Study of Prevention of Mode Collapse in Generative Adversarial Network (GAN)

Bhagyashree, Vandana Kushwaha, G. C. Nandi
Center of Intelligent Robotics
Indian Institute of Information Technology, Allahabad
Prayagraj, India-211015
bhagyashreemahi27@gmail.com, kush.vandu@gmail.com, gcnandi@iiita.ac.in

***Abstract*—With the advancement of Deep Neural Network and its increased applications, requirement of data has increased exponentially. To fulfil this requirement Deep generative models specifically Generative Adversarial Networks (GANs) have emerged as a very powerful tool. However, tuning GAN parameters are extremely difficult due to its instability and it is very prone to miss modes while training, which is termed as Mode Collapse. Mode collapse leads generators to generate the images of a particular mode while ignoring the other mode classes. In the present research, we propose a novel method to deal with Mode Collapse by using multiple generator architecture. Initially, We have shown the comparison of different GAN architectures which deals with the Mode collapse problem. We use Inception score (IS) as a evaluation metric to evaluate the performance of GAN. We started analysing GAN on a simple dataset (MNIST) using DCGAN architecture. To produce better results, the present work describes the implementation of two other different approaches. We have experimented on Wasserstein GAN (WGAN) which improves GANs training by adopting a different metric which is Wasserstein distance for calculating the distance between two probability distributions. Subsequently we have proposed the approach of multiple generator GAN architecture which uses multiple generators to provide a better solution to the missing modes problem. We evaluate our approach on several datasets (MNIST, CIFAR-10, SVHN, CelebA Face dataset) with encouraging results compared to the other existing architectures.**

***Index Terms*—Deep learning, Generative Adversarial Networks, Mode Collapse, Inception Score, Frechet Inception Distance, Wasserstein metric**

I. Introduction

Generative adversarial network (GANs) [1] has proven to be an interesting and rapidly changing field in the area of Deep generative models because of its ability to generate samples as realistic as possible. In Deep neural networks there are basically two basic approaches to use it as data generative models are Variational auto-encoders (VAEs) [2] and generative adversarial nets (GAN) [1]. Generative Adversarial Nets (GANs) belongs to the class of deep generative models which are based on game theory which is applied to a various tasks such as image generation, video generation [3], image in-painting [4], semantic segmentation, image-to-image translation, and text-to-image synthesis. GAN model consists two nets: Generator and Discriminator network. Both these networks are trained in adversarial model following a two-player mini-max game theory. The task of the Generator is to generate images which resembles the training data (real images) and while Discriminator distinguishes between the generated samples and real samples.

978-0-7381-2447-6/20/$31.00 ©2020 IEEE

Despite the enormous success of GAN, GANs suffers from many problems as it is very hard to train due to their training instability because of simultaneous training of generator and discriminator. GAN also suffers from a major problem of Mode collapse [1], [5], [6], where generator tends to generate a limited variety of samples. Generator constantly tries to maximize the mistakes of Discriminator so that it won't be able to identify fake images. At convergence, Generator learns to produce samples resembling to real samples. Though, theoretically, Convergence means that generator learns perfectly to generate samples like real images. However, practically, reaching to convergence is very difficult which leads to the problem of mode collapse. Theoretically, the general cause behind generator getting into the trap of generating the limited modes is the lack of control on discriminator during training.

There has been a lot of research in the area of solving the mode collapse problem. There are two ways to address the problem of mode collapse: (1) use of better metric to improve the learning of GANs to reach better optima by using a better metric [7]; (2) explicitly enforcing GANs to capture diverse modes by using multiple generators [5], [6]. Quan Hoang et al. [8] addresses the problem of Mode collapse by using the concept of multiple generators. Inspired by their work, we propose an alternative approach in this paper.

In this work, we have proposed the approach of using multiple generators with just one discriminator. Some recent work has been done using multiple generator approach [8], by simultaneously training a set of generators, encouraging them to generate samples of diverse modes. Our main goal is to ensure that diverse modes are captured using multiple generator approach. In this approach we enforce the different generators to generate diverse samples with a slight modification to the Discriminator's objective. Discriminator network will also be able to identify that which of the generator have been used to generate the samples along with classifying the sample as real or fake.

In addition, we analyze our model performance by experimenting on various datasets and by comparing it with the other architecture models. Firstly, we have trained our model for the

basic MNIST digit dataset and we observe that our method has outperformed WGAN and DCGAN models by generating higher quality images. Further, we have trained our model for more complex datasets such as CIFAR-10, SVHN and CelebA face dataset.

Our main contribution could be listed as:

- We define a novel GAN model with multiple generators whose discriminator can classify the samples as real or fake and also can identify the generator which is been used to generate the sample.
- We observe that using multiple generators, we achieve superior performance in case of MNIST dataset as compared to the previous state-of-the-art models.

II. ANALYSIS OF PREVIOUS RESEARCH

There has been a lot of research on GAN and it's different variations. In 2014, Goodfellow et al. [1] proposed a new framework where both generator and discriminator are trained in adversarial manner. The goal of the GAN is to generate realistic data as much as possible. GAN produced promising results in many fields such as image generation, video generation [3], image in-painting [4], semantic segmentation, image-to-image translation, and text-to-image synthesis. However training GAN is highly unstable [7], hard to achieve Nash equilibrium ($p_r = p_g$), problem of vanishing gradients (where Discriminator plays its part perfectly in identifying the images as real or fake) and it tends to generate a specific mode samples leading it to the mode collapse in GAN. One main cause behind this problem is lack of control on discriminator during the training.

To deal with the problem of unstable training of GAN, Martin Arjovsky et al. [7] proposed a new approach Wasserstein GAN (WGAN), which uses wasserstein distance [7] as their objective function. In WGAN, Clipping parameter is used to clip the weights which allows training of the discriminator till optimality before updating each generator. Though, enforcing Lipschitz constraint on the critic with weight clipping is difficult. Weight clipping might lead to optimization difficulties [9] in case of large value of clipping parameter and can easily lead to vanishing gradients when clipping value is small. WGAN proves to be more robust than GAN architecture because of its ability to train the critic till optimality. However, when high learning rates are used or momentum based optimizer such as Adam [10] is used then training WGAN becomes unstable at times.

To solve the problem of WGAN, Ishaan Gulrajani et al. [9] proposed an alternative approach to add the gradient penalties in the objective function to enforce the Lipschitz constraint. In WGAN, hyperparameter c which is clipping parameter needs to be tuned carefully else it might result in the problem of Vanishing gradients or Exploding gradients. In improved GAN, there's no such need to tune the hyperparameter c. Although, Weight-gradient WGAN overcomes the problem of WGAN, Batch normalization is not used in network architecture. Since Batch normalization maps whole batch of inputs to the output batch which doesn't fit with the approach of WGAN-GP, as critic's gradients are penalized with each input independently.

Further, many experiments have been made to deal with the problem of Mode collapse either by improving the training by adapting a better metric or by enforcing multiple generators. Luke Metz et al. [11] proposed the method of Unrolled GAN, which modifies the objective function of the Discriminator enforcing mini-batch discrimination. Mini-batch discrimination will help the Discriminator to easily detect if similar samples are being generated by the generator. Although, Unrolled GAN tackles the problem of Mode collapse effectively, it is computationally expensive for the large scale datasets.

Ming-Yu Liu et al. [12] proposed the approach of Coupled GAN, to learn the joint distribution parameters are shared between two different generators. Since the generators are trained independently, it produces diverse generations. In case of multiple generators, Quan Hoang at al. [8] proposed a method to simultaneously train a set of generators by encouraging them to generate samples of diverse modes by sharing parameters. In MGAN [8], an additional classifier is added which differentiates that which generator is used to generate the samples. In this approach, they try to maximize the JSD between the generators so that it will enforce the generators to generate more varied samples which will help in avoiding the problem of mode collapse.

Our approach is closely related to the approach of MGAN, instead of using an additional classifier we have modified the objective of discriminator in such a way that along with identifying the real or fake samples it also identifies the generator which is used to generated the fake sample and each generator is then pushed towards generating the diverse samples.

III. PRELIMINARIES

A. *Generative Adversarial Networks*

GAN is basically inspired by the mini-max game theory. GAN model consists a pair of deep neural nets termed as Generator and Discriminator. Generator G(z) accepts random noise distribution z as an input and it outputs a generated sample (fake image). A discriminator network D(x) which will provide the distribution of the probability sample belonging to the real data samples.

Both the models work in adversarial manner to each other where Generator G(z) is being trained to confuse the discriminator by making it to believe that the generated samples by the generator came from the training data distribution and it constantly tries to maximize this confusion.

The main purpose of the GAN generator G is to generate the images which can confuse discriminator D the most as produced by [1]:

$$\min_{\theta} E_{z\sim p_z(z)}[log(1 - D_{\Phi}(G_{\theta}(z)))] \tag{1}$$

However, Discriminator's objective is to maximizing probability distribution of true data and minimizing the produced data probability distribution.

$$\max_{\Phi} \left[E_{x\sim p_{data}(x)}[log(D_{\Phi}(x))] + E_{z\sim p_z}[1 - log(D_{\Phi}(G_{\theta}(z)))]\right] \tag{2}$$

Overall mathematical formulation of the GAN is [1],

$$\min_{\theta}\max_{\Phi} E_{x\sim p_{data}(x)}[log(D_{\phi}(x)] + E_{z\sim p_z(z)}[log(1 - D_{\Phi}(G_{\theta}(z))] \tag{3}$$

As illustrated in Equation 3, G maps every noise sample z to a single x which is to be classified as real data, which leads it to collapse around only few modes. Another reason which causes the mode collapse is that it tries to minimize the loss in between the true data and the generated sample [8].

IV. METHODOLOGY

To deal with the problem of Mode collapse in GAN, we have proposed a new approach of using multiple generators instead of just one generator with only one discriminator which will act as a classifier as well as will identify the generator from which the sample has been generated. Our main focus is that generator should be able to generate various samples instead of just trying to confuse the discriminator by generating just few samples resembling to the real data samples. We try to enforce the diversity between the generators.

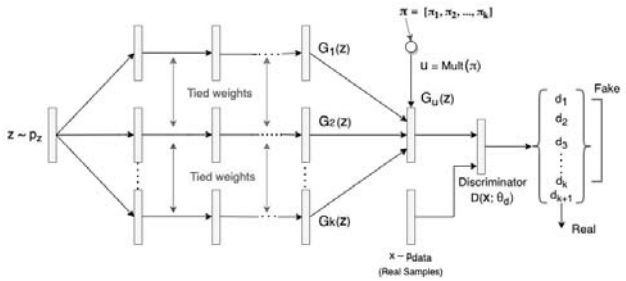

Fig. 1. Multi generator GAN architecture with k generators and one discriminator which outputs (k+1) soft-max scores classifying samples as real or fake and identify the generator used to generate the samples.

As illustrated in Fig 1, our proposed model of multiple generator GAN have k generators with just one discriminator. All the generators share the same latent space. Parameters are shared in between the hidden layers. This parameter sharing mechanism enables the network to borrow or share the common information, leading in effective training of the model. It also helps in minimizing the number of parameter and also adds minimal complexity to the standard GAN model.

In this work, we have multiple generators, Let G_k denote a generator in a set G = $\{G_1, G_2, , G_k\}$, and z is a random latent vector. Each generator G_k(z) will map z (latent space) to x (generated sample). We have inherited this idea of using multiple generators from the previous work [8]. Every generator will induce a individual sample distribution and then one of them will be selected randomly to produce the output and fed it as input to the discriminator. An index u will be drawn from the distribution Mult (π) where $\pi = [\pi_1, \pi_2, ..., \pi_k]$ is the coefficients of the mixture; and then the sample G_u(z) is used as the output [8].

Discriminator plays the role of classifier to distinguish between the generated samples and the real samples along with identifying the generator from which the sample has been generated. We have modified the objective function of the discriminator as opposed to the standard GAN, where it outputs only a scalar value identifying whether the generated sample is fake or real. Our discriminator output is (k+1) soft-max values for k generators. The soft-max value of D_{k+1} gives the probability of the sample belonging to the real data distribution whereas values at D_1 to D_k represents the generated samples probability generated by which generator.

As explained in Standard GAN, Discriminator's objective is to optimize θ_d while keeping θ_g constant. We have modified discriminator's objective function in such a way that it enforces each generator to learn different modes of a dataset. Thus, the objective of the discriminator would be:

$$\max_{\theta_d} E_{x\sim p_r(x)}(\delta; D(x; \theta_d)) \tag{4}$$

Where $\delta \in \{0, 1\}^{k+1}$. However, while learning θ_d cross-entropy between the soft-max outputs of δ and the discriminator outputs.

However, each generator's objective will remain same as explained in the standard GAN.

$$\min_{G_{1:k}} E_{z\sim p_z(z)}[log(1 - D(G_k(z; {\theta_g}^k)))] \tag{5}$$

Overall objective function of the GAN is,

$$\min_{G_{1:k}}\max_{D} L(D, G) = E_{x\sim p_r}[log(D_{k+1}(x; \theta_d))] + E_{z\sim p_z}[log(1 - D_{k+1}(G_k(z; {\theta_g}^k); \theta_d))] \tag{6}$$

Algorithm 1 presents the pseudo code of the proposed approach.

V. RESULTS AND ANALYSIS

Experiments are carried out on various large scale datasets. We started experiments on the basic dataset of MNIST handwritten digits dataset to analyse and to evaluate our proposed approach performance by visualizing the generated images at each iteration. Further, we have used various large scale datasets such as CIFAR-10, SVHN and CelebA to examine the behaviour of our model on different datasets. Results are then being compared to previous state-of-art models.

A. *Dataset Description*

We use 4 widely used real world datasets: MNIST [13], CIFAR-10 [14], CelebA [15], SVHN [16]. MNIST dataset is a handwritten digits dataset in which there are 60,000 training samples and 10,000 test samples with 10 modes. CIFAR-10

Algorithm 1 Algorithm for updating the gradients for each generator in multiple generator GAN

Input:
Noise sample(z), θ_d, ${\theta_g}^k$
learning rate λ
Batch Size
Output:
Samples generated by the k generators

procedure GENERATOR(Noise Samples: z)
  for number of iterations **do**
    for number of generators i = 1 to k **do**
      Sample noise z from distribution $z \sim P_z(z)$
      Sample m training samples of given batch size $x^i \sim P_{data}(x)$
      Obtain generated samples $G_i(z; \theta_g(i))$ along with generator index i
      Update weight (θ_d) of discriminator D:
      for j in i $\epsilon 1...k$ **do**
        Compute delta $\delta\epsilon(0,1)^{K+1}$
        Update parameters of each generator : $\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} logD(G_j(z; \theta_g^i); \theta_d)$
      end for
      $\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} log(D(G_i(z; \theta_g^i); \theta_d)) + logD(x^{(i)})$
      Update weight of generator G: $\nabla_{\theta_{g^i}} \frac{1}{m} \sum i = 1^m log(1 - D(G_i(z; \theta_g^{(i)}); \theta_d))$
    end for
  end for
end procedure

is a dataset which contains 60,000 samples of 32*32 color images in 10 modes: airplane, bird, cat, deer, ship, truck, horse, frog, dog and automobile.

CelebA dataset is a face attribute dataset consisting of more than 200k images of 218*178 color images. We resize it to 32*32*3 to train in our model. The Street View House Numbers (SVHN) dataset has 73,257 digits and 26,032 digits for training and testing set respectively of 32*32 color images.

B. Experimental Settings

To perform experiments Google Colaboratory is used. For all experiments, we use: (i) TensorFlow to implement the model; (ii) Weights randomly initialized with gaussian distribution with noise dimension 100; (iii) Adam optimizer [10] with learning rate **1e-5**; (iv) Batch size of 128 samples to train discriminator; (v) LeakyReLU activation [17] with a slope of 0.2 for generators and with slope 0.02 for discriminators. Our Model is trained for 100 epochs.

C. Performance Evaluation

As mentioned before that there is not a very clear way to evaluate GAN models. For Analysis, we can observe the generated images and the loss plot of Generator and Discriminator.

GAN Model	CIFAR-10	SVHN	CelebA
Vanilla GAN [1]	1.14 ± 0.16	2.06 ± 0.02	-
DCGAN [3]	6.34	3.05	3.82 ± 0.02
WGAN [7]	4.83 ± 0.04	3.97 ± 0.02	3.07 ± 0.22
Multiple generator GAN	2.01 ± 0.007	3.64 ± 0.04	4.21 ± 0.16

TABLE I
RESULT OF INCEPTION SCORE(HIGHER BETTER) ON DIFFERENT DATASETS FROM THE DIFFERENT MODELS FOR 50K GENERATED SAMPLES. **NOTE: AS OF NOW, IS CALCULATED FOR OUR APPROACH IS FOR 5K SAMPLES ONLY.**

For evaluating the generated images quality, we have used Inception score proposed by Salimans et al. [18], which is computed by using a Inception v3 model which is pre-trained, to predict the class probabilities for each generated image. Inception score metric rewards good and varied samples and is found to be well-correlated with human judgement [18]. We have used the code given by Salimans et al. [18] to calculate the IS for 50000 generated samples.

*1) **Inception Results**:* Inception Network takes 3 channel images as an input as it is pre-trained on the ImageNet Dataset. Therefore, We are only able to test our model for CIFAR-10, SVHN and CelebA dataset only. We have scaled the samples from (32, 32) to (299, 299) to get the Inception score.

Table I illustrates the Inception Score computed on different model for different datasets. As for our proposed approach, GANs were relatively under-trained due to the use of multiple generators, the IS results did not turn out to be as expected. But as we have observed that when using multiple generators, at least (n-1) generators are able to learn different modes samples, it is highly likely that fully trained GAN would give better output contributing towards a better IS score.

*2) **Images generated**:* To sum up the results we have got for MNIST dataset using our proposed model, we are clearly able to observe in Fig. 2 that the samples generated by this model are of good quality and clearly identifiable. Also, it captures diverse modes as well which is the main purpose of our method. Therefore, we claim that using Multiple generators we are able to generate more diverse and good quality samples.

Fig. 2. Images generated by Multiple generator GAN at 100th Epoch for MNIST dataset with 3 generators

Fig. 3 depicts samples taken from each generator at random iterations to show the difference in the quality of the samples more clearly. We observe that at least (n-1) generators are able to learn diverse modes in case of multiple generators. One

of the generator is always getting trapped in generating the noisy samples. Thus, we can observe that Increasing number of generators does help a bit with Mode collapse but somehow only (n-1) of them behaves as expected.

Fig. 4 and Fig. 5 illustrates the qualitative comparison of the samples generated by different models with our proposed model.

Fig. 6, Fig. 7, Fig. 8, and Fig. 9 show the training losses for MNIST dataset of different models with our proposed model.

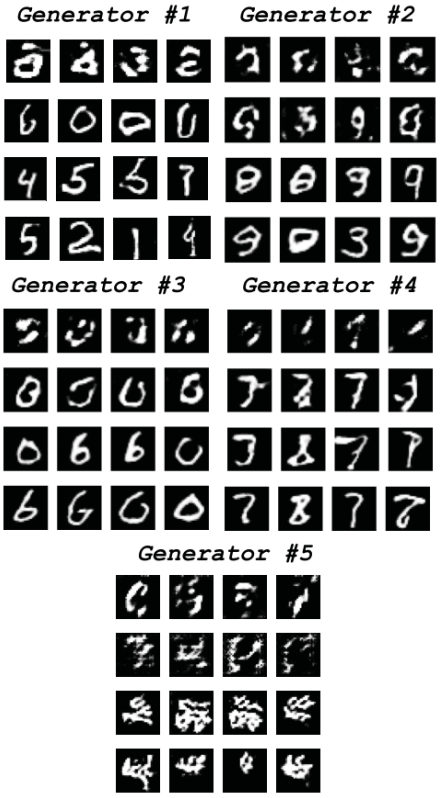

Fig. 3. Images generated by Multiple generator GAN for MNIST dataset with 5 generators. Top to bottom rows represents the samples generated at 1st, 20th, 50th and 100th epoch respectively by each generator.

Fig. 4. Images generated by DCGAN, WGAN and Multiple generator GAN respectively at 100th epoch for MNIST dataset. Generated samples by multiple generators approach are of better quality and less distorted as compared to WGAN and DCGAN.

Fig. 5. Images generated by DCGAN, WGAN and Multiple generator GAN respectively at 100th epoch for CelebA dataset. Training process is a bit resource intensive for our approach. Thus, The results presented here are from continuous runs which did not terminate prematurely. We hope to investigate and address this issue before running next set of experiments.

VI. CONCLUSION

In this work, we have proposed a method of using multiple generators which adapt the basic framework of DCGAN by changing the number of generators and slight modification to the Discriminator objective. Upon applying multiple generators to generate the different modes data samples, we have observed that it gives successful results in handling the problem of Mode collapse. For MNIST dataset we have got promising results using our approach.

One major issue we faced was not to have a proper idea of how many number of generators should be used so that all the mode classes of the training data distribution can be covered. As we look at MNIST dataset result, we have observed the scenario where using more number of generators produced better results in terms of capturing diverse modes of the distribution. Although, it gives successful results to handle the problem of mode collapse, only (n-1) generators are actually able to learn the modes. One of the generators keeps on generating noisy samples in each iteration. Thus, this remains one of the open questions and needs to be investigated further.

As for the development of future work is concerned, We can try the same approach with other loss function like Wasserstein metric. Some future extension is to improve the behaviour of the generator model to enforce the diversity between the generators where each generator should be able to capture specific modes. We can also try to push different generators to generate different modes and then to fuse the output of these generators to capture varied diverse modes of data distribution. As we have no prior information of the number of generators that needs to be used in the model, it would be better to know about the number of generators to be used before hand and how the different number of generators are affecting in data generation.

REFERENCES

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

[2] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[3] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

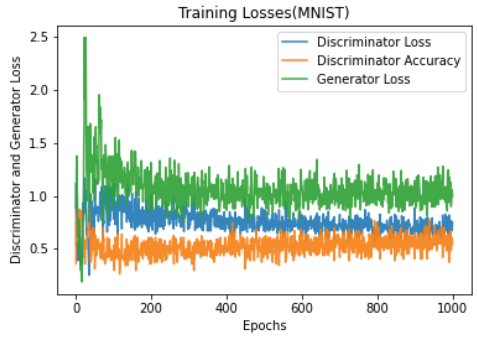

Fig. 6. DCGAN Loss

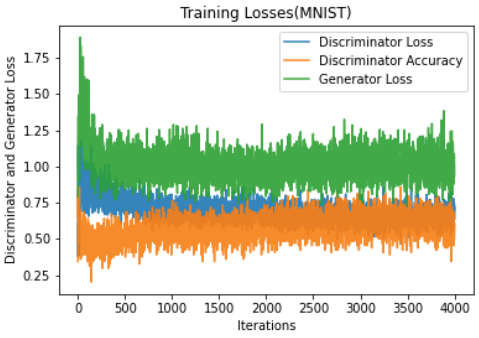

Fig. 7. WGAN Loss

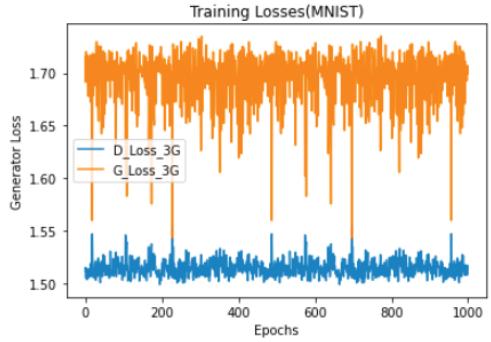

Fig. 8. Multiple generator loss for 3 generators

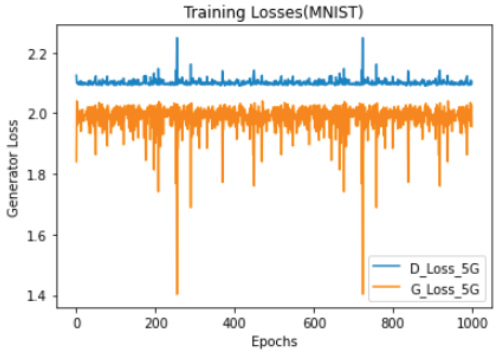

Fig. 9. Multiple generator loss for 5 generators

[4] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2536–2544.

[5] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," *arXiv preprint arXiv:1701.04862*, 2017.

[6] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li, "Mode regularized generative adversarial networks," *arXiv preprint arXiv:1612.02136*, 2016.

[7] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[8] Q. Hoang, T. D. Nguyen, T. Le, and D. Phung, "Mgan: Training generative adversarial nets with multiple generators," in *International Conference on Learning Representations*, 2018.

[9] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in neural information processing systems*, 2017, pp. 5767–5777.

[10] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[11] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," *arXiv preprint arXiv:1611.02163*, 2016.

[12] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Advances in neural information processing systems*, 2016, pp. 469–477.

[13] Y. LeCun, C. Cortes, and C. J. Burges, "The mnist database of handwritten digits, 1998," *URL http://yann. lecun. com/exdb/mnist*, vol. 10, no. 34, p. 14, 1998.

[14] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[15] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730–3738.

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

[17] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1, 2013, p. 3.

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