# Lecture 9: Are GANs Truly Distributive Learners?

*Lecturer: Aleksander Mądry* *Scribes: Dhroova Aiylam, Shahul Alam, Austin Wang*
*(Revised by Andrew Ilyas and Dimitris Tsipras)*

## 1 Recap: Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) have been the focus of much attention recently. The aim of this lecture will be to understand whether GANs really work (in theory and in practice), and how to tell when they don't. Recall that a GAN consists of a generator neural network $G$ and a discriminator network $V$ which are obtained by solving the following minimax optimization problem:

$$\min_G \max_V \mathop{\mathbb{E}}_{x\sim D}[\log(V(x))] + \mathop{\mathbb{E}}_{u\sim\mathcal{N}(0,I)}[\log(1-V(G(u)))].$$

The first term in the loss samples from the true distribution $D$; the second term samples from $G$ by seeding it with a random input (e.g. $\mathcal{N}(0, I)$). Viewed as a two-player game, the goal of $V$ is to maximize its discriminative ability, i.e. maximize the expectation of the output on samples from the true distribution and minimize the expectation on samples generated from $G$ from the latent distribution. The goal of $G$ is to model the true distribution as closely as possible, so that $V$ is unable to distinguish between samples from $G$ and samples from $D$. In addition, since the logarithmic function is strictly increasing, we consider each element of the loss function in log scale without consequence.

There is really no theory to support the fact that GANs can be effectively trained. The original paper, for instance, makes several assumptions which do not hold in most situations to which GANs are applied [7]. For instance, the authors assume arbitrarily expressive generator and discriminator neural networks and the ability to completely optimize the discriminator network between steps of updating the generator network. Assumptions are also made about the function over which we are optimizing, as the model of a concave-convex game results in certain well-defined convergence properties that are not necessarily generalizable. Nagarajan and Kolter are able to show that

> Under suitable conditions on the representational powers of the discriminator and the generator, the resulting GAN dynamical system is locally exponentially stable. [7]

That is, if generator and discriminator neural networks $(G, V)$ are initialized sufficiently close to the optimal $(G^*, V^*)$, then training with gradient descent will converge to this equilibrium. However, this sort of local result is of little practical use.

There is no consensus on the best way to train GANs in practice either. Some have suggested that GANs be trained via alternating gradient descent steps on $G$ and $V$, whereas, others suggest that training should take multiple steps on $V$, i.e. further optimize the discriminator, for each step on $G$ since the job of the discriminator is "harder." Yet other papers state that in fact $V$ should not be near-optimal, since this kills the gradient in $G$ and can cause the training to settle in a local optima.

To understand better what GANs are actually learning, and if they are learning at all, we turn our attention to analyzing how and in what aspects people have tried to evaluate GANs and to understanding exactly what they produce.

## 2 Learning Gaussian Distributions with GANs

To begin understanding the dynamics of training and convergence in GANs, Li, Mądry, Peebles, and Schmidt [5] investigate the GAN dynamics in simple cases related to Gaussian distributions, with the goal of finding an example simple enough to mathematically analyze but complex enough to still encapsulate

all of the relevant phenomena and potential complications—in particular, vanishing gradients and mode collapse—that arise in GAN convergence.

As Li et. al. find that is easily provable that, when the true distribution is a single univariate Gaussian, convergence of the generator to the true distribution is guaranteed, they focus mainly on analyzing true distributions constructed as uniform mixtures of two univariate Gaussian distributions with unit weights, showing ultimately that the optimal discriminator always converges, whereas the first order dynamics does not always converge nicely. Formally, they define the generator $G$ to be

$$\mathcal{G} = \left\{ \frac{1}{2}\mathcal{N}(\mu_1, 1) + \frac{1}{2}\mathcal{N}(\mu_2, 1) \mid \mu_1, \mu_2 \in \mathbb{R} \right\},$$

and the discriminator to be

$$\mathcal{D} = \{\mathbb{I}_{[l_1,r_1]} + \mathbb{I}_{[l_2,r_2]} \mid l, r \in \mathbb{R}^2 \mid l_1 \leq r_1 \leq l_2 \leq r_2\},$$

namely the set of indicator functions of sets expressible as two disjoint intervals, a simplification we can make due to the restriction to mixtures of Gaussian distributions and the resulting simplifications in the total variation distance between generators [5]. The optimization task of finding the best fit in total variation distance to the generator is thus

$$\begin{aligned} \hat{\mu} &= \arg\min_{\mu} \max_{l,r} L(\mu, l, r), \text{ where} \\ L(\mu, l, r) &= \mathop{\mathbb{E}}_{x \sim G_{\mu^*}}[D(x)] + \mathop{\mathbb{E}}_{x \sim G_{\mu}}[1 - D(x)] \end{aligned}$$

Li et. al. consider two common approaches to solve this optimization problem, *optimal discriminator dynamics* and *first order dynamics*.

*Optimal discriminator dynamics* involves performing stochastic gradient descent on $G(\hat{\mu}) = \max_{l,r} L(\hat{\mu}, l, r)$. Formally, given an initial $\hat{\mu}^{(0)}$ and step size $\eta_g$, we have updates defined as

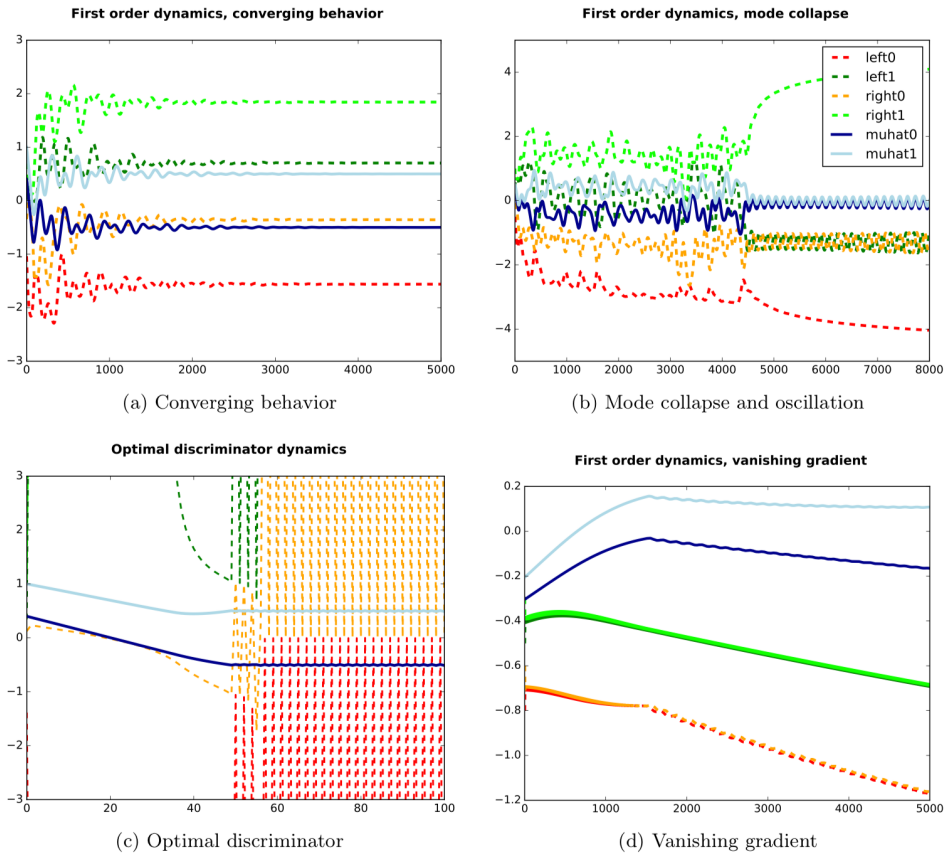
$$\begin{aligned} l^{(t)}, r^{(t)} &= \arg\max_{l,r} L(\hat{\mu}^{(t)}, l, r), \\ \hat{\mu}^{(t+1)} &= \hat{\mu}^{(t)} - \eta_g \nabla_{\mu} L(\hat{\mu}^{(t)}, l^{(t)}, r^{(t)}) \end{aligned}$$

Because this is difficult to perform in general for more complicated generators and discriminators, one often uses simultaneous gradient iterations on the generator and discriminator, as in *first order dynamics*. Formally, given initial $\hat{\mu}^{(0)}, l^{(0)}, r^{(0)}$ and step sizes $\eta_g, \eta_d$, we have

$$\begin{aligned} \hat{\mu}^{(t+1)} &= \hat{\mu}^{(t)} - \eta_g \nabla_{\mu} L(\hat{\mu}^{(t)}, l^{(t)}, r^{(t)}) \\ r^{(t+1)} &= r^{(t)} + \eta_d \nabla_{r} L(\hat{\mu}^{(t)}, l^{(t)}, r^{(t)}) \\ l^{(t+1)} &= l^{(t)} + \eta_d \nabla_{l} L(\hat{\mu}^{(t)}, l^{(t)}, r^{(t)}) \end{aligned}$$

In applying each to the mixture of Gaussian distributions, it was found that optimal discriminator dynamics always converged to the true distribution, whereas first order dynamics exhibited a variety of behaviors—either converging, experiencing mode collapse, or converging to a wrong value due to vanishing gradients. Figure 2 shows graphically some of the results of both, in particular demonstrating the different end outcomes that could result from first order dynamics.

We see in figure 2 that, in (a) and (c), the discriminators, or specifically the $\hat{\mu}$ values converge stably to the correct means. On the other hand, more often than not the first order dynamics fail, either when gradients vanish or when modes collapse, i.e. $\hat{\mu}_0$ and $\hat{\mu}_1$ converge on each other, resulting in only one mean represented.




(a) Converging behavior

(b) Mode collapse and oscillation

(c) Optimal discriminator

(d) Vanishing gradient

Figure 1: Different GAN behaviors for optimal discriminator dynamics and first order dynamics. The true distribution was $G_{\mu^*}$ with $\mu^* = (-0.5, 0.5)$, and the step size was taken to be 0.1. Solid lines represent the coordinates of $\hat{\mu}$, and the dotted lines represent discriminator intervals.

Li et. al. also study a phenomenon they call *discriminator collapse*, in which the local optimization landscape around the current discriminator encourages change in such a way that it loses representational power.

In figure 2 we see an example of discriminator collapse. In each graph, the difference of the true distribution and the generator distribution is given, with regions covered by the discriminator shaded. plot (a) shows the initial configuration of the example, plot (b) shows the result of the globally optimal discriminator for the initial condition, and plots (c), (d), and (e) each show the state of the generator and discriminator after 1000 steps, in slightly different variations—the first after just updating the discriminator against a fixed generator, and the other two after applying first order dynamics. From these we see discriminator collapse: the discriminator has incentive to have mass only on regions where the difference between the true distribution and generator distributions is positive, so it risks the collapsing of one of its intervals if in a negative region.

Ultimately, the results of the first order dynamics suggest that they are fundamentally flawed, or at least that any promising implementation of them should somehow be able to resolve the issues of vanishing gradients and mode collapse. The idea of analyzing a restricted set of simple but still interesting distributions helps to give us starting insight into the convergence behavior of GANs at more complicated levels.

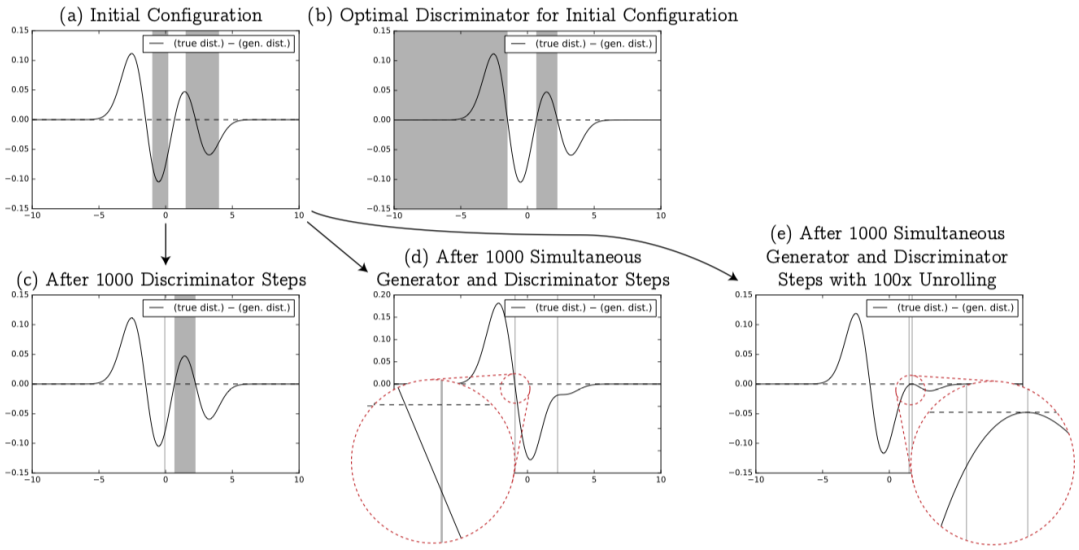



Figure 2: Example of Discriminator Collapse. Initial configuration has $\mu^* = \{-2, 2\}$, $\hat{\mu} = \{-1, 2, 5\}$, left discriminator $[-1, 0.2]$, and right discriminator $[-1, 2.5]$. The step size for plots (c) through (e) was 0.3.

# 3 What Is the Promise of GANs?

Even assuming that training converges to the equilibrium $(G^*, V^*)$, it is not obvious that the result is the model we want. It is natural to hope that $G^*$ has learned a distribution close to the true underlying $D$, but is this true? And if so, how might one certify it?

The generator $G^*$ determines a distribution $P_S \leftarrow G^*(u)$; consider fixing $P_S$ and solving for the best discriminator $V$ among *all* functions. The loss is

$$\text{Loss} = \int \Big( P_D(x) \log V(x) + P_S(x) \log(1 - V(x)) \Big) dx$$

where $V$ is non-parametric, so we can simply optimize it pointwise. Setting the derivative w.r.t. $V$ equal to 0 and solving yields

$$V(x) = \frac{P_D(x)}{P_D(x) + P_S(x)}$$

Thus if $(G^*, V^*)$ correspond to a GAN loss of 0,

$$\begin{aligned} 0 &= \int P_D(x) \log \left( \frac{P_D(x)}{P_D(x) + P_{S^*}(x)} \right) + P_{S^*}(x) \log \left( \frac{P_{S^*}(x)}{P_D(x) + P_{S^*}(x)} \right) \\ &= \frac{1}{2} KL(P_D || \frac{P_D + P_{S^*}}{2}) + \frac{1}{2} KL(P_{S^*} || \frac{P_D + P_{S^*}}{2}) \\ &= JS(P_D || P_{S^*}), \end{aligned}$$

where $KL$ is the Kullback-Leibler divergence and $JS$ is the Jensen-Shannon divergence.

Hence a GAN $(G^*, V^*)$ that has 0 loss has in fact succeeded in learning a distribution $P_{S^*}$ which is close to $P_D$ in an information-theoretic sense. However, we have assumed that

1. it's possible to find optimal $G^*, V^*$, and that we have a large number of examples to do so, and
2. the discriminator is infinitely expressive.

The second assumption is quite significant (and perhaps should not be take for granted), since we find that limiting the expressive power of the discriminator to something more feasible, i.e. assuming smoothness rather than infinite expressibility and training examples, we find that the discriminator is less able to discriminate between distributions in a way that makes minimizing the distance between the generated and real distribution, and moreover the generalization of the estimated generator distribution, more difficult. Arora, Ge, Liang, Ma, and Zhang [1] observed, for instance, that a discriminator $V$ with $n$ parameters cannot distinguish between the true distribution $P_D$ and an $N$-sample approximation $P_{\hat{S}}$, in the sense of minimizing the population distance between the two distributions to within a margin, i.e. $\hat{JS}(P_{\hat{S}}||P_D) < \epsilon$. They first show the following lemma:

**Lemma 3.1.** *Let $\mu$ be uniform Gaussian distributions $\mathcal{N}(0, \frac{1}{d}I)$ and $\hat{\mu}$ be empirical versions of $\mu$ with $m$ examples. Then we have $\hat{JS}(\mu, \hat{\mu}) = \log 2, \hat{W}(\mu, \hat{\mu}) \geq 1.1$.*

The proof for JS divergence follows from the fact that $\mu$ is a continuous distribution while $\hat{\mu}$ is discrete. For the Wasserstein distance, given empirical examples $x_1, \ldots, x_m$ and letting $y$ be a sample of the normal distribution, we have by concentration and union bounds that

$$P[\forall i \in [m] \|y - x_i\| \geq 1.2] \geq 1 - m \exp(-\Omega(d)) \geq 1 - o(1).$$

By the earth-mover interpretation of the Wasserstein distance, we get then that $d_W(\mu, \hat{\mu}) \geq 1.2P[\forall i \in [m] \|y - x_i\| \geq 1.2] \geq 1.1$.

In a more general case, consider the following metric:

**Definition 3.1** ($\mathcal{F}$-distance)**.** *Let $\mathcal{F}$ be a class of functions from $\mathbb{R}^d$ to $[0, 1]$ such that if $f \in \mathcal{F}, 1 - f \in \mathcal{F}$. Let $\phi$ be a concave measuring function. Then the $\mathcal{F}$-divergence with respect to $\phi$ between two distributions $\mu$ and $\nu$ supported on $\mathbb{R}^d$ is defined as*

$$d_{\mathcal{F},\phi}(\mu, \nu) = \sup_{D \in \mathcal{F}} \mathbb{E}_{x \sim \mu}[\phi(D(x))] + \mathbb{E}_{x \sim \nu}[\phi(1 - D(x))] - 2\phi(1/2).$$

In the case of GANs, we define our class $\mathcal{F}$ to be the class of neural nets with at most $p$ parameters, which Arora et. al. [1] find to be a better distance metric than JS or Wasserstein through the following theorem, by the guarantees of generalization. From this we, work toward a more generalized theorem about the limits of a smooth or finitely expressible discriminator, in effect showing that because of the limited number of discriminators a sufficient number of samples guarantees up to a high probability convergence of the empirical distribution to the true distribution for all discriminators. To show this, Arora et. al. [1] propose the following theorem:

**Theorem 3.1.** *Let $\mu, \nu$ be two distributions supported on $\mathbb{R}^d$, and $\hat{\mu}, \hat{\nu}$ be empirical versions with at least $m$ samples each. There is a universal constant $c$ such that when $m \geq \frac{cp\Delta^2 \log(LL_\phi p/\epsilon)}{\epsilon^2}$, we have with probability at least $1 - \exp(-p)$ over the randomness of $\hat{\mu}$ and $\hat{\nu}$,*

$$|d_{\mathcal{F},\phi}(\hat{\mu}, \hat{\nu}) - d_{\mathcal{F},\phi}(\mu, \nu)| \leq \epsilon$$

The proof follows from concentration bounds. We can show that with high probability, for every discriminator $D_v$,

$$|\mathbb{E}_{x \sim \mu}[\phi(D_v(x))] - \mathbb{E}_{x \sim \hat{\mu}}[\phi(D_v(x))]| \leq \epsilon/2$$

$$|\mathbb{E}_{x \sim \nu}[\phi(1 - D_v(x))] - \mathbb{E}_{x \sim \hat{\nu}}[\phi(1 - D_v(x))]| \leq \epsilon/2.$$

To show this, we can in essence use the Chernoff bound applied with the stated bound in the proof, $O(\frac{p\Delta^2 \log(LL_\phi p/\epsilon)}{\epsilon^2})$, and union bound, assuming a sufficiently large $c$, gives us the above inequalities, from which the generalization claim of the theorem follows.

Thus a model with a finite parameterization can be fooled with a finite number of samples – the GAN has learned nothing but a very coarse approximation of $P_D$. While we would like to be able to prove that training does not settle to such $(\hat{G}, \hat{V})$ in practice, in the next section we will focus on ways to verify this for a GAN that has been trained.

# 4 Verification of GANs

Given a GAN $G \sim P_S$ and a distribution $P_D$, there are a number of heuristics which can be used to check whether $G \sim P_D$, i.e. whether $G$ has learned the target distribution. While heuristics are not themselves sufficient to conclude that $G \sim P_D$, they can provide evidence for or against the claim that GANs are truly distribution learners.

One simple idea is to just compare the supports. In particular, one can generate samples from $P_S$ and verify that that they look approximately like samples from $P_D$. Historically this was done by inspection, so if a GAN was trained on a distribution of e.g. human faces, its output would be examined in the hopes that the results were indeed "face-like."

One heuristic to quantify this quality is the *inception score*[3]:

$$\textit{inception score} = \exp\left(\mathbb{E}_{x \sim P_S}\left[D_{KL}(p(y \mid x)||p(x))\right]\right),$$

where $D_{KL}(p(y \mid x)||p(x))$ is the Kullback-Leibler distance between the distribution of labels $p(y)$ and the distribution of softmax probabilities $p(y|x)$.

To calculate the inception score, a neural network is first trained on ImageNet and then fed images sampled from the GAN. Note that, if $D_{KL}(p(y \mid x)||p(y))$ is high, then the inception score is high. Thus, the intuition behind this metric is clear: if the output of a GAN is meaningful, then the neural network should have a low-entropy belief about the label of any particular image, and thus the KL distance should be high.

Although the inception score is an intuitive heuristic, it can be artificially high even if the GAN in question has learned nothing about the target distribution. Consider, for example, a potentially complex distribution with three modes of different labels. A GAN which simply randomizes over the modes will earn a high inception score, but it has clearly failed to learn the target distribution. For this reason, it is encouraged to first train on a dataset and only then evaluate the result using the inception score (which can check the support, but does not guarantee correctness).

Another approach to GAN verification is to show that the GAN has not simply memorized the training data. In concrete terms, consider sampling from a GAN and measuring the distance of the sample to its nearest neighbor in the training set. If the GAN has done more than just memorize its training data, then this value should not be too small. However, it is not clear what notion of distance is appropriate for this sort of estimation. The usual $\ell_2$-norm, for instance, is not robust to transformations such as shifts or changing of pixel values which have little impact in the image-space.

Radford, Metz, and Chintala [8] propose using the latent space embedding to understand the output of GANs. Namely, the authors interpolate a seed $x$ from $u$ to $u'$ and study how the output $G(x)$ varies from $G(u)$ to $G(u')$. Rather than obtaining images that were simply a convex combination of the endpoints $G(u)$ and $G(u')$, they found the output was semantically meaningful (à la word2vec) 4. This evidence would seem to support the fact that GANs truly are learning.

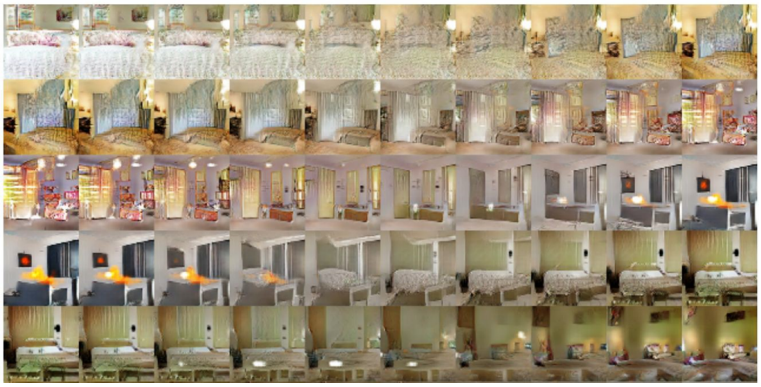


Figure 3: Interpolating between points in latent space yields semantically meaningful outputs.

Another property of a GAN that has truly learned the target distribution is that its support should not be too small. Arora and Zhang [2] suggest a heuristic based on the birthday paradox: if after sampling $s$ images from the GAN there is a duplicate, then one would expect the support of the distribution to have size $O(s^2)$. There are, however, a couple of subtleties here.

First, the birthday paradox only applies when the distribution over images is roughly uniform. If, for instance, the distribution assigns 10% probability to a single image and is uniform on a large number of other images, then a duplicate is likely with just 20 samples. The authors address this point by noting that non-uniformity over the images should itself be considered a failure mode for GAN training.

More importantly, it's not clear how to apply the birthday paradox argument when the support is infinite (as it is for a GAN), or even what is meant by a duplicate. The authors resolve this issue by looking instead at near-duplicate: from a finite sample the e.g. 20 closest pairs are selected using some heuristic for distance, and those 20 pairs are visually inspected to determine whether any of them would be considered duplicates by a human.

There is still the question of what distance heuristic is best to use. For the CelebFaces dataset the authors chose to use standard Euclidean distance, since the samples are centered and aligned so Euclidean distance is effectively robust. For CIFAR-10 data, they instead trained a CNN on the images and used the top-layer representation as a latent space embedding; these embeddings were then compared via Euclidean distance.

Using this birthday-paradox heuristic, the authors were able to experimentally verify the not-so-surprising result that as the number of parameters of the discriminator grew, so did the diversity of the learned distribution. Unfortunately, their experiments also suggest that current GAN approaches (especially for images of high quality) fall short of learning the target distribution since their support is too small, indicating the possibility of mode collapse.

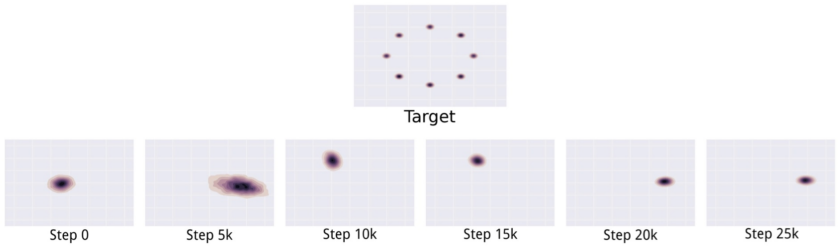



Figure 4: An illustration of the mode collapse problem from [4] on a toy dataset. The first image is the model that the GANs should learn, whereas the bottom ones are the models that are actually learned.

Santurkar, Schmidt, and Mądry [9] studied the diversity of the generated distributions of several popular GANs with a focus on mode collapse using automated classification-based measures. Their methodology involved taking the Large-Scale CelebFaces Attributes dataset [6] and the Large-Scale Scene Understanding datasets [10], both of which are rich with annotations—for instance, the faces are labeled as male or female and smile or no smile—and training a simple classifier to distinguish attributes based on image input, which proved to do so successfully with high confidence. The intent of such a classifier was to be able to label images sampled from the generator trained on these datasets and so give a measure of the distribution of the generator output, all in a fairly automated process as opposed to via manual annotation. While the distribution of the annotations among the four classes was close to uniform for some GANs, like ALI, there were many other GANs for which the distribution of annotations were highly asymmetric. This is further evidence that mode collapse can be a problem in practice (see figure 4).

There are a plethora of other evidence that further encourage this skeptical take on GANs. For example, when the authors compared the spectrum of the covariance matrix of the GAN-generated images to that of the true training data they found that most directions were dropped. In addition, classifiers trained on the GAN distribution generalized considerably more poorly than those trained on the actual images due to severe overfitting. Only the simplest model, a linear classifier, was able to avoid

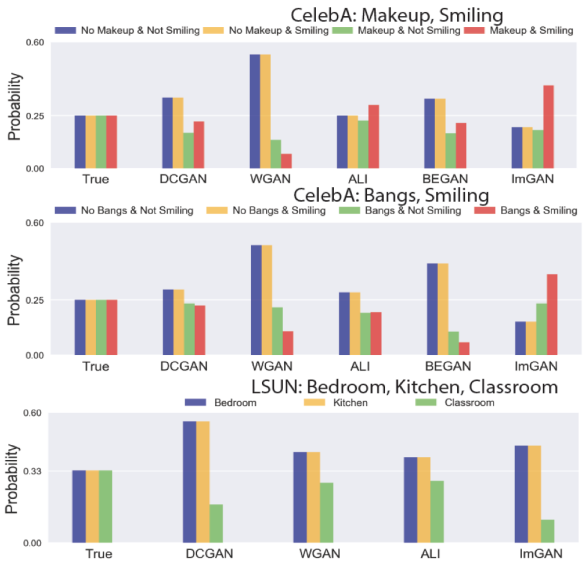



Figure 5: Distributions of images generated from GANs within classification categories, based on trained annotator

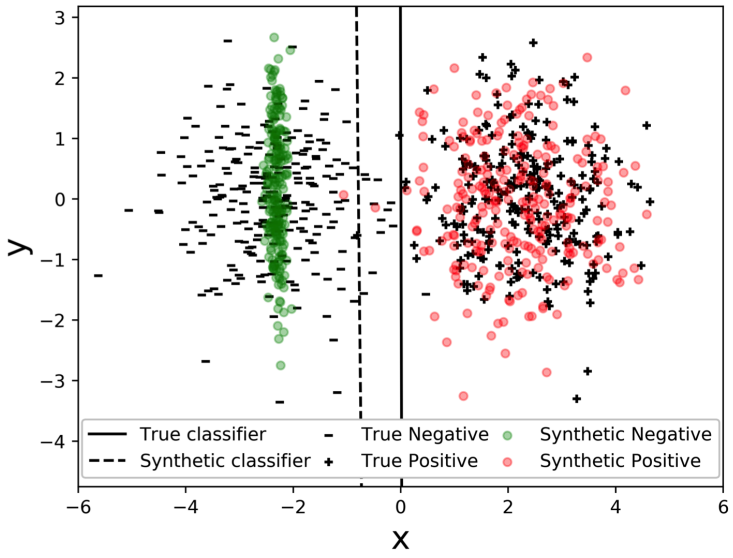



Figure 6: Underfitting of linear classifier on GAN distribution. A simple linear classifier misclassifies much of the original data when applied to the output from a GAN trained on the original data.

overfitting but in turn significantly underfit on the training set (4).

Yet another piece of empirical evidence is that the authors found training with hundreds of thousands of GAN-generated images to be just as effective as using a few hundred samples from the actual training data. Taken as a unified corpus, these bits of evidence support the idea that GAN-generated data is derived from a distribution that is much less diverse than the true training distribution. If we are to believe that GANs do indeed learn the underlying distributions on which they are trained then the various aforementioned heuristics should confirm this fact; however, this does not appear to be the case.

# References


[1] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). *CoRR*, abs/1703.00573, 2017.

[2] Sanjeev Arora and Yi Zhang. Do gans actually learn the distribution? an empirical study. *CoRR*, abs/1706.08224, 2017.

[3] Shane Barratt and Rishi Sharma. A note on the inception score. *CoRR*, abs/1801.01973, 2018.

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

[5] Jerry Li, Aleksander Madry, John Peebles, and Ludwig Schmidt. Towards understanding the dynamics of generative adversarial networks. *CoRR*, abs/1706.09884, 2017.

[6] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. *CoRR*, abs/1411.7766, 2014.

[7] Vaishnavh Nagarajan and J. Zico Kolter. Gradient descent GAN optimization is locally stable. *CoRR*, abs/1706.04156, 2017.

[8] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

[9] Shibani Santurkar, Ludwig Schmidt, and Aleksander Madry. A classification-based perspective on GAN distributions. *CoRR*, abs/1711.00970, 2017.

[10] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365, 2015.