability and sensitivity to hyperparameters. Training instability causes two problems: poor image quality and lack of image diversity. Existing studies [5, 10] have shown that these two issues are in a trade-off relationship with each other. Thus, the goal of existing GAN models is mainly to focus on improving either the image quality or the image diversity. In this study, our primary interest is to improve image diversity without sacrificing image quality.

The lack of image diversity in GAN training is also known as mode collapse, in which P_{model} captures a single or a few major modes of P_{data} while ignoring many minor modes. To address this problem, we propose a novel algorithm, namely, the manifold-guided generative adversarial network (MGGAN), which integrates a newly proposed guidance network to the existing GAN architecture. Note that the standard GAN consists of a discriminator network and a generator network. The discriminator aims to distinguish the fake images produced by the generator from the real images. Meanwhile, the generator aims to fool the discriminator by generating fake images that look as realistic as possible. On the basis of the standard GAN architecture, we leverage the guidance network, which induces the generator to learn the overall modes of P_{data}. The goal of the guidance network is to induce the generator such that P_{model} matches P_{data} in the learned manifold space. To this end, the guidance network consists of an encoder for manifold mapping and a discriminator for evaluating the dissimilarity between the distributions of P_{data} and P_{model} in the manifold space. In this way, we enforce that the characteristics of the learned manifold space be reflected in the generator training. It is important to note that the encoder of the guidance network should be predetermined; encoder training is independent of the discriminator of the guidance network. The reason is that the learned manifold should

*This work was done during his doctoral course in Yonsei University.

consistently provide a meaningful representation that is coherent with the goal of the guidance network. Then, all parameters from the discriminator of the guidance network are trained jointly with the generator and discriminator networks.

The guidance network plays a role in penalizing the mode imbalance in GAN training. Therefore, the manifold space of the guidance network should represent the mode coverage of a data distribution well. In this regard, we employ an encoder layer of a pretrained autoencoder to define the manifold mapping of the guidance network. Since the autoencoder can be interpreted as minimizing a forward Kullback–Leibler (KL) divergence [34], the manifold learned by the encoder is effective in representing the mode coverage of P_{data} [30]. Hence, the feedback from this encoder manifold regularizes the mode imbalance. By learning two distributions (i.e., data distribution and regularized data distribution by manifold), the generator effectively covers various modes. Concurrently, we keep the objective of the original discriminator as modeling each mode correctly, thus our model does not sacrifice the image quality; for example, a non-saturated GAN model tends to follow a reverse Kullback-Leibler (KL) divergence.

Finally, on the basis of extensive evaluations on various benchmark datasets, we show that the proposed algorithm is effective in resolving mode collapse without losing the image quality. Moreover, our manifold-guided training can be adopted to various baseline GAN models and consistently improve their performance.

2. Related Works

Regularizing the discriminator. To address mode collapse, Arjovsky et al. [3] suggested the Wasserstein distance for GAN metrics, namely, WGAN. Although this new metric is effective in alleviating mode collapse, their weight clipping implementation unfortunately causes a pathological behavior [13]. D2GAN [27] employed two antithetical discriminators: one minimizes the forward KL divergence and the other minimizes the reverse KL divergence. Since the generator aims to fool both discriminators simultaneously, it is effective to escape from mode collapse. However, their scheme increases training instability by oscillating between two antithetical discriminators. Our MGGAN is similar to D2GAN in that our guidance network holds the forward KL properties inherited by the pretrained autoencoder. However, training the MGGAN is quite stable because two discriminators are formulated with the same divergence.

Unrolled GAN [25] introduced a surrogate objective function to better simulate a discriminator response and achieved the robust performance against mode collapse. However, it is not clear whether their achievement sacrifices the visual quality because of no real data experiments. Heavy computational overheads due to k-step discriminator updates is also a well-known drawback of Unrolled GAN. DRAGAN [16] proposed a gradient penalty (GP) term to regularize large gradients, which is also effective in mitigating mode collapse. LSGAN [23] replaced the sigmoid cross-entropy loss term in standard GAN with a least squares loss term, equivalent to Pearson $\tilde{\chi}^2$ divergence. While such a replacement reduces the possibility of mode collapse, neither DRAGAN nor LSGAN has shown a significant achievement for improving diverse image generation with real datasets.

Later, SNGAN [26] introduced advanced normalization layers and achieved state-of-the-art image quality on various benchmark datasets, better than GP-based methods [16, 31]. Mescheder et al. [24] analyzed the convergence of existing regularization strategies on a simple yet prototypical example and showed that the unregularized gradient-based GAN optimization does not always converge to the Nash equilibrium. Then, they chose the simplified version of zero-centered gradient penalties that leads to local convergence. Progressively growing GAN [14] suggested a new training method that educates the generator and the discriminator progressively from low resolution to high resolution. Although these studies have successfully stabilized GAN training, they mostly focus on generating high-quality and high-resolution samples rather than on resolving the mode collapse problem. PacGAN [21] provided theoretical analysis of the benefits of showing multiple samples simultaneously, by packing to the discriminator and successfully mitigated the mode collapse problem. The main difference between PacGAN and our MGGAN is that PacGAN utilizes the relationships among multiple samples (i.e., data space), while MGGAN utilizes the feature distribution formed by the predefined encoder (i.e., manifold space) to improve GAN training.

Learning to map between the latent and the data domain. Several recent studies have proposed to learn a mapping function from P_{data} to P_z, namely, an inference mapping. ALI [9] and BiGAN [8] suggested a discriminator for joint distribution matching, which learns a relationship between data and latent distribution. They used the inference mapping for sample reconstruction and conditional generation, but do not improve either the image quality or the image diversity.

MDGAN [6], α-GAN [30] and VEEGAN [33] utilized a reconstruction loss as an additional constraint to the inference mapping. Although the reconstruction loss is effective, its training suffers from instability because its unit mismatches that of an adversarial loss; reconstruction loss is a distance measure, whereas adversarial loss is a divergence measure. MDGAN separated training into a mode regularization step (encouraging the mode coverage) and a diffusion step (leading the high quality generation). Follow-

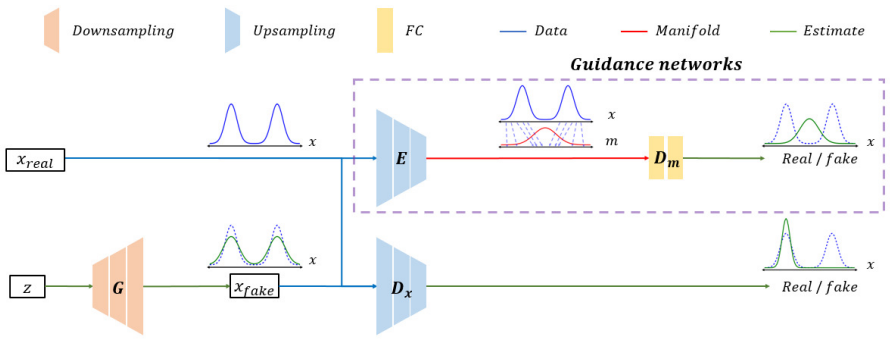


Figure 1. Structure of the proposed model. x_{real} and x_{fake} are samples of P_{data} and P_{model}, respectively; z is a latent vector; E, G, and D are an encoder, a generator, and a discriminator network, respectively. The subscript of D means input sample space. The guidance network consists of E and D_m, where m implies manifold space. FC means fully connected layer.

ing the principle of the variational inference [15], α-GAN adopted the adversarial loss to match the latent distribution; α-GAN replaces the autoencoder part of MDGAN by the variational autoencoder. Unlike MDGAN and $\alpha-$ GAN , VEEGAN [18] applies the reconstruction loss in the latent domain rather than in the data domain to mitigate the image-quality degradation (i.e., image blur). Although both MDGAN and VEEGAN are effective in handling mode collapse, their results often sacrifice the image quality. ALICE [19] aimed to improve the training instability of GANs by adopting a conditional entropy, formulated as the cycle consistency [36]. In summary, the aforementioned techniques improve joint distribution matching toward either reconstructing samples or resolving mode collapse. However, they commonly suffer from the discrepancy between the theoretical optimum and the practical convergence [19]. This results in either image blurs during the generation or inaccurate inference mapping.

3. Manifold-guided GAN

Our goal is to generate diverse samples, i.e., solving mode collapse, without sacrificing the image quality. To achieve this, we propose a new algorithm that induces a generator to learn the entire modes of P_{data} as well as produces realistic samples. Specifically, we introduce a guidance network, which leads the generator to produce samples reflecting the specific manifold characteristics. The proposed model is constructed by combining the standard GAN, which consists of a generator, G, and a discriminator, D_x, with this guidance network, and this is shown in Fig. 1.

For the sake of distinguishing between the true and the estimated probability distribution, we mark with a hat the estimated variables; in our study, since the encoder maps the true probability distribution to the manifold, $E(x \sim P_{data})$ is mapped onto P_m and $E(x \sim P_{model})$ is mapped onto $P_{\widehat{m}}$, where m represents the manifold space.

Our guidance network aims to reduce the divergence between the projections of P_{data} and P_{model} on the manifold space. The guidance network is composed of an encoder, E, and a discriminator, D. The encoder maps P_{data} and P_{model} to the manifold space. The discriminator for the guidance network, D_m, distinguishes the encoded P_{model} from the encoded P_{data}, i.e., P_m and $P_{\widehat{m}}$, respectively. The following equations show the objective function for our MGGAN, where the guidance network is implemented with the non-saturated GAN:

$$\begin{aligned}&\min_{D_x,D_m} \mathbb{E}_{x\sim P_{data}}\left[\log\left(D_x\left(x\right)\right) + \log\left(D_m\left(E\left(x\right)\right)\right)\right] + \\ &\qquad \mathbb{E}_{z\sim P_z}\left[\log\left(1-D_x\left(G\left(z\right)\right)\right) + \log\left(1-D_m\left(E\left(G\left(z\right)\right)\right)\right)\right], \\ &\min_{G} -\mathbb{E}_{z\sim P_z}\left[\,\alpha\log\left(D_x\left(G\left(z\right)\right)\right) + (1-\alpha)\log\left(D_m\left(E\left(G\left(z\right)\right)\right)\right)\right].\end{aligned} \tag{1}$$

As described in the above equations, the two discriminators, D_x and D_m, do not explicitly affect each other, although both of them affect the generator. From the bottom equation, the generator attempts to meet two goals simultaneously: the first is to minimize the dissimilarity between P_{data} and P_{model}, equivalent to that of a non-saturated GAN, and the second is to minimize the dissimilarity between their mapped distributions onto a manifold space. It is worth noting that our two discriminators concurrently affect the generator training, and, thus, the two discriminators are implicitly influenced by each other through the generator. Also, the encoder of a guidance network is designed to derive the most representative manifold of P_{data} where the coverage of all modes of P_{data} is captured. As a result, the guidance network can induce the generator training in that the generator is capable of producing diverse samples because P_{model} tends to encapsulate all modes of P_{data} influenced by the characteristics of the encoder of the guidance network.

3.1. Characteristics of the guidance network

As shown in Fig. 1, the guidance network consists of an encoder and a discriminator. To solve the mode collapse, we designed the encoder E such that the output distribution

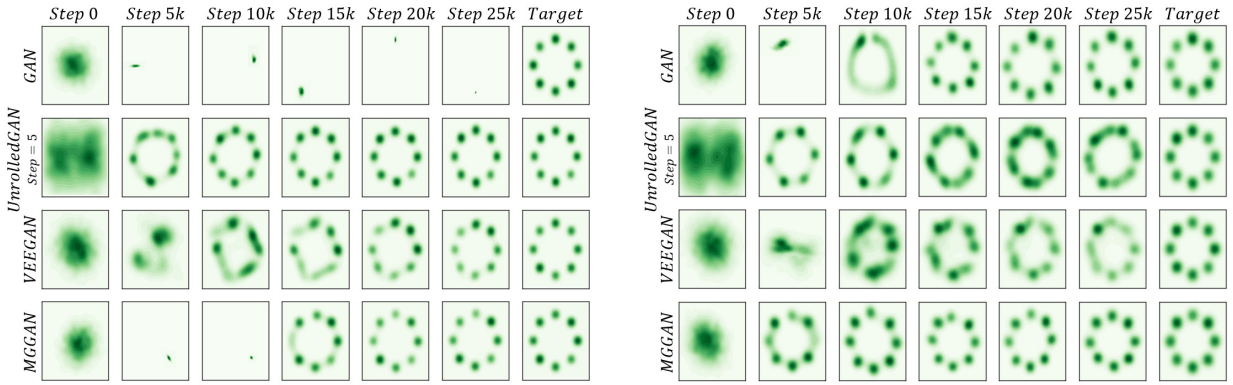


Figure 2. Mode collapse test learning a mixture of eight Gaussian spreads in a circle with standard deviations of 0.01 (left) and 0.35 (right).

of encoder P_m can cover the overall range of all modes of P_{data}; all modes of P_{data} should be reflected in constructing P_m. To meet these criteria, we employ the encoder of a pretrained autoencoder.

The autoencoder first learns the latent representation of a dataset using the encoder and then reconstructs it by decoding it from the latent. Because the autoencoder network is trained to minimize the reconstruction errors (i.e., L1, L2, or a cross-entropy loss between the input and its reconstruction), the autoencoder could observe all modes of a true data distribution. In fact, this is closely related to the forward KL divergence property [34]. Suppose we model the reconstruction errors by the cross-entropy loss (i.e., H(P_{data},P_{model}), where H is the entropy); the autoencoder can be interpreted as following a forward KL divergence between P_{data} and P_{model} (i.e., KL(P_{data} $||P_{model}$) = H(P_{data},P_{model}) - H(P_{data})). This forward KL property ensures that the manifold derived by the encoder can account for all modes of a data distribution regardless of the reconstruction quality; it tends to average all P_{data} modes [12], as shown by the red graph in Fig. 1. Although using the autoencoder alone causes quality degradation of the image generation (e.g., image blurs), this is advantageous to achieve the goal of the guidance network, which induces the generator to learn a true distribution without missing modes. Owing to its being a useful property of the autoencoder, the encoder serves as an effective manifold space such that P_m can reflect the overall modes of P_{data}. Specially, we pretrain and fix the parameters of the autoencoder using a real dataset. In this way, it is possible to keep the manifold property of the encoder and reduce uncertainty of the inference.

Since the manifold space is a topological space, a general distance measure is not suitable for the dissimilarity measure between two samples, one from P_m and the other from $P_{\widehat{m}}$ [20]. To measure the dissimilarity between P_m and $P_{\widehat{m}}$, the discriminator of the guidance network D_m learns to separate two distributions in the manifold space according to the adversarial learning. To construct D_m, we use a structure and a divergence identical to those of the discriminator of the standard GAN. By controlling the weights for two loss terms, we can strengthen or weaken the effect of the guidance network. That is, when the weight from the feedback of the guidance network is increased, MGGAN can achieve greater diversity. The detailed analysis is provided in the quantitative evaluation (Table 1). Because two terms consistently use adversarial loss, free from unit mismatch, we still achieve stable training under various choices of weights.

3.2. Relationship between D_m and D_x

To better understand the role of the guidance network, we focus on the relationship between the discriminator of the guidance network D_m and the original discriminator D_x. Interpreting it in depth, we argue that D_m and D_x have a complementary relationship. Suppose the data distribution is a mixture of two Gaussian modes (blue line), as depicted in the graph in Fig. 1. Then, each discriminator leads the generator to reproduce the target distribution, which is represented by the green line in the figure. In ordinary GAN models, D_x leads the generator to choose either of the two modes, and not both; thus, the generator influenced only by D_x is susceptible to mode collapse. This is because D_x follows the properties of reverse KL divergence [2]. On the contrary, D_m induces the generator to cover the overall modes, such as the average distribution of the two modes. It is because D_m follows the properties of forward KL divergence inherited by the latent space of the autoencoder. We call this phenomenon mode averaging because the generator tends to learn the average-like distribution. In this study, we intend to utilize the complementary properties of D_x and D_m. As MGGAN utilizes two discriminators, the mode collapse is mitigated by mode averaging, while the mode averaging is improved by mode collapse. As a result, we expect to improve the sample diversity while preserving the sample quality.

3.3. Comparison to ALI/BiGAN, MDGAN, and VEEGAN

Several recent studies have stated that traditional GANs imposing unidirectional mapping (i.e., generation mapping)

are insufficient for solving the lack of diversity in GAN training. To address this problem, they suggest utilizing an inference mapping to regularize the generator training [9, 8, 6, 33]. The network architecture used in these studies is similar to the proposed model in that they also exploited an encoder architecture to map P_{data} into a low-dimensional manifold space. However, their encoders are designed to map P_{data} into P_z (i.e., inference mapping) to reproduce P_{data} through the generator. Meanwhile, we intend to use the encoder only for regularizing P_{data} to follow the forward KL properties.

More specifically, previous studies utilizing inference mapping employed the discriminator for joint distribution matching [9, 8], reconstruction loss (i.e., pixel-wise L1 or L2 loss) [6], or both [33]. Although introducing inference mapping is effective in addressing mode collapse in GAN training, it cannot avoid either training instability or a trade-off issue between the image diversity and image equality.

The discriminator for joint distribution matching is used to evaluate both the generation and the inference mapping by distinguishing between two joint distributions: the joint distribution of the real data and its inferred latent vector from an encoder, and that of the real latent vector and its generated data from a generator. In other words, a single discriminator should achieve two different goals. D evaluates 1) whether the generated data are real and 2) whether both joint distributions match. Thus, the discriminator becomes insensitive to subtle changes in each distribution. Consequently, the training is difficult to converge, thus leading to the degeneration of sample quality and to the increase in training instability. With this reconstruction loss, MDGAN and VEEGAN improve inference mapping compared to ALI and BiGAN. However, it is difficult to tune parameters for balancing between adversarial loss and reconstruction loss because their units are different. (e.g., adversarial loss measures the divergence, whereas reconstruction loss measures the pixel difference)

In our work, we use the encoder only for regularizing the generator training in a way that the generator tends to follow the forward KL properties. It is possible because the data distribution projected onto the encoder, P_m, is a mode regularized version of its target data distribution. Therefore, our guidance network encourages the generator to learn the overall modes of P_{data} and to not be distracted by either joint distribution matching or reconstruction loss.

4. Evaluation

For the quantitative and qualitative evaluations, we utilized one simulated and three real datasets: CelebA [22], CIFAR-10 [17], and Stacked MNIST [25]. Note that the input dimensionality of CelebA is (64, 64, 3), that of CIFAR-10 is (32, 32, 3), and that of Stacked MNIST is (28, 28, 3). A denoising autoencoder [7] was adopted for the guidance network to encourage robust feature extraction.

Table 1. Results of the Stacked MNIST evaluation measuring the covered modes over the 1k modes MNIST and the KL divergence between the model distribution and the data distribution. MGGAN achieved better performance than those of the other competitive GANs (i.e., DCGAN, ALI/BiGAN, and VEEGAN).

MODEL	COVERED MODES	KL DIVERGENCE
DCGAN	99	3.4741
ALI/BIGAN	148	3.0982
VEEGAN	182	2.9534
MGGAN (α)		
0.7	382	2.4436
0.5	418	2.4088
0.3	999	0.3642

4.1. Synthetic data

To demonstrate that the guidance network helps GANs prevent mode missing, we trained and tested the network using a simple 2D Gaussian mixture model, eight modes of which were evenly distributed along a circle [25]. We set the standard deviation (std) to 0.01 and 0.35, respectively, to investigate how the interval among modes affects mode collapse. Fig. 2 compares the MGGAN, GAN, Unrolled GAN [1], and VEEGAN [2] models. When modes were far apart (i.e., std = 0.01), the GAN suffered from mode collapse, whereas other models effectively solved this problem. In contrast, when the modes were adjacent (i.e., std = 0.35), unrolled GAN and VEEGAN captured almost all modes, but generated highly scattered samples that did not accurately represent the true distribution. Unlike in the earlier example, the GAN outperformed both unrolled GAN and VEEGAN in the latter experiment. In both cases, our MGGAN consistently resolved mode collapse with an accurate representation.

Interestingly, we observed that MGGAN first captured each mode and then deviated from mode collapse; Fig. 2 supports this when the std is 0.01. This is because MGGAN is based on the standard GAN, but the guidance network induces a generator to learn the entire modes. For this reason, MGGAN shows learning patterns similar to those of the GAN with a std of 0.35 and can generate samples of fine quality similar to that of the GAN.

4.2. Quantitative evaluation

Here, we evaluated MGGAN on the Stacked MNIST, CelebA, and CIFAR-10 datasets.

Stacked MNIST. For the quantitative evaluation to solve the mode collapse, we utilized the Stacked MNIST dataset. This dataset was synthesized by randomly concatenating three MNIST samples to construct 1k modes (i.e., from 000 to 999). With the Stacked MNIST, we first trained other

[1]Refer to http://github.com/poolio/unrolled_gan.
[2]Refer to http://github.com/akashgit/VEEGAN.

Table 2. Comparison of the image diversity using the MS-SSIM with the diversity ratio of each algorithm (i.e., the diversity of the generated images divided by the diversity of a real dataset) and FID. Four baseline GANs and our MGGANs were compared. Note that the MS-SIMM of a real dataset is 0.3727. NB: The lower the MS-SSIM and FID, the higher the diversity.

METRIC		DCGAN	LSGAN	DRAGAN	DFM
MS-SSIM	ORIGINAL	0.4695 (79.38%)	0.3904 (95.46%)	0.3934 (94.74%)	0.3996 (93.27%)
	WITH MG	0.3872 (96.26%)	0.3784 (98.50%)	0.3899 (95.59%)	0.3814 (97.72%)
FID	ORIGINAL	14.9491	16.6938	14.8731	13.0080
	WITH MG	14.0103	15.9083	14.7922	12.8047

competitive GANs (i.e., DCGAN, ALI/BiGAN, and VEEGAN) and our MGGAN, where the architecture of all models is identical to that of DCGAN[3]. Then, we counted the number of modes produced by each GAN model. To do this, we applied a pre-trained MNIST classifier on 50k samples generated from each model and aggregated the number of distinct classes [25]. On the assumption that at least one sample per mode is generated out of 50k generated samples, the ideal performance is to cover 1k modes. Additionally, we measured a reverse KL divergence between the generated sample distribution (i.e., model distribution) and the data distribution, which was considered as a uniform distribution over all 1k modes. The evaluation results are summarized in Table 1. From this experiment, our MGGAN achieved better performance than those of other models in both measures; we recovered more modes, but obtained a smaller divergence. Interestingly, the more we relied on the guidance network (i.e., the smaller α), the greater we recovered the modes; almost all modes were recovered at $\alpha = 0.3$. From these results, we confirmed that adding the guidance network with the proper α effectively alleviates the mode collapse.

CelebA. To evaluate the image diversity using the MS-SSIM [28], we used only the CelebA dataset.

To evaluate the extendibility of our MGGAN, we constructed four variants of MGGAN. That is, we selected four different GANs as baseline networks and then modified each by adding the guidance network. The baseline GANs reported state-of-the-art visual quality in data generation, but were prone to mode collapse. In this study, we utilized four baseline networks, namely, DCGAN [29], LSGAN [23], DRAGAN [16], and DFM [35], and developed variants of MGGAN, namely, DCGAN-MG, LSGAN-MG, DRAGAN-MG, and DFM-MG. For a fair comparison, the network architectures of both a generator and a discriminator follow that of DCGAN. Moreover, we utilized the suggested hyperparameters from each baseline work without any fine-tuning. Our implementation code has been made publicly available at https://github.com/QuickSolverKyle/Tensorflow-MyGANs.

To compare the four variants of MGGAN with their respective baseline GANs, we measure FID and MS-SSIM; 10K and 100 samples generated from four baseline GANs with and without the guidance network are used. Table 2 summarizes the average score of the MS-SSIM and FID measurements repeated 10 times for each model. From this experiment, we find that the four variants of our MGGAN significantly improve the image diversity compared to the baseline GANs all the time. Note that a smaller MS-SSIM implies better diversity and the lower FID indicates better diversity and quality. Furthermore, the MS-SSIM values of all MGGANs were close to that of real data (i.e., 0.3727).

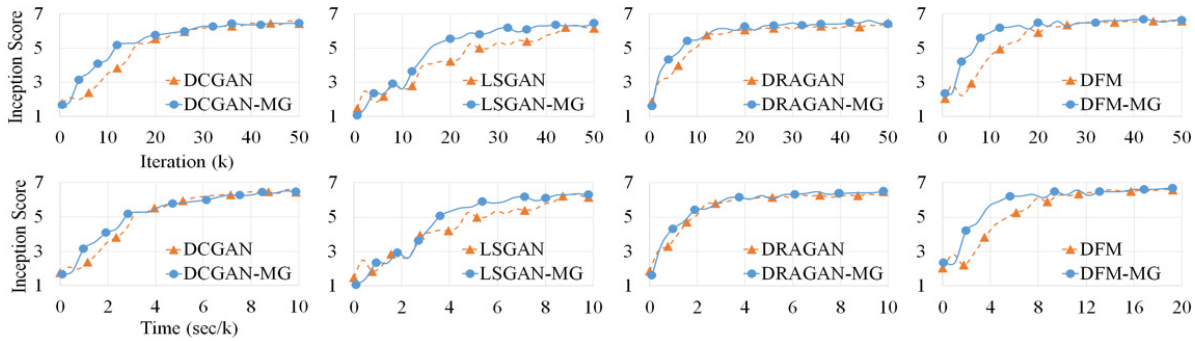
These show that the proposed model is effective in handling mode collapse. The reason is that the level of image diversity from the proposed model nearly approaches its optimal limit, which is the image diversity of a real dataset. Particularly, DCGAN-MG showed a notable improvement over DCGAN because DCGAN is more prone to mode collapse.

PacGAN [21] reported impressive performance with regard to increasing the generation diversity by using the Stacked MNIST and CelebA datasets. Specifically, they performed best on Stacked MNIST by covering all modes. To compare MGGAN with PacGAN on a real dataset (i.e., CelebA), we reproduce PacGAN2 based on DCGAN by referring the official code [4] and compare it with ours by using FID scores. While PacGan scores 14.1194, MGGAN achieves 14.0103 with an α of 0.5 and a latent dimension of 128. This result indicates that MGGAN provides comparable yet slightly better performance than PacGAN in the real dataset. Other than the comparable performance for improving the diversity, we highlight that the idea of MGGAN can be extended to other applications, such as Resembled GAN [4].

CIFAR-10. The inception score [32] was used to assess the visual quality of GANs using the CIFAR-10 dataset, and a larger score represents higher quality. Following Salimans et al. [32], we computed the inception score for 50k generated images from baseline GANs and our MGGANs using CIFAR-10. Fig. 3 plots the inception score as a function of iteration (top) and time (bottom), respectively. The table in Fig. 3 summarizes the average of the inception score from four baseline GANs and the corresponding MGGANs. We observed that the inception score from DFM was not as high as that reported in [29]. This drop might be caused by the modification to the network architecture of DCGAN. Still, DFM showed the highest score among other GANs. From this experiment, we observed that the inception scores did not decrease in our model, and this observation held for four different variants. More specifically, we confirmed

[3]Refer to github.com/LazarValkov/GanModeCollapseEvaluation.

[4]Refer to http://github.com/fjxmlzn/PacGAN



Metric	Method	DCGAN	LSGAN	DRAGAN	DFM	Metric	Method	DCGAN	LSGAN	DRAGAN	DFM
Inception score	Original	6.4706	6.3243	6.4468	6.5854	FID	Original	53.6387	56.8106	52.8582	52.4390
	with MG	6.4728	6.3416	6.4942	6.6076		with MG	53.2985	53.1518	52.4649	51.7607

Figure 3. Comparison of the inception scores as a function of iteration and time and FID (mean $\pm$ std) on CIFAR-10. The inception scores and FID (mean $\pm$ std) in the table are the average scores of five repeated measurements of each model.

that our MGGAN can achieve the image quality of baseline GANs within approximately 0.04 tolerance of the inception score. We conduct an additional evaluation of measuring the FID scores on CIFAR-10 and confirm the improvement in all baseline networks. As the FID score measures the sample quality and diversity simultaneously, the amount of improvement clearly demonstrates our improvement.

We show the benefit of MGGAN by using the state-of-the art model. Among many recent models [26, 24, 14, 1] reporting a high inception score, we choose [26] as a baseline GAN model because other models have nontrivial implementation issues in modifying the training scheme. To reproduce [26], we replace the batch normalization of the DCGAN discriminator with spectral normalization. We observe that the FID scores of SNDCGAN with/without the manifold guidance algorithm are 12.4458 and 13.3389 on CIFAR10 dataset, respectively. From the diversity improvement in SNDCGAN, we confirm that our algorithm can improve the performance of the recent state-of-the art model with a meaningful margin.

4.3. Qualitative evaluation

In this section, we discuss the effect of the guidance network and determine whether it 1) causes degradation in visual quality, and 2) induces a meaningful manifold mapping to increase the image diversity.

First, we compared the generated images from the baseline models and the corresponding MGGANs. Fig. 4 visualizes those results. The left-hand side shows the generated images from the baseline GAN, whereas the right-hand side presents those from the MGGAN. From this qualitative comparison, it is difficult to recognize the quality difference from both results. Therefore, our achievement in improving the image diversity is not the result of sacrificing the visual quality. These results are analogous to the quantitative evaluation reported in the table of Fig. 3.

Second, we examined whether the encoder mapping induces a meaningful manifold space for enhancing diversity. In this experiment, we claim that the generator can cover more modes if it reproduces more accurate images. It is a reasonable statement because the image reconstruction can be an effective tool for measuring the quality of mapping between the latent and the image. To assess the mode coverage, we devised a new method to generate the reconstructed image using the manifold and the generator. For that, we built an additional network that transforms our manifold space P_m into a latent space P_z to infer the latent vector of real data. Because this network transforms the encoder output into a latent vector, we could construct a cyclic mapping, i.e., $z \Rightarrow x \Rightarrow m \Rightarrow z$. Although this additional network is never utilized during our training, we intentionally developed this network to derive $\widehat{z}$ corresponding to x and then reconstruct x using the generator $G(\widehat{z})$. On the basis of this reconstruction experiment, we could evaluate how accurate our model can reproduce the real data, even without explicitly imposing the reconstruction loss. A network for linking P_m and P_z is composed of 1024 fully connected layer (FC) $-$ batch normalization (BN) $-$ rectified linear unit (ReLU) (1024 FC $-$ BN $-$ ReLU) dimension of P_z FC. Fig. 5 shows the reconstructed images with their target images. They were from the CelebA test dataset, and all four variants (i.e., DCGAN-MG, LSGAN-MG, DRAGAN-MG, and DFM-MG) were investigated. Odd columns show the target images, whereas even columns are their reconstructed images. The results from ALI did not faithfully restore the attribute of the target faces, such as the gender, glasses, and background color. On the contrary, our MGGANs reproduced the target images reasonably well, maintaining the original attributes. From this experiment, we could confirm that our MGGAN produced more accurate reconstruction

Figure 4. Comparison between randomly generated samples from the original baseline GANs (DCGAN, LSGAN, DRAGAN, and DFM) and the corresponding MGGANs (DCGAN-MG, LSGAN-MG, DRAGAN-MG, and DFM-MG).

	ALI	DCGAN-MG	LSGAN-MG	DRAGAN-MG	DFM-MG
PSNR	11.95 ± 0.12	16.52 ± 0.06	16.47 ± 0.13	16.54 ± 0.04	16.46 ± 0.10
SSIM	0.2731 ± 0.0085	0.5112 ± 0.0060	0.5007 ± 0.0042	0.4999 ± 0.0060	0.5015 ± 0.0087

Figure 5. Reconstruction quality comparison of MGGAN variants (DCGAN-MG, LSGAN-MG, DRAGAN-MG, and DFM-MG) with ALI [6]. The architecture of ALI follows that of the DCGAN networks; the encoder and decoder architectures of ALI are identical to the discriminator and generator of DCGAN, respectively. Odd columns are test images of the CelebA dataset, whereas even columns are the corresponding reconstructions from each model. To quantitatively measure how well each algorithm reconstructed the original image, we employed the PSNR (mean ± std) and SSIM (mean ± std) as summarized in the following table.

results than those of the bidirectional mapping approach, namely, ALI. We believe that our high reconstruction accuracy on the overall subjects supports the effectiveness of our guidance network toward diverse mode coverage.

5. Conclusions

In this paper, we propose a new algorithm that induces a generator to produce diverse samples without sacrificing visual quality, by matching the distribution under the designed manifold using the guidance network. We found that the encoder of a pretrained autoencoder is effective to reflect the mode coverage of a true distribution, thus adopted it as a guidance network. Consequently, the generator avoids mode missing during training because it receives the feedback for the mode coverage of a data distribution from the guidance network.

We believe that this idea of manifold mapping can be further extended toward integrating prior information into generator training. We hope that our work provides a basis for future work for controlling the generator with prior knowledge.

Acknowledgement

This research was supported by the NRF Korea funded by the MSIT (NRF-2019R1A2C2006123) and the Korea Medical Device Development Fund grant (Project Number: 202011D06).

References

[1] Jonas Adler and Sebastian Lunz. Banach wasserstein gan. In *Advances in Neural Information Processing Systems*, pages 6754–6763, 2018. 7

[2] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*, 2017. 4

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. 2

[4] Duhyeon Bang and Hyunjung Shim. Resembled generative adversarial networks: Two domains with similar attributes. In *BMVC*, 2018. 6

[5] David Berthelot, Tom Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017. 1

[6] Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks. In *International Conference on Learning Representations*, 2017. 2, 5, 8

[7] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *International Conference on Learning Representations*, 2017. 5

[8] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *International Conference on Learning Representations*, 2017. 2, 5

[9] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron Courville. Adversarially learned inference. In *International Conference on Learning Representations*, 2017. 2, 5

[10] William Fedus, Mihaela Rosca, Balaji Lakshminarayanan, Andrew M Dai, Shakir Mohamed, and Ian Goodfellow. Many paths to equilibrium: Gans do not need to decrease a divergence at every step. In *International Conference on Learning Representations*, 2018. 1

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1

[12] Ian J. Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *CoRR*, abs/1701.00160, 2017. 4

[13] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5769–5779, 2017. 2

[14] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. 2, 7

[15] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2016. 3

[16] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*, 2017. 2, 6

[17] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 5

[18] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015. 3

[19] Chunyuan Li, Hao Liu, Changyou Chen, Yuchen Pu, Liqun Chen, Ricardo Henao, and Lawrence Carin. Alice: Towards understanding adversarial learning for joint distribution matching. In *Advances in Neural Information Processing Systems*, pages 5495–5503, 2017. 3

[20] Hongyu Li and I-Fan Shen. Similarity measure for vector field learning. In *International Symposium on Neural Networks*, pages 436–441. Springer, 2006. 4

[21] Zinan Lin, Ashish Khetan, Giulia Fanti, and Sewoong Oh. Pacgan: The power of two samples in generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 1498–1507, 2018. 2, 6

[22] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015. 5

[23] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *2017 IEEE International Conference on Computer Vision*, pages 2813–2821. IEEE, 2017. 2, 6

[24] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International Conference on Machine Learning*, pages 3478–3487, 2018. 2, 7

[25] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. In *International Conference on Learning Representations*, 2017. 2, 5, 6

[26] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. 2, 7

[27] Tu Nguyen, Trung Le, Hung Vu, and Dinh Phung. Dual discriminator generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2667–2677, 2017. 2

[28] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier GANs. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2642–2651, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. 6

[29] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*, 2016. 6

[30] Mihaela Rosca, Balaji Lakshminarayanan, David Warde-Farley, and Shakir Mohamed. Variational approaches for auto-encoding generative adversarial networks. *arXiv preprint arXiv:1706.04987*, 2017. 2

[31] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. In *Advances in Neural Information Processing Systems*, pages 2018–2028, 2017. 2

[32] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016. 6

[33] Akash Srivastava, Lazar Valkoz, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, pages 3310–3320, 2017. 2, 5

[34] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010. 2, 4

[35] David Warde-Farley and Yoshua Bengio. Improving generative adversarial networks with denoising feature matching. In *IEEE International Conference on Learning Representations*, 2017. 6

[36] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. 3