GANs Optimizing KL Divergence Gain Resistance To Mode Collapse

Houze Liu, Songyao Jiang
Department of Engineering
Northeastern University
360 Huntington Ave, Boston, MA 02115
{liu.hou, jiang.so}@husky.neu.edu

Abstract

In this paper, we aim to tackle Mode Collapse problem in GANs, and report two methods: 1.a new loss function that approximates Jenson-Shannon Weighted divergence. The new loss function enables us switching GANs' objective between KL divergence and reverse KL divergence[5, 11] smoothly through a changeable parameter; 2.adding a regularization term which allows minimizing KL Divergence between real and generator distribution during updating so that generators can directly learn from real data to cover whole modes. We present crude yet straightforward analysis to both methods. Our experiments demonstrate the effectiveness of both methods.

1 Introduction

Generative Adversarial Network[5](GAN) is one kind of power generative model and has achieved huge success. Different from Markov Chain based generative models, GANs need no explicit density function; instead, they directly transform one distribution to another. In practice, sampling is achieved by drawing a noise vector from Gaussian or uniform distribution and feed it to a neural network, which is called a generator. There is another neural network called discriminator trained to label real and fake samples correctly. The objective of the generator is to fool the discriminator into making mistakes. In [5], updating generator can be seen as optimizing Jenson Shannon Divergence(JSD) between real distribution and generator distribution, which reaches minimum when the two distributions are equal. [6, 15] show that optimizing JSD results in similar to optimizing reverse KL Divergence, hence lead to higher sample quality and poor mode coverage. However, GANs have Mode Collapse problem: instead of covering whole distribution, the generator only produce samples at single or few points. This problem severely limit GANs application in real world. And optimizing KL Divergence as Maximum Likelihood Methods usually lead to better mode coverage. Hence properly introducing KL Divergence[11, 3] to GANs objective is a nature way to help them resist mode coverage.

In this paper, we propose two methods to introduce KL Divergence to GANs' objective: 1. we add a weight parameter to original GAN loss function which theoretically enables GANs to be optimized either more towards KL divergence or Reverse KL Divergence by changing the weight parameter. Our experiments show that by changing the parameter, we can put more weights on KL Divergence resulting trained models being more resistant to Mode Collapse. The motivation is to demonstrate that optimization over KL Divergence makes GAN more resistant to Mode Collapse; 2. inspired by 1., we introduce a regularization term to the generator's objective which directly allow it minimize KL Divergence between

Preprint. Under review.

real and generator distributions. GANs trained with our Method2 are resistant to Mode Collapse while keep the ability of generating sharp and realistic samples. We experiment our method on toy data set and image generation tasks. Our experiments demonstrate that our method successfully helps GAN cover more modes and generate more diverse samples, while we don't observe decrease in sample quality.

2 Background

2.1 Generative Adversarial Network

Training a GAN can be seen as making two models, a discriminator and generator, play a two-player zero-sum game:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x\sim P_r} \log(D(x)) + \mathbb{E}_{x\sim P_G} \log(1 - D(x))$$

Where P_r is data distribution and P_G is a generator distribution with trainable parameters. The game converges to a Nash Equilibrium through iteratively updating generator and discriminator given assumption that at every step the discriminator is optimal. [5] proves that under optimal discriminator assumption, training generator can be approximated as minimizing Jenson-Shannon divergence:

$$C(G) = -\log(4) + 2 \cdot JSD(P_r||P_G)$$

Where:

$$JSD(P_r||P_G) = 0.5 \cdot KL(P_r||0.5 \cdot (P_r + P_G)) + 0.5 \cdot KL(P_G||0.5 \cdot (P_r + P_G))$$

2.2 KL Divergence And Reverse KL Divergence

There have been many literature about difference between optimizing over KL and Reverse KL[15, 10]. In short, optimizing over KL divergence leads to merging operation, which approximates a pair of components in the same mixture density with a single component of the same type. This usually encourage distribution to cover wider range and larger number of modes; On the contrary, optimizing over Reverse KL divergence will cause prune operation, which reduces the number of components in a mixture distribution. This allows distribution to concentrate on few or single mode. As in Figure 1, optimizing over JSD is similar to Reverse KL divergence hence partly explain the Mode Collapse phenomenon where GANs only cover few or single mode. If this is caused by prune operation, then introducing merging operation by adding KL could be a potential remedy.

2.3 Related Works

[12] generalizes divergence to f-divergence; [1] uses Wasserstein distance; [8]uses Pearson-χ divergence; [11] uses 2 discriminator, one for KL and another for rKL; [13] uses α-divergences for KL and rKL; [12] has mentioned a loss function for Jenson-Shanon Weighted divergence[6] in Supplementary Material GAN 3. [6] mentions the potential shifting property of general JS divergence. [7, 14, 2, 16] and others.

2.4 Recently proposed methods for solving mode collapse

There have been many works tackling Mode Collapse in GANs from different perspectives. [9] uses a surrogate loss which reveals more discriminator steps to generator to prevent the generator from putting the whole mass on any single point. [11] applies the idea of using two discriminators, one for KL Divergence and another for reverse KL Divergence. [4] uses multiple generators, each of which deals with some modes out of all. [3] uses a teacher model called Estimator to include optimizing KL Divergence into GAN objective. From above work, we see that it has been presumed that KL Divergence is related to mitigating Mode Collapse problem in GANs. Though optimization over KL Divergence is generally reasonable, as mentioned above for simple mixture Gaussian with only two dimensions,

it is still not clear that whether its effectiveness generalizes well to GANs setting, where high dimensional inputs are considered. Also, many methods presented in the works above introduce extra parameters, like using an auxiliary model, or couldn't demonstrate separately if the effectiveness reported in the paper is solely due to KL Divergence or other techniques used together. In order to study the separated effectiveness of KL Divergence in the context of GANs, we use a modified version of loss function to train them(Method1). Our experiments show that solely optimizing GANs towards KL can help them gain resistance to Modes Collapse. Then we push this idea to another direction: keeping discriminator loss function unchanged, and adding a regularization term to generators' loss function, which allows generators to directly minimize KL Divergence between generator distribution and real distribution. This means that now the generator can get gradients both from the discriminator and directly from real data. Our method is the implementation of idea in [16]. We present it in Method2. We conducted experiments to empirically demonstrate its effectiveness.

3 Method 1: Adding A Weight Parameter To Loss Function

We propose a modification of original objective:

$$\min_G \max_D V(G,D) = \pi \mathbb{E}_{x\sim P_r} \log(\frac{1}{\pi} D(x)) + (1-\pi)\mathbb{E}_{x\sim P_G} \log(\frac{1}{1-\pi}(1-D(x)))$$

Where $\pi \in (0,1)$ is a extra parameter that allows GAN to allocate weights between its two terms. By applying the technique in [5], it's straightforward to get:

Proposition 1 *Given a fixed G, minimizing $V(G,D)$ gives following optimal discriminator D^*:*

$$D^* = \frac{\pi P_r}{\pi P_r + (1-\pi)P_G}$$

Theorem 2 *The objective of optimizing G according to this optimal D^* is:*

$$\begin{aligned} C(G) &= \pi KL(P_r||\pi P_r + (1-\pi)P_G) + (1-\pi)KL(P_G||\pi P_r + (1-\pi)P_G) \\ &= JSD_\pi(P_r||P_G) \end{aligned}$$

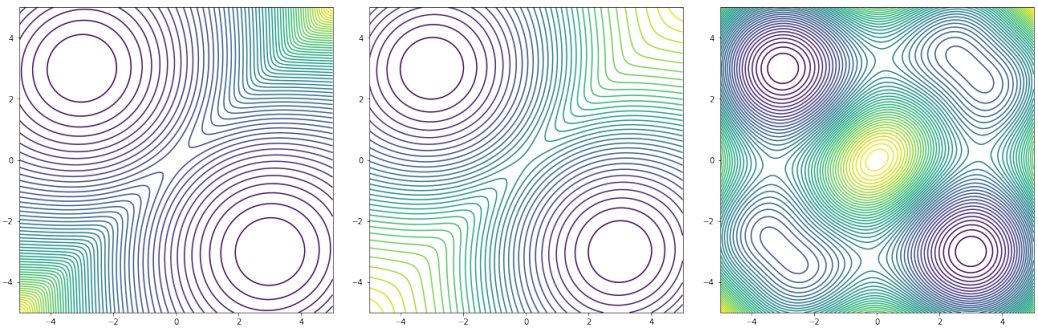
Same as original objective, given the optimal discriminator, training generator can be seen as reducing Jensen-Shannon Divergence Weighted(JSDW)[6] between data distribution P_r and generator distribution P_G; The minimum value 0 is attained if and only if $P_r = P_G$ What's more, by changing $\pi \in (0,1)$, we can assign more weights on P_r if π gets closer to 1 or more weights on P_G if 0, which should make the behavior of P_G more like one optimizing over forward KL or reverse KL[6]. If setting $\pi = 0.5$ the objective is JSD and can be seen as that of vanilla GAN.

3.1 The Effectiveness Of Changing π

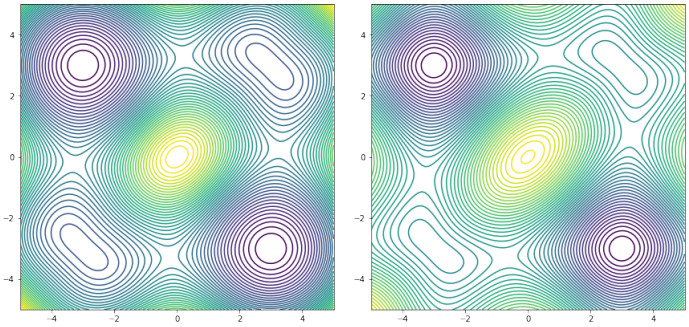
To see why changing π can make objective switching between KL and reverse KL, we conduct a small experiment on fitting a mixture Gaussian distribution of two components. We are going to use terminology in [15]; Namely, forward KL-divergence(FKLD) is $D_{KL}(P_r||P_G)$ and has property called mode merging operation that leads to covering whole modes; Reverse KL-divergence(RKLD) is $D_{KL}(P_G||P_r)$ and has property called pruning operation that leads to concentrate on certain mode.

Proposition 3 *If π is close to 0, $C(G)$ has merging operation similar to FKLD; if π is close to 1, $C(G)$ has pruning operation similar to RKLD.*

This is demonstrated in Figure 1.



(a) Contour of KL (b) Contour of JSW;$\pi = 0.000001$ (c) Contour of Reverse KL



(d) Contour of JSW;$\pi = 0.999999$ (e) Contour of JSD

Figure 1: Illustration of different divergences when used to fit model. The original data is a mixture Gaussian distribution: $p(x) = 0.5N(-3, 1) + 0.5N(3, 1)$; The reference distribution is also a mixture Gaussian distribution that has density: $q(x) = 0.5N(\theta_1, 1) + 0.5N(\theta_2, 1)$. The contour line represents numeric value of divergence, e.g. $D_{KL}(p||q)$. The x axis and y axis are θ_1 and θ_2 respectively. For example, when (θ_1, θ_2) takes value: (-3, 3), $D_{KL}(p||q) = 0$, which is one of the global minimum in (a). KL and JSW($\pi = 0.000001$) have similar contour while Reverse KL, JSW($\pi = 0.999999$) and JSD have similar contour. In this example, pruning operation is that parameters are (3, 3) or (-3, -3) where Reverse KL alike contours reach local minimum. Notice that JSD also has those local minimum areas, which might be related to Mode Collapse in GANs on high dimensional data like images.

4 Method2: Adding Regularization Term To Minimize KL Divergence

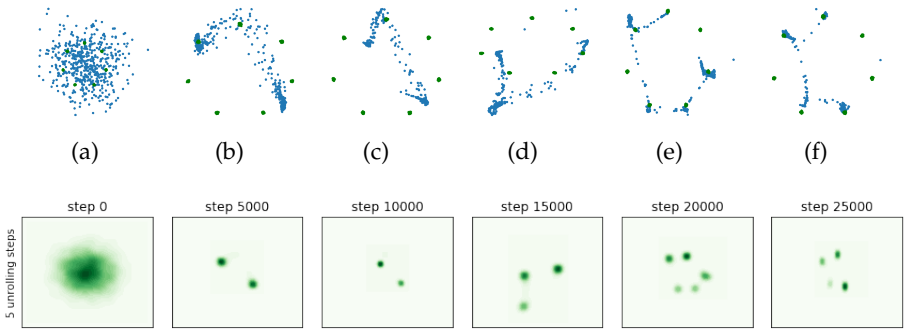
The motivation for Method2 is that Method1 may degenerate sample quality. Hence, we consider applying a loss function similar to reconstruction loss function used in VAE to match generated samples and real data directly. The loss function we use in our paper is cross entropy. Optimization can be seen as to KL divergence:

$$\min_{\theta} KL(P_{data}(x)||P_{model}(x;\theta)) <=> \min_{\theta} \mathbb{H}[P_{data}(x)||P_{model}(x;\theta)]$$

Hence, by reducing cross entropy between generator distribution real distribution, we actually reduce KL divergence between the two. The minimum is reached when two distributions are equal, hence it's obvious that adding the term won't change convergence property.

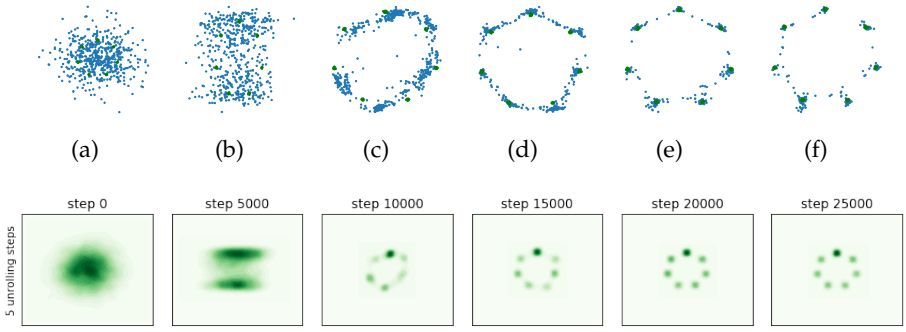
5 Experiments

In this section, we conduct comprehensive experiments to demonstrate the capability of improving mode coverage and the scalability of our proposed model on real world datasets. We use a synthetic 2D dataset for both visual and numerical verification, and three datasets



(g) corresponding map

Figure 2: Vanilla GAn failed to capture all modes



(g) corresponding map

Figure 3: Method 1: Adding weight parameter $\pi = 0.33$ helps GANs successfully cover all modes

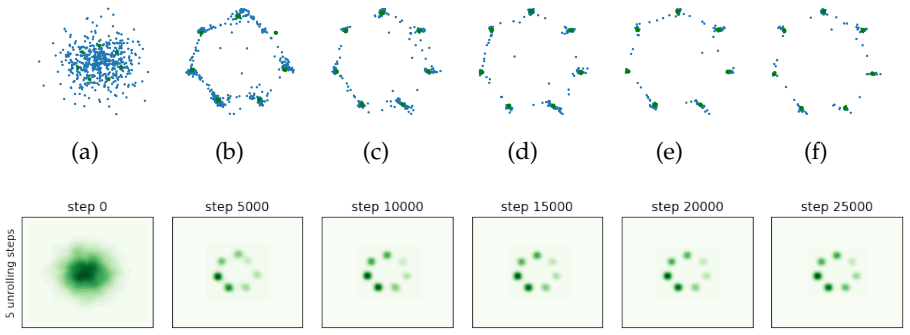
for numerical verification. We have made our best effort to compare the results of our method with those of the latest state-of-the-art GAN's variants by replicating experimental settings in the original work whenever possible.

5.1 Synthetic Data

Mode collapse can be accurately measured on synthetic datasets, since the true distribution and its modes are known. There are two data sets of increasing difficulty: a mixture of eight 2D Gaussian distributions arranged in a ring, a mixture of 100 2D Gaussian distributions arranged in 10x10 grid. For Method 1, we demonstrate its effectiveness on 2D Ring data; For Method 2, we compare all four competing methods on 2D Ring data and 2D Grid data.

5.2 Stacked Mnist

Following, we evaluate our methods on the stacked MNIST dataset, a variant of the MNIST data specifically designed to increase the number of discrete modes. The data is synthesized by stacking three randomly sampled MNIST digits along the color channel resulting in a 28x28x3 image. We now expect 1000 modes in this data set, corresponding to the number of possible triples of digits. Instead of manually count modes, we use a pretrained classifier of 99.75 test accuracy on MNIST data to predict the modes generated by our GAN model. And we also calculate KL divergence between predicted label distribution and true label distribution in MNIST data set. Our results are reported in Table 1 and Figure 6. To focus the evaluation on the difference in the learning algorithms, we use the same generator



(g) corresponding map

Figure 4: Method 2: Directly adding KL Divergence to generator objective helps GANs cover all modes

Table 1: Method 2: Quantitative Results On Stacked MNIST Data

Model	Mode Coverage(max 1000)	KL
DCGAN	99	3.4
ALI	16.0	5.4
Unrolled GAN	48.7	4.32
VEEGAN	150.0	2.95
PacDCGAN2	1000.0	0.06
DCGAN(our implementation)	25.84	4.81
Our Method 2	187.6	3.7

architecture for our implementations. In particular, the generator architecture is an off-the-shelf standard implementation of DCGAN.

5.3 CIFAR-10

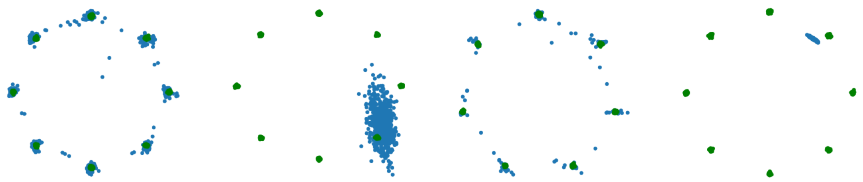
We also experiment our Method2 on CIFAR-10 data set. Our method basically maintains IS score and slightly increase sample diversity.

5.4 CelebA Data

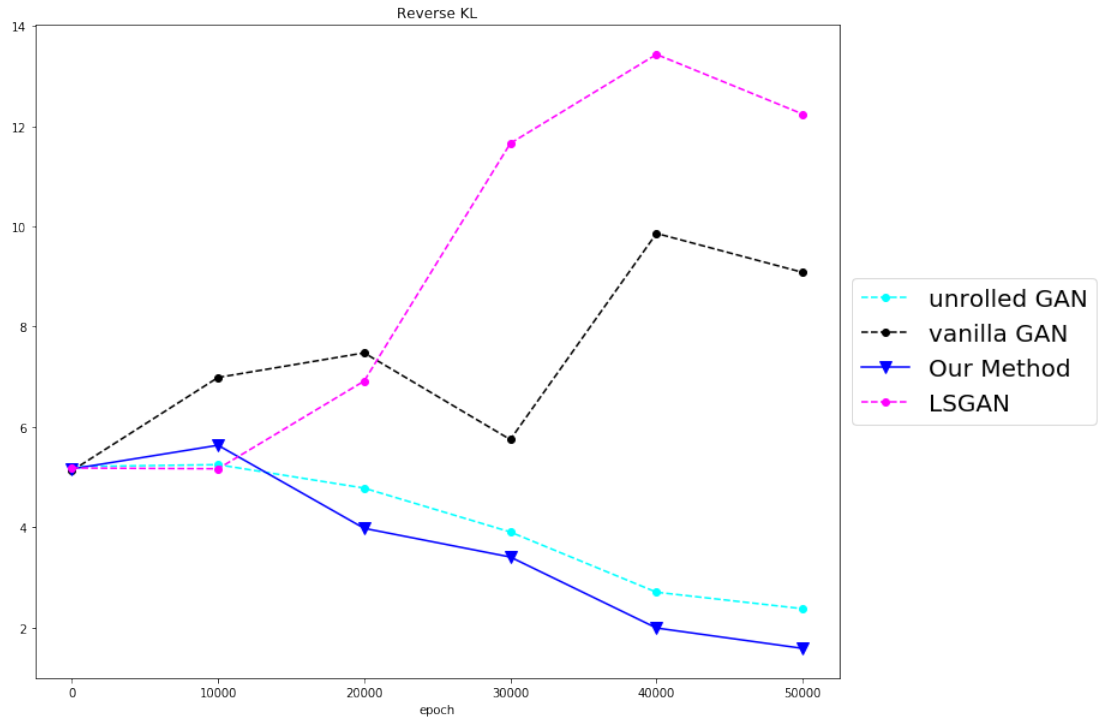
We report samples generated using with DCGAN architecture with Method

6 Conclusion And Future Work

In this paper, we present two methods which allow GANs optimize over KL Divergence in order to cover more modes of real data distribution. Through experiments, we found that GANs optimizing over KL Divergence become resistant to Mode Collapse(Method 1); We directly allow generators to minimize KL Divergence between real and generator distributions. Our main focus is on Method 2: we show that Method2 is an effective way to alleviate Mode Collapse. Still, there is some problems in Method2: 1. the implementation we use($tf.binary\_cross\_entropy$) tends to encourage generating samples whose values are either very high or very low. This situation causes generated samples to distribute in areas where all dimension values are very low or very high(e.g. a image of 3x64x64 where all pixels are close to 0) instead of evenly distributed. And definitely it's not treating the two inputs as distributions(like unnormalized energy). However, we found that there is no much difference between normalization or not. Another flaw is that as reported in [16], Method 2 can hardly perform well on image data of high resolution. Our future work is to find better way to introduce KL Divergence and further study the mechanism.

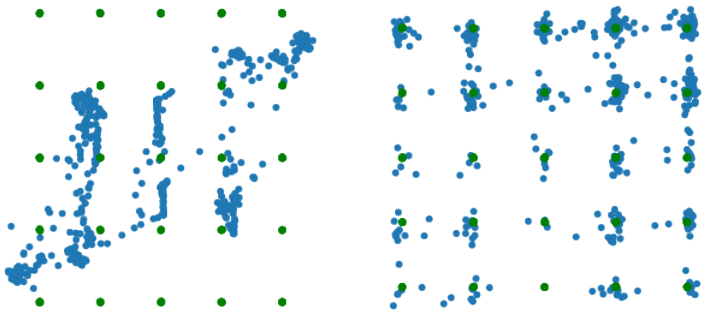


(a) Unrolled GAN (b) Vanilla GAN (c) **Our Mtehod 2** (d) LSGAN



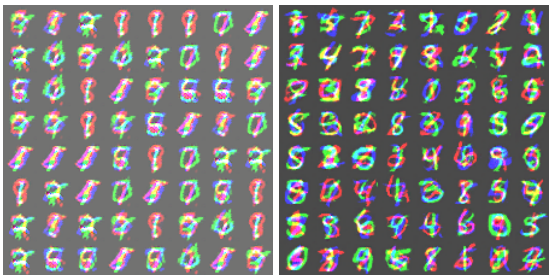
(e) Reverse KL Divergence; Our Method here is **Method 2**

Figure 5: Method 2: 2D Ring Dataset Experiments. In this experiments, we train different GANs on a dataset consists of points taken from 8 Gaussian distributions with different means and variance. The distributions are arranged on a ring. Our experiments show that under our setting, vanilla GAN and LSGAN collapse into certain mode or area, while Unrolled GAN and our mehod won't. The final epoch images are presented. Moreover, in order to quantitatively compare Unrolled GAN and our method, we calculate Reverse KL Divergence between both 2560 generated samples and real data points. The results show that our method outperform Unrolled GAN as training proceeds. The unstable phenomenon we observe at the early stage of training can be seen as a mild price paid for a gain of strong resistance to mode collapse using our method.



(a) Vanilla (b) Our Method 2

Figure 6: Method 2: 2D Grid Dataset; Vanilla GAN suffers from mode collapses. GAN using our method successfully captures all modes.



(a) DCGAN (b) Our Method 2

Figure 7: Method 2: Comparison Between DCGAN and our method

References

[1] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *CoRR*, abs/1701.07875, 2017.

[2] Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks. *ArXiv*, abs/1612.02136, 2016.

[3] Chao Du, Kun Xu, Chongxuan Li, Jun Zhu, and Bo Zhang. Learning implicit generative models by teaching explicit ones, 2019.

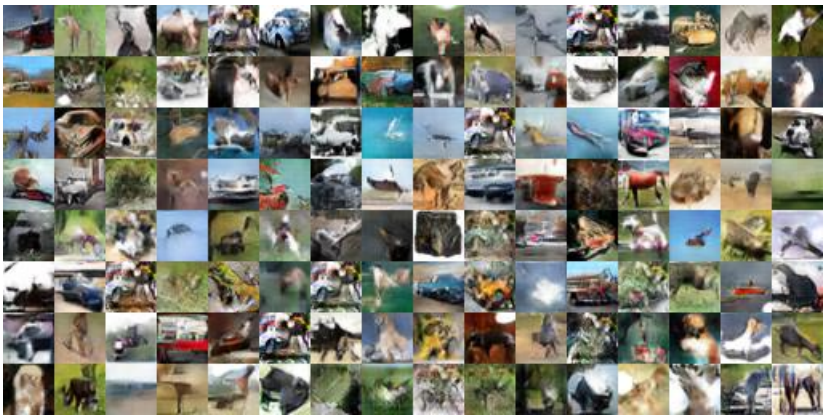
[4] Arnab Ghosh, Viveka Kulharia, Vinay P. Namboodiri, Philip H. S. Torr, and Puneet Kumar Dokania. Multi-agent diverse generative adversarial networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8513–8521, 2018.

[5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, 2014.

[6] Ferenc Huszar. How (not) to train your generative model: Scheduled sampling, likelihood, adversarial? 2015.

[7] Zinan Lin, Ashish Khetan, Giulia C. Fanti, and Sewoong Oh. Pacgan: The power of two samples in generative adversarial networks. In *NeurIPS*, 2017.

[8] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2813–2821, 2017.



(a) DCGAN

(b) Our Method 2

Figure 8: Method 2: Comparison Between DCGAN and our method. Our method helps GAN increase sample diversity.

[9] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *ICLR*, abs/1611.02163, 2017.

[10] Thomas Minka. Divergence measures and message passing. Technical report, 2005.

[11] Tu Nguyen, Trung Le, Hung Vu, and Dinh Phung. Dual discriminator generative adversarial nets. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2667–2677. Curran Associates, Inc., 2017.

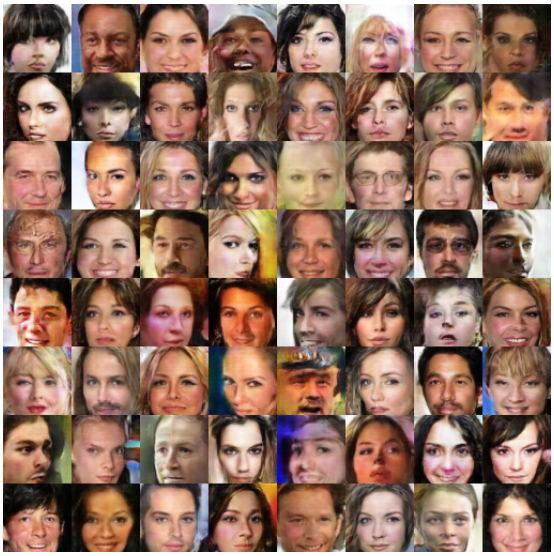
[12] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *NIPS*, 2016.

[13] Ben Poole, Alex Alemi, Jascha Sohl-dickstein, and Anelia Angelova. Improved generator objectives for gans. 2016.

[14] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U. Gutmann, and Charles A. Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *NIPS*, 2017.

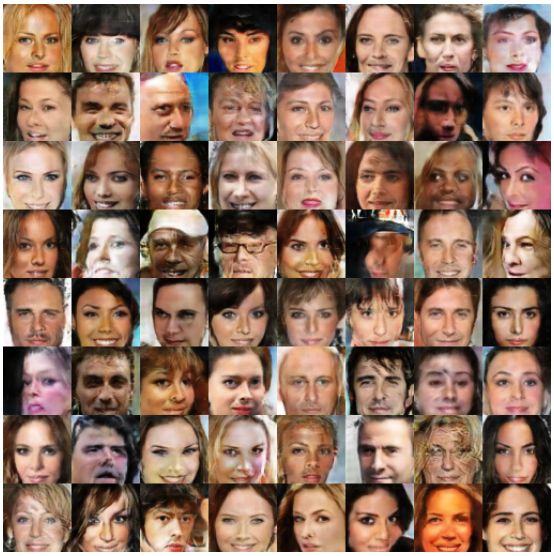
[15] Emre Ozkan Tohid Ardeshiri, Umut Orguner. Gaussian mixture reduction using reverse kullback-leibler divergence. 2015.

[16] Mingzhang Yin and Mingyuan Zhou. Semi-implicit generative model. *ArXiv*, abs/1905.12659, 2019.



(a) LSGAN

(b) Vanilla GAN



(c) **Our Method 2**

Figure 9: Method 2: images are more diverse and doesn't have decrease in quality