Adversarial Learning with Local Coordinate Coding (Paper Review)

Han Liang Wee Eric
School of Computing
National University of Singapore
eric_han@nus.edu.sg

Lee Wei Qing
School of Computing
National University of Singapore
e0032074@u.nus.edu

Abstract

Generative adversarial networks (GANs) is a machine learning system where two neural networks (generator and discriminator) are competing against each other. They try to exploit each other's weaknesses to win over time; the generator tries to make the input real and the discriminator tries to differentiate a real and generated image. In this way, both networks mutually improve over time. GANs have been successfully applied to many tasks such as motion prediction, 3D model reconstruction, image scaling and more. We have found the method proposed in 'Adversarial Learning with Local Coordinate Coding' to be interesting as existing GANs employ some prior distribution (Gaussian, uniform) and they propose a sampling method (Local Coordinate Coding) for GANs to capture local information. We will investigate the method through theory and experiments. We have found that LCC-GANs produce no better images than GAN while suffering from the same issues (ie. mode collapse) as GAN, even though the approach is interesting.

This review will attempt to illuminate and understand the key contributions of the paper by Jiezhang Cao [3], titled 'Adversarial Learning with Local Coordinate Coding'. It was recently published in ICML 2018, that generated some interest, 4 citations as of 11 April 2019. We will investigate the paper as per our proposal without any substantial change. This paper review is organized in the order given below.

1. **Motivation** - We will investigate and comment on the motivations given by the authors. In addition, we will present some further motivation and alternative approaches.
2. **Introduction** - We will summarize the work done in the paper, presenting on the necessary definitions and theorems needed as background for the reader to understand the paper. Additionally, We will discuss and comment on key ideas.
3. **Key contributions** - We will discuss the contributions and significance of the paper. We will highlight the claims by the author that investigate in the following section. We will present both mathematical derivations and experiments to understand the paper in a deep manner.
4. **Our Investigation** - We will organize and investigate the claims mentioned, studying them both theoretically and experimentally.
5. **Conclusion** - We will summarize the paper review and conclude on the results presented.

1 Motivation

Firstly, the authors motivated their work on GAN by referring to GAN's successful applications: video prediction [19, 17] and image translation[10, 7]. We augment this list by adding more impressive use cases of GANs: the popular Deepfake software (based on [14]), allowing people to swap faces

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

out of videos. The technique is so good that it is almost believable, most notable is the Deepfake Obama Public Service Announcement (See Figure 1) and Deepfake porn so real that it is plaguing the internet. Another interesting example is Pose Guided Person Image Generation[15] which allows swapping of clothing articles given an image of a particular pose (see Figure 2).

Figure 1: Deepfake Obama screenshot

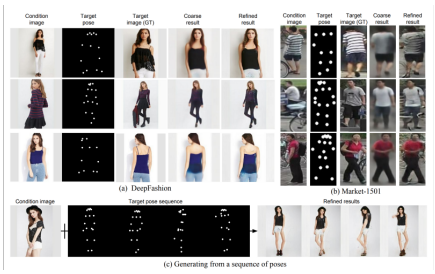

Figure 2: Pose Guided Person Image Generation

Next, the authors identified 2 limitations of GANs, which motivate their work with improves upon the current state-of-the-art GANs:

1. GAN employs simple priors[5], often independent of data distribution, resulting in images with distorted structures from the lack of semantic information in the latent space. He gave many other examples such as DC-GANs[18], WGANs[1] and PGANs[8]. We have investigated his argument and found that it is lacking discussion on disentangled representation[6], which seeks to disentangle the latent space into disjoint parts: salient features and presentation features. The result is a very interpretable latent space full of semantic information, as demonstrated by InfoGAN[4] which uses information theory to optimize the disentanglement. For instance, InfoGAN is capable of identifying semantically meaningful latent variables even though it assumes a prior. Hence, we opine his question should be pivoted to would be can dependence on the data distribution enhance interpretability in the latent space.
2. GAN's performance is sensitive to the latent distribution (dimensionality and quality - relating to data distribution), which is not used by most generative models. The author gave examples of several generative models that does account for the posterior distribution: VAE[11], WAE[21], AAE[16]. We have studied the literature, updating the list with FactorVAE[9], and have found this limitation to be reasonable. It is indeed difficult to sample from posterior, due to lack of a generation mechanism of the autoencoder.

2 Introduction

This paper relies on the key insight that *high dimensional data often lie on some low dimensional latent manifold*. The paper is heavily based on the assumption that the low dimensional latent manifold can be approximated using local coordinate coding (LCC) which Yu formulated[22]. LCC coding relies on Yu's non-typical definition of Lipschitz Smoothness, as shown below in Definition 2.1. We will also state the well-accepted definitions[20], taken from SSBD, of Lipschitzness and Lipschitz smoothness in Definitions A.1 and A.2 respectively. In our discussions, the distance measure will refer to be the Neural Network Distance [2] as defined by Arora.

Definition 2.1 (Yu's Lipschitz Smoothness)**.** $f(x)$ on $\mathbb{R}^d$ is (α, β, ρ)-Lipschitz smooth if $|f(x') - f(x)| \leq \alpha\|x - x'\|$ and $|f(x') - f(x) - \nabla f(x)^\top (x' - x)| \leq \beta\|x - x'\|^{1+p}$, $p \in (0, 1]$, $\alpha, \beta > 0$

The authors' proposed method is summarised into 3 distinct steps. The objective of the steps would be to learn a set of bases, through LCC, that the GAN can use which will minimize the distance between the generated and empirical distributions.

1. **Use AutoEncoder** - Learn latent manifold using a typical AutoEncoder
2. **Train LCC** - Learn a set of bases over the latent manifold
3. **Train GAN** - Learn a typical GAN, using the bases in the generator

3 Key Contributions

The author contributed, analyzed and tested LCC-GAN, and concluded with the following claims:

1. LCC-GAN generates perceptually convincing images by exploiting local information. The author performed a qualitative comparison of the images generated by LCC-GAN with others.
2. LCC-GAN has superior performance over several state-of-the-art methods. The author compared LCC-GAN with others using Inception-Score quantitatively.
3. LCC-GAN needs small dimensional latent space for good generalization performance. The author gave an empirical generalization error (GE) bound for LCC-GAN (Thm 3.1).
4. Able to draw meaningful samples from the latent space. The author gave many random draws of data from the latent space and displayed the images and give a qualitative argument.

Theorem 3.1 (Empirical GE Bound)**.** *With probability* $1-\delta$*, generalization error* $\epsilon \leq 2\mathcal{R}_{\mathcal{X}}(\mathcal{F}) + 2\Delta\sqrt{2/N\log[1/\delta]} + 2E(d_m)$*, where* $\mathcal{R}_{\mathcal{X}}(\mathcal{F})$ *is the Rademarcher complexity of* $\mathcal{F}$*, upper bound for the loss function* Δ*, number of training examples* N*,* $E(d_m)$ *is an error term given in Theorem A.4.*

We note that LCC-GAN relies on the underlying assumptions made in defining the LCC scheme (ie. Def 2.1). The results and claims look impressive, we investigate the claims both theoretically and experimentally to determine if the results and claims hold up.

4 Our Investigation

Referring to the same claims mentioned in the previous section, we device investigation plans:

1. Perform a blind survey on images generated from LCC-GAN, GAN, and real images. We want to determine if a person can identify the real images and also which identifying the method (LCC-GAN/GAN) gives more realistic images.
2. Due to time constraints, we have decided not to validate this claim.
3. Investigate the LCC assumptions and further analysis of the generalization bound.
4. Perform vector arithmetic on the LCC latent space, as this is an indicator of semantics.

4.1 Blind Survey

We use the CIFAR-10 dataset (different from paper) with LCC-GAN and GAN implementations (trained similarly). We first produce the image sets, each consist of 8x8 randomly selected/generated images. Then, we performed a blind survey on 30 unique image set pairings of the following composition: 10 LCC-GAN vs GAN, 10 GAN vs real (from the dataset), 10 LCC-GAN vs real. Each human was asked to choose (for 30 pairings) the image set that looks most real for every 2-image pairing. The human is blind to the method used to generate/select the image set, we have taken steps to randomize and equalize the process to ensure that there are no markings to help the human identity the real images. Please refer to the Appendix for more details. We set up the survey in this manner to:

1. Perform some sort of Image Turing Test, to see if a human can identify the real image (for LCC-GAN vs real and GAN vs real). According to the author's claims, we would expect the percentage of LCC-GAN chosen over real images be higher than that of GAN. This would roughly translate to images produced from LCC-GAN to be more indistinguishable from real images than images produced by GAN.
2. Determine if LCC-GAN or GAN generates images set that is most real to the human. Similarly, according to the author's claims, we will expect that more humans choosing LCC-GAN compared with GAN.

We have 21 fully completed and 37 partially completed unique submissions after disqualifying some spam/repeat/invalid submissions. We will also use the results from partially completed surveys as the sequence of pairings presented to the human is random and inclusion of the data has no negative consequences (ie. skew). We note that our participants are largely from Singapore, but there should

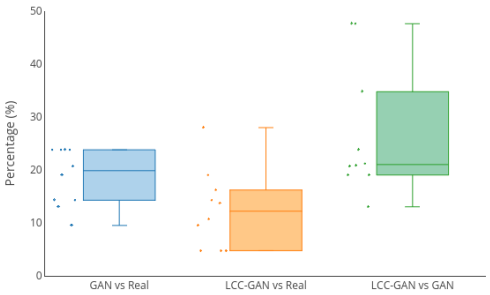

Figure 3: Blind Survey Results

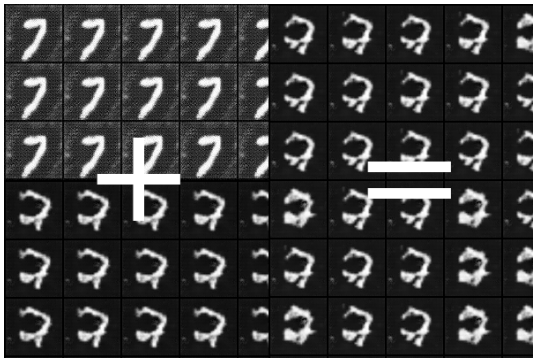

Figure 4: Vector addition of 7 and 2

be no biases associated with Singapore. The results are presented in a box plot in Figure 3, with the percentages of humans choosing GANs (GAN vs Real), LCC-GANs (LCC-GAN vs Real) and LCC-GANs (LCC-GAN vs GAN) respectively. We analyze according to our 2 purposes:

1. It is obvious that both GAN and LCC-GAN failed (statistically significant) to fool humans. We note that the distribution of LCC-GAN vs Real is almost entirely lower than GAN vs Real. Hence, we conclude that images produced from GAN are more indistinguishable to humans than images produced by LCC-GAN.
2. It is also obvious from the figure that images produced by LCC-GAN look less real than images produced by GAN. Specifically, the average percentage of instances where humans chose images produced by LCC-GAN over GAN is $(26.8 \pm 20.8)\%$. We note that there were 2 cases where images from LCC-GANs came close. However, these cases are still $< 50\%$ and are outliers as LCC-GAN consistently produce lower quality images than GAN.

Hence, we conclude that LCC-GAN does not generate perceptually convincing images, even producing worse images than GAN. We note that there may be better hyperparameters that can potentially generate more realistic images, so more investigation is needed.

4.2 LCC assumptions

We first note that the typical definition of Lipschitz Smoothness as shown in Def A.2 requires the $\nabla f(x)$ to be β-Lipschitz. We contrast that to the definition of Lipschitz Smoothness by Yu shown in Def 2.1, which does not require the gradient function to be Lipschitz. We that Yu's definition is equivalent to the typical Lipschitz Smoothness definition under certain conditions.

Lemma 4.1 (Lipschitz Smoothness Equivalence)**.** *Definition 2.1 is equivalent to Definition A.2 when f is convex and ∇f is 2β-Lipschitz when $p = 1$. We will prove this below.*

Proof. From Definition 2.1: $\nabla f(x')^\top (x - x') - \nabla f(x)^\top (x - x') \leq 2\beta \|x - x'\|^{1+p}$

$\iff \big(\nabla f(x') - \nabla f(x)\big)^\top (x - x') \leq 2\beta \|x - x'\|^{1+p}$

$\overset{(A.6)}{\iff} \big(\nabla f(x) - \nabla f(x')\big)^\top (x - x') \leq \|\nabla f(x) - \nabla f(x')\| \|x - x'\|$

$\overset{(A.5)}{\iff} \frac{1}{2\beta} \|\nabla f(x') - \nabla f(x)\|^{1+p} \leq \|\nabla f(x) - \nabla f(x')\| \|x - x'\|$

$\iff \|\nabla f(x') - \nabla f(x)\|^p \leq 2\beta \|x - x'\| \overset{p=1}{\iff} \|\nabla f(x') - \nabla f(x)\| \leq 2\beta \|x - x'\|$ □

The consequneces of Lemma 4.1 is that the assumptions needed for smoothness are relaxed. The $f(x)$ does not need to be 2β-Lipschitz but almost 2β-Lipschitz, controlled by the parameter p. In Yu's paper[22], the relaxed assumption allowed the authors to linearize the arbitary coodinate coding, allowing a localization measure and subsequently a quality measure (Definition A.3) which can be optimized. In conclusion, the choice of α, β, p is important to bound the Lipschitz behavior to allow the LCC to work. With a poor choice of α, β, p, the LCC and its implications would break down.

4.3 Generalization Bound

We further analyze the empirical GE bound given by the author in Theorem 3.1, by deriving a complexity measure.

Theorem 4.2 (Empirical GE Complexity)**.** *The empirical GE is bounded by* $\epsilon = O(\mathcal{R}_{\mathcal{X}}(\mathcal{F}) + \sqrt{1/N} + \sqrt{d_m})$ *where* $\mathcal{R}_{\mathcal{X}}(\mathcal{F})$ *is the Rademarcher complexity of discriminator,* N *is the number of training examples and* d_m *is the dimentionality of the latent space.*

Proof. From Theorem 3.1: $\epsilon \leq 2\mathcal{R}_{\mathcal{X}}(\mathcal{F}) + 2\Delta\sqrt{2/N \log[1/\delta]} + 2E(d_m)$
$\overset{(A.4)}{\iff} \epsilon \leq 2\mathcal{R}_{\mathcal{X}}(\mathcal{F}) + 2\Delta\sqrt{2/N \log[1/\delta]} + 2(L_\phi Q_{L_h, L_G}(\gamma, C) + 2\Delta)$
$\overset{(A.7)}{\iff} \epsilon = O(\mathcal{R}_{\mathcal{X}}(\mathcal{F}) + \sqrt{1/N} + 2(L_h c_M + (1 + \sqrt{d_m} + 4\sqrt{d_m} L_G)]\epsilon'^2)$
$\iff \epsilon = O(\mathcal{R}_{\mathcal{X}}(\mathcal{F}) + \sqrt{1/N} + \sqrt{d_m})$ □

The consequence of Theorem 3.1 is that increasing the number of samples, decreasing the predictive power of the discriminator and decreasing the dimensionality of the latent space would improve generalization. We note that the cost of such a change is that the predictive power will be reduced. We will assume the same number of samples and perform experiments with various values to the hyper-parameters 'ndf' (number of features for discriminator, which approximate to the number of functions of the discriminator) and 'nz' (dimensionality of the latent space).
We performed the experiment using the MNIST dataset, drawing the images at the 70th iteration, leaving every other setting the same. The results are shown in Figure 7. We note that the LCC-GAN began to exhibit signs of mode collapse, poorer strokes and confused digits as the nz (less affected) and ndf (more affected) increases. Note that the image for ndf=64, nz=16 is greying out. These experiments support the result in Theorem 4.2, we note that the model increasingly failed to generalize as ndf and nz increases. In addition, the experiments match the theory that the generalization is more affected by $\mathcal{R}_{\mathcal{X}}(\mathcal{F})$ (represented by ndf) than $\sqrt{d_m}$ (represented by nz).

4.4 Vector Arithmetic

We also analyze the latent space and perform vector arithmetic on vectors drawn from the latent space. This is a common experiment known as vector arithmetic in the latent space done in GAN papers[4] to discover if the latent space has some semantics/interpretability. We have attempted to produce pairs of latent vectors that allow us to produce something meaningful when added, but we are not successful. All of our attempts look similar to Figure 4, where it tries to add the latent vectors for 2 and 9, but it produced some garbage. We conclude that the latent space does not have much semantic meaning, but note that we need more experiments to analyze the semantics of the latent space.

5 Conclusion

In summary, we have studied LCC-GANs, GAN, etc and gave some other motivations in the earlier section. We stated the problem from LCC-GAN and gave definitions of LCC and the generalization error to aid our discussions. We investigated the theory of LCC and the GE bound, understanding the assumptions for which the model hold and deriving new insights/consequences. We gave an example of LCC-GAN working on a new dataset (CIFAR-10) and conducted a survey to validate the author's claims of convincing images and found that LCC-GAN produces lower quality images than GAN. From GE bound, we performed an experiment to validate the GE bound experimentally. Lastly, we also tried to perform vector arithmetic and was unsuccessful. In conclusion, we have spent a considerable amount of time to fix the author's original code. We have also spent much time trying to overcome LCC-GAN's non-robust nature. We think that this is due to GANs being very difficult to train and also mode collapses when the generator produces the same kind type of images as the generator thinks that it has a higher chance of fooling the discriminator. We noticed that LCC-GAN suffers from mode collapse as GANs. A tell-tale sign is that it starts to produce a limited diversity of samples. Specifically, it will produce many instances of the digit 7 as it is very similar to 1. The method used is interesting, especially if it can help to improve the robustness of LCC-GAN through data prior. We think maybe an applying a parameter search for α, β, p during training might be useful to help LCC-GAN to learn the latent space. We opine that more testing is needed, it does not seem that LCC-GAN is better than GAN.

Appendix A Supplementary Equations

Definition A.1 (Lipschitzness)**.** Let $C \subset \mathbb{R}^d$. A function $f : \mathbb{R}^d \to \mathbb{R}^k$ is a ρ-Lipschitz over C if for every $x, x' \in C$ we have $\|f(x') - f(x)\| \leq \rho\|x' - x\|$

Definition A.2 (Lipschitz Smoothness)**.** A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is β-smooth if its gradient is β-Lipschitz; for every x, x' we have $\|\nabla f(x') - \nabla f(x)\| \leq \beta\|x' - x\|$

Definition A.3 (Yu's Localization Measure)**.** α, β, p are tunable parameters. γ is the coding. C is the set of basis vectors.

$$Q_{\alpha,\beta,p}(\gamma, C) = \mathbb{E}\left[\alpha\|s - \gamma(x)\| + \beta \sum_{v \in C} |\gamma_v(x)|\|v - \gamma(x)\|^{1+p}\right]$$

Theorem A.4 (GE Bound over distribution)**.** *Suppose $\phi'(.) \leq L_\phi$ and bounded in $[-\Delta, \Delta]$. Given the coordindate coding (γ, C) then the generalization bound is $\epsilon \leq E(d_m) = L_\phi Q_{L_h,L_G}(\gamma, C) + 2\Delta$*

Lemma A.5 (Co-coercivity of gradient)**.** *f is convex and ∇f is L-Lipschitz, then $\forall_{x,y}$:* $1/L\|\nabla f(x) - \nabla f(y)\|^2 \leq \big(\nabla f(x) - \nabla f(y)\big)^\top (x - y)$.

Lemma A.6 (Cauchy–Schwarz inequality)**.** $\forall_{u,v} : |\langle u, v\rangle| \leq \|u\|\|v\|$

Lemma A.7 (Yu's Manifold Coding)**.** $Q_{L_h,L_G}(\gamma, C) \leq [L_h c_M + (1 + \sqrt{d_m} + 4\sqrt{d_m}L_G)]\epsilon'^2$

Appendix B Experimental Setup

We will be using Google Cloud Platform (GCP) as it is easy to use, requires less setup and they give free credits. Most importantly, GCP allows easy deployment of Deep Learning VM with a pre-packaged pre-configured OS and software needed for Deep Learning.

B.1 Hardware

We will be using a VM equipped with NVIDIA® Graphics Processing Unit. We have chosen a GPU accelerated VM as our compute workload (neural network) can benefit from GPU (Nvidia) acceleration. We used a variety of different GPUs, but that will not have any impact on our experiments as we are not measuring time or speed performance of the models. We used a different GPU as our needs require. The variety of GPUs used are as follows:

- NVIDIA Tesla K80 - slowest, but very cheap
- NVIDIA Tesla P100 - fast, moderately priced
- NVIDIA Tesla V100 - very fast, but expensive

We note that all CPU and Memory configurations are typical, we do not use any special settings.

B.2 Software

For all our experiments, we will be running them on Debian 9 operating system as per VM image. Our code implementations will be all using Python 2.7, and Pytorch and Tensorflow libraries.

Our LCC-GAN and GAN implementations are adapted from the repositories guoyongcs/LCCGAN and csinva/pytorch_gan_pretrained. We note that the LCCGAN repository is very buggy and we end up going back and forth, contacting the author and fixing a lot of the bugs and inefficiencies in the original code. Our code repositories are made available online:

- Repositoy containing our results and report
- Our implementation of LCC-GAN that saves the model
- Our implementation of LCC-GAN, bug-fixed, improved and optimized

B.3 Datasets

We only used widely avaliable datasets, which are well used in literature, from reputable sources:

- MNIST[13] dataset from Pytorch
- CIFAR-10[12] dataset from Pytorch

Appendix C Survey

Refer to the Repository for the images used for the survey and the script used to randomize. We have opted for an online survey, with the choice of platform to be Survey Legend, as they have mechanisms to ensure that spam/repeat submissions can be monitored, tracked and removed easily. Our survey can be found here - Can the computer fool you?.

C.1 Screenshot

A screenshot of the survey is given below.

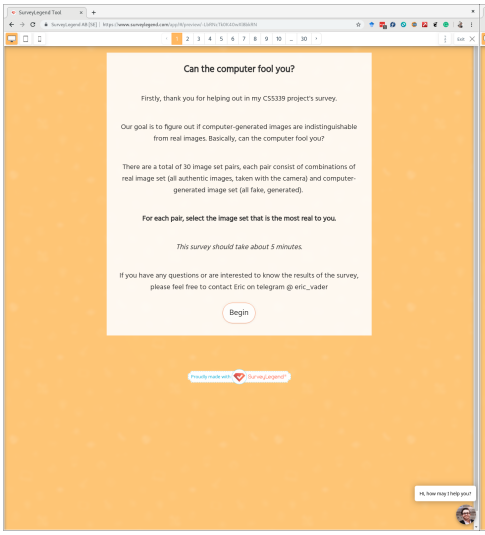

Figure 5: Screenshot of Start Survey

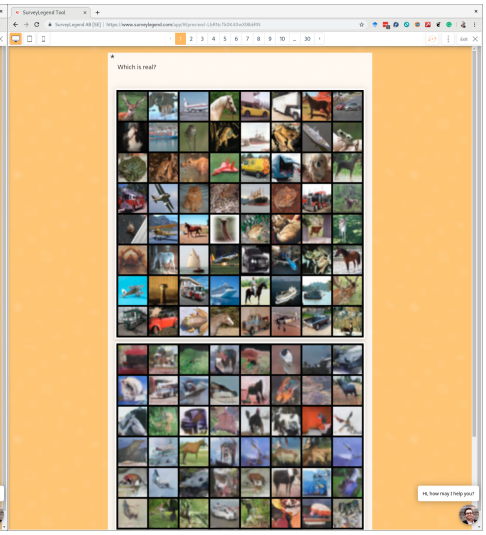

Figure 6: Screenshot of a question

C.2 Results

Completion statistics:

- 21 Fully completed
- 37 Partially completed
- 3 Voided due to spam

GAN vs Real	LCC-GAN vs GAN	LCC-GAN vs Real
9.523809524	4.761904762	13.04347826
13.04347826	4.761904762	19.04761905
14.28571429	4.761904762	19.04761905
14.28571429	9.523809524	20.68965517
19.04761905	10.71428571	20.83333333
20.68965517	13.7254902	21.21212121
23.80952381	14.28571429	23.80952381
23.80952381	16.21621622	34.7826087
23.80952381	19.04761905	47.61904762
23.80952381	28	47.61904762

Table 1: Raw results from survey, in percentages

Appendix D Generalization Bound

We performed the experiments using the MNIST dataset. We fixed all parameters except ndf and nz. The generated images are drawn from the model at the 70th iteration.

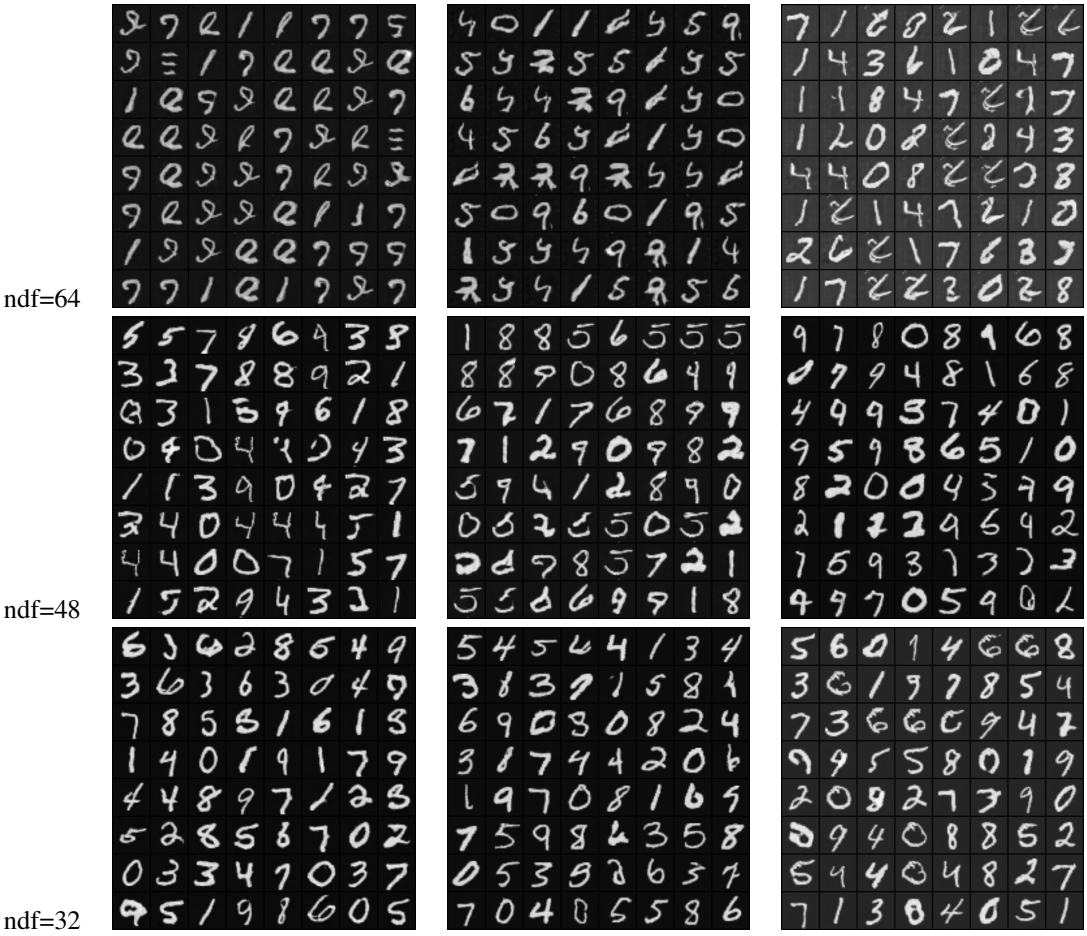

Figure 7: Generated Images from varying 'ndf' and 'nz' at 70th iteration

References

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

[2] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang. Generalization and equilibrium in generative adversarial nets (gans). In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 224–232. JMLR. org, 2017.

[3] J. Cao, Y. Guo, Q. Wu, C. Shen, J. Huang, and M. Tan. Adversarial learning with local coordinate coding. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 707–715, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[4] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.

[5] 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*, pages 2672–2680, 2014.

[6] I. Higgins, D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, and A. Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

[7] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

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

[9] H. Kim and A. Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

[10] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1857–1865. JMLR. org, 2017.

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

[12] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[13] Y. LeCun and C. Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

[14] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, pages 700–708, 2017.

[15] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool. Pose guided person image generation. In *Advances in Neural Information Processing Systems*, pages 406–416, 2017.

[16] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[17] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

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

[19] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014.

[20] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[21] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

[22] K. Yu, T. Zhang, and Y. Gong. Nonlinear learning using local coordinate coding. In *Advances in neural information processing systems*, pages 2223–2231, 2009.