Appendix–MGGAN: Solving Mode Collapse Using Manifold-Guided Training

1. Preliminaries

In this section, we first show why the GAN is vulnerable to the mode collapse problem by analyzing its formulation.

GAN Training utilizes two separate networks with competitive goals. First, a discriminator, D, is trained to distinguish between the real and the fake data. Then, a generator, G, aims to create the fake data to be as real as possible to fool the discriminator. More specifically, the generator learns the generation process, which maps from the prior distribution P_z to the data distribution P_{data}. This is equivalent to an implicit model P_{model}, which approximates P_{data}. This problem can be formulated as a minimax game [7, 8]:

$$\min_G \max_D \mathop{\mathbb{E}}_{\mathrm{x}\sim \mathrm{P}_{\mathrm{data}}} [\log(\mathrm{D}(\mathrm{x}))] + \mathop{\mathbb{E}}_{z\sim \mathrm{P}_{\mathrm{z}}} [\log(1-\mathrm{D}\left(\mathrm{G}\left(\mathrm{z}\right)\right)],$$

where $\mathbb{E}$ denotes expectation, and x and z are samples drawn from P_{data} and P_{model}, respectively. Suppose that the generator produces perfect samples (i.e., $P_{model} \equiv P_{data}$). Then, the discriminator no longer distinguishes the fake data from the real data, and, thus, the game ends. This is called the Nash equilibrium.

However, this adversarial minimax game causes training instability, which is associated with two major problems of GAN training: gradient vanishing and mode collapse. Gradient vanishing becomes a serious problem because GAN training is forced to terminate. There are two well-known scenarios of gradient vanishing. The first is when the discriminator easily wins against the generator. At the early stage of the training, the generator produces images with poor quality. Hence, distinguishing those with poor fake data from those with real data becomes a relatively easy task. The second scenario is that any subset of P_{data} and P_{model} is disjoint such that the discriminator separates the real from the fake data perfectly; i.e., the generator no longer improves the data because the discriminator has reached its optimum [1]. For both scenarios, poor results are generated because training stops even though P_{model} has not learned P_{data} properly. Mode collapse describes the case in which the generator repeatedly produces the same or a similar output. The reason is that P_{model} only encapsulates a single or a few modes of P_{data} to easily fool the discriminator.

To alleviate the training instability, Goodfellow et al. [7] recommended implementing an alternative cost function for alleviating the gradient vanishing problem caused by the first scenario and it is defined as

$$\min_G \; -\mathop{\mathbb{E}}_{\mathrm{x}\sim \mathrm{P}_{\mathrm{data}}} \log(\mathrm{D}(\mathrm{G}(\mathrm{x}))).$$

From this modification, the authors intend to accelerate the early stage of generator training, preventing the discriminator from easily reaching the optimum. Fedus et al. [6] referred to it as non-saturated GAN to distinguish it from the standard GAN.

Although the standard GAN [7] theoretically proves that generative modeling can be formulated by minimizing the Jensen-Shannon divergence (JSD), its authors recommend the non-saturated GAN for the actual implementation [7, 6]. The non-saturated GAN is designed to minimize $KL(P_{model}||P_{data}) - 2JSD$ for generator update, which holds a property of the reverse KL divergence between P_{data} and P_{model} [1]. Arjovsky and Bottou [1] and Arjovsky et al. [2] pointed out that the reverse KL divergence is vulnerable to mode collapse. Because the reverse KL divergence evaluates the dissimilarity between two distributions in every fake sample (i.e., $P_{model}(x) > 0$, for all x), there is no penalty for covering only a fraction of the true data distribution. Such a theoretical justification is analogous to empirical observations; non-saturated GANs suffer frequently from mode collapse.

2. Ablation study

Types of autoencoders. Our final model used the denoising autoencoder for the guidance network. The dimension reduction methods affect the performance because each method follows a different objective function. We investigated different dimension reduction methods using a conventional autoencoder, denoising autoencoder (DAE), variational autoencoder (VAE) [10], and adversarial autoencoder (AAE) [11]. We set α to 0.5 in all the evaluations and the results are in Table 1. We conclude that the DAE-based guidance network is better than the other networks for improving the diversity. It was expected because DAE faithfully recovers the overall modes by minimizing the sample reconstruction error and obeying the forward KL property.

Table 1. Comparison of the performance following the different dimension reduction methods on the evaluation of the Stacked MNIST dataset.

MODEL	COVERED MODES	KL DIVERGENCE
AUTOENCODER	306	2.8994
DAE	418	2.4088
VAE	262	3.0238
AAE	301	2.8992

Meanwhile, VAE and AAE opt to sacrifice the reconstruction accuracy in favor of generation ability.

Effect of the latent dimension. We investigate the effect of the latent dimension using the Stacked MNIST dataset as this dimension can be an important hyperparameter. As shown in Table 3, MGGAN became less effective in capturing various modes when the dimension was too high. The reason was that the P_{data} projected on that manifold may not be much different from P_{data}; as the latent dimension approaches to the data dimension, the feedback from guidance network is nearly identical to that of the discriminator. Likewise, if the dimension was too low, the encoder discards too much information of P_{data}; the loss from the guidance network was less informative to GAN training. Yet, in either case, MGGAN still achieved better performance than those of the other competitive GANs (i.e., DCGAN, ALI/BiGAN, and VEEGAN).

To analyze the effect of latent size on various datasets, we evaluate the FID of the generated samples using CIFAR-10 and CelebA datasets. From this empirical study, we aim to observe whether the complexity of datasets affects the optimal size of the latent dimension (CIFAR-10 and CelebA are more complicated than Stacked MNIST). Interestingly, the optimal latent size is 128 regardless of the complexity of the dataset.

Effect of alpha. We measured the inception score on CIFAR-10 along the change in the alpha value and the FID score on both CIFAR-10 and CelebA. Table 2 shows that the inception score was similar to those of the baseline GANs when α was greater than 0.5. However, the lower the alpha value, the poorer the quality. Also, we conduct an additional evaluation of measuring the FID scores on both CIFAR-10 and CelebA, 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. From these results, one might consider that MGGAN still suffers from the trade-off between quality and diversity discussed in existing studies [4, 6]. However, we consistently observed that MGGAN improved the diversity while preserving the quality with the proper α, which, in our case, was 0.5. Using the proper alpha value, MGGAN improved the diversity without sacrificing the quality.

Table 2. Comparison of the inception scores and FID (mean $\pm$ std) along the change in α. We evaluated the DCGAN-based MGGAN. When α is zero, training is failed in all cases.

Dataset	α	1.0	0.7	0.5	0.3
CIFAR-10	Inception score	6.4706	6.4709	6.4728	6.4259
	FID (mean $\pm$ std)	53.6387 $\pm$1.4198	53.0271 $\pm$1.4224	53.2985 $\pm$1.3253	52.6721 $\pm$1.4255
CelebA	FID (mean $\pm$ std)	14.9491 $\pm$0.3549	13.5833 $\pm$0.4388	14.0103 $\pm$0.3889	14.7570 $\pm$0.5833

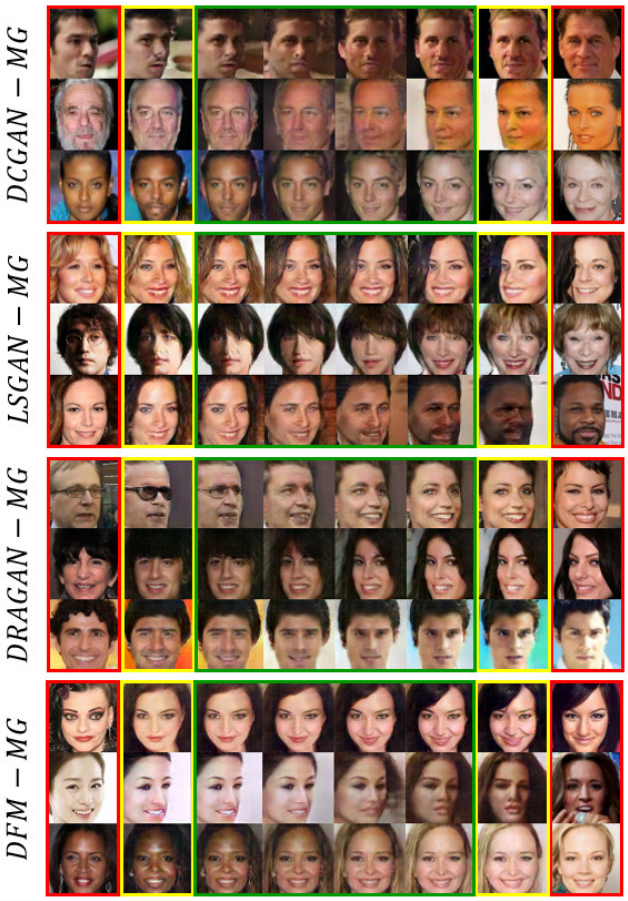

Figure 1. Latent space interpolations from the CelebA dataset. Left- and rightmost columns (marked red box) are the test images and, just beside them (marked yellow box) are the corresponding reconstructions. Intermediate columns between them (marked green box) are linear interpolations in the latent space between reconstructions.

3. More qualitative results

In this section, we generated samples by walking in a latent space to verify whether data generation is the result of data memorization. Because our generator learns representative features in manifold, P_m, derived from P_{data} solely, it might be reasonable to suspect overfitting of the training data. To clarify this issue, we show the image generation results by latent walking in Fig. 1. Note that we chose

Table 3. Comparison of the performance by varying the latent dimension on the evaluation of the Stacked MNIST, CIFAR-10, and CelebA dataset.

	Stacked MNIST		CIFAR-10	CelebA
Dimension	Covered modes	KL divergence	FID	FID
1024	238	2.5289	54.6684 ± 1.3484	15.1771 ± 0.6553
512	382	2.5585	53.9497 ± 1.4322	15.1282 ± 0.3831
256	418	2.4088	53.3623 ± 1.4224	14.2799 ± 0.4322
128	407	2.6080	53.2985 ± 1.3253	14.0103 ± 0.3889
64	327	2.9929	54.4036 ± 1.2877	15.2169 ± 0.4121

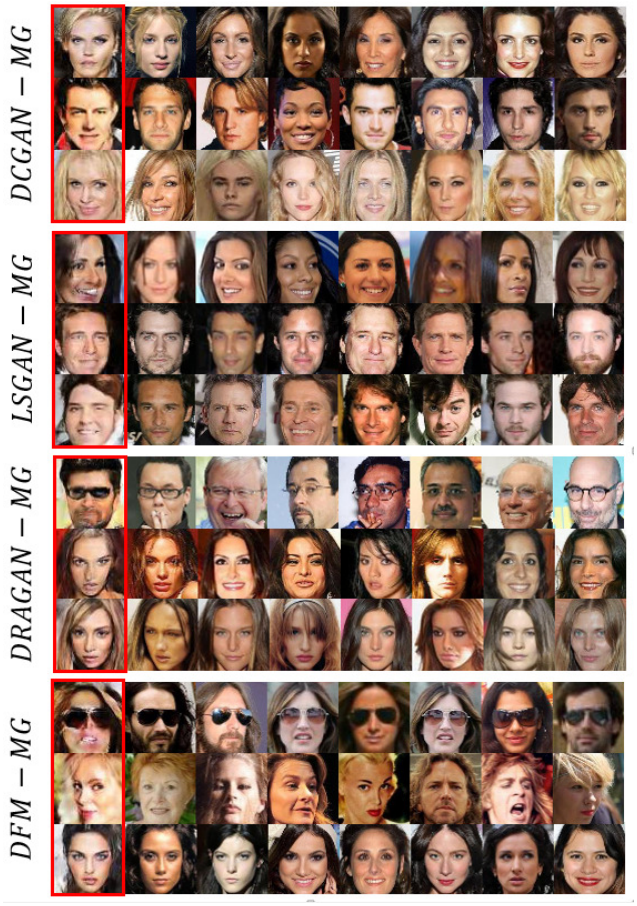

Figure 2. Nearest neighbors in ResNet-18-avgpool feature space among the generated samples and training dataset from CelebA. Leftmost columns (marked red box) are the generated images and, just beside them are the nearest neighbors from rank one to seven in order.

two latent vectors, which were derived from the CelebA test dataset using the above network (connecting the manifold to the latent space). According to Radford et al. [12], Dinh et al. [5], and Bengio et al. [3], the interpolated images between two images in a latent space do not have meaningful connectivity when the networks just memorize the dataset, such as the lack of smooth transitions or the failure to generate. However, because our MGGAN produces natural interpolations with various examples, we conclude that MGGAN learns the meaningful landscape in a latent space. Furthermore, we search the top-7 most similar training samples for each generated sample by performing the nearest neighbors in the feature space of pretrained classifier networks (i.e., ResNet-18-avgpool [9]) and show them in Fig. 2. The leftmost columns (marked by the red box) are the generated images, and the seven samples in each row are the corresponding neighbors from rank one to seven. As the nearest neighbors are similar to the generated image but visually distinct, we confirm that our achievement is not caused by memorizing the training dataset.

References

[1] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*, 2017. 1

[2] 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. 1

[3] Yoshua Bengio, Grégoire Mesnil, Yann Dauphin, and Salah Rifai. Better mixing via deep representations. In *International Conference on Machine Learning*, pages 552–560, 2013. 3

[4] David Berthelot, Tom Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017. 2

[5] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *International Conference on Learning Representations*, 2017. 3

[6] 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, 2

[7] 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

[8] Ian J. Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *CoRR*, abs/1701.00160, 2017. 1

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 3

[10] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2016. 1

[11] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. In *International Conference on Learning Representations*, 2016. 1

[12] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*, 2016. 3