A Study on Generative Adversarial Networks Exacerbating Social Data Bias

by

Niharika Jain

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2020 by the
Graduate Supervisory Committee:

Subbarao Kambhampati, Chair
Huan Liu
Lydia Manikonda

ARIZONA STATE UNIVERSITY

August 2020

ABSTRACT

Generative Adversarial Networks are designed, in theory, to replicate the distribution of the data they are trained on. With real-world limitations, such as finite network capacity and training set size, they inevitably suffer a yet unavoidable technical failure: mode collapse. GAN-generated data is not nearly as diverse as the real-world data the network is trained on; this work shows that this effect is especially drastic when the training data is highly non-uniform. Specifically, GANs learn to exacerbate the social biases which exist in the training set along sensitive axes such as gender and race. In an age where many datasets are curated from web and social media data (which are almost never balanced), this has dangerous implications for downstream tasks using GAN-generated synthetic data, such as data augmentation for classification. This thesis presents an empirical demonstration of this phenomenon and illustrates its real-world ramifications. It starts by showing that when asked to sample images from an illustrative dataset of engineering faculty headshots from 47 U.S. universities, unfortunately skewed toward white males, a DCGAN's generator "imagines" faces with light skin colors and masculine features. In addition, this work verifies that the generated distribution diverges more from the real-world distribution when the training data is non-uniform than when it is uniform. This work also shows that a conditional variant of GAN is not immune to exacerbating sensitive social biases. Finally, this work contributes a preliminary case study on Snapchat's explosively popular GAN-enabled "My Twin" selfie lens, which consistently lightens the skin tone for women of color in an attempt to make faces more feminine. The results and discussion of the study are meant to caution machine learning practitioners who may unsuspectingly increase the biases in their applications.

DEDICATION

This work is dedicated to my parents, my sister, my teachers, my lab mates, my mentors, and my friends.

ACKNOWLEDGMENTS

First and foremost, I thank Dr. Subbarao Kambhampati for being my mentor in the world of academia and research for over two years. I joined Dr. Rao's lab after taking his Intro to AI course in my junior year. To date (even after having taken a degree's worth of graduate classes), it's by far the most challenging and stressful course I've ever taken. It's also the most rewarding. Thank you to Dr. Rao for giving me an A+ in that class from his mystery, smoke-and-mirrors grading metrics, and thank you for taking me on as one of two undergraduate researchers in the lab. Thank you for the idea for this entire culminating project. And thank you for keeping the world of AI just as challenging and wondrous for a researcher as you were able to for the Intro to AI student who had never heard of a neural network before.

Thank you to my committee for pushing me to be a better researcher: Dr. Subbarao Kambhampati, Dr. Huan Liu, and Dr. Lydia Manikonda for generously giving me your time to discuss ideas, provide feedback, and push me to creating better work.

Thank you to my lab mates, past and present. My co-authors to this work – Alberto Olmo, Sailik Sengupta, and Lydia – thank you so much for the countless hours of time over the past year and a half spent in planning, reading, re-planning, re-reading, experiments, analysis, and writing, writing, writing. Thank you for a few sleepless nights with Anywhere on Earth deadlines. Sailik, you were the first lab mate I ever spoke to (and it stayed that way for the first several months as I struggled to stop being so nervous). Thank you for somehow finding hours in your already negative free time to answer my thousands of questions and talk through concepts with me until I could

understand them. Thank you for always knowing what to do next when we felt stuck. Thank you for always being welcoming and inviting me to watch the last season of Game of Thrones with the lab. Alberto, thank you for being a great salsa dancing partner, for sneakily taking pictures of me dozing off in lab, and for reading my entire thesis and giving me comments as I frantically worked to complete it the day it was due. My fellow official and unofficial TAs for the Intro to AI section this fall: Mudit Verma, Lin Guan, Utkarsh Soni, Alberto, and Zahra Zahedi. Thank you for the long days of grading, setting assignments, office hours, fighting over the answers to Dr. Rao's so-called-not-that-hard midterm, and the bike/skateboard races to and from class. To the senior members in the lab: Sailik, Anagha Kulkarni, and Sarath Sreedharan: I have come to each and every one of you for your mentorship and stolen what I presume is some of your best advice. I look up to you. I have no idea how you do the whole Ph.D. student thing so perfectly, but I hope one day I'm half as good of a student and researcher as you are. Sachin Grover, thank you so much for the Starbucks and Insomnia cookie runs and for always encouraging me by reminding me of my accomplishments in the lab when I haven't been able to see them. Sriram Gopalakrishnan, thank you so much for talking through ideas with me, for being so welcoming, for the usually-way-too-loud and way-too-irrelevant conversations in the lab, and for telling me I didn't have to sit in a corner for the first lab meeting I attended. Yantian Zha, thank you for time and again offering me room in your work in computer vision because I told you it interested me, and thank you for playing piano with (for) me. Sarath, Lin, and Mudit, thank you for taking graduate classes with me, doing group projects with me, and helping me understand the assignments. It's safe to say I would have had a much harder time passing without you. Tathagatha

Chakraborti, thank you for imparting your wisdom that papers will always get rejected if they're not submitted right before the deadline at 5am, and thank you for being the first Ph.D. student I've ever witnessed crossing the threshold into being a real-life doctor. Lydia, again, thank you so much for being such an impactful mentor in my life and for celebrating me when I fail to do so myself. Thank you for your persistence, kindness, and encouragement when I've been dejected from rejection after rejection. As a woman of color, it's important to see other women of color find success; we all need people in our lives to hold as role models.

Thank you to my family. My parents, my first role models, through the sacrifices you've made throughout your life for our home, you made sure I never struggled an ounce more than I needed to. My only real burden in my entire life has been focusing on academics, and you've offered me all your help to navigate through even that; I never had to go it alone. I probably cooked two weeks' worth of my own meals over the past two years, never took the car that you gave me for a single repair on my own. Thank you to my sister whose every footstep I follow. Thank you for teaching me how to code, bragging about me, and for always being there for me. Thank you for sending me a Snapchat which inspired a case study in this work. Thank you to my friends. Those who helped me pass Intro to AI, for all the whiteboarding to help me understand concepts like Bellman backup and gradient descent. Thank you for the friends who carried one another through the trifecta. Thank you to those who supported me through all the messes of college. Thank you to my previous teachers and mentors: the engineers at my software internships, my supervisors at all the jobs I've been blessed to have called labors of love, and to all the faculty I've been fortunate to work with and look up to in my time at ASU.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

Despite the breakthroughs in deep learning for the field of computer vision, one glaring drawback of the technique persists: neural networks require massive amounts of data to train. Where data acquisition can prove expensive or infeasible, this places a burden on machine learning practitioners who seek to utilize the pattern-recognition capabilities of the technology. Those frustrated by their limited, sparse, and unbalanced training datasets have turned to an explosively popular method of data augmentation: generation of synthetic data. Generative Adversarial Networks (GAN, Goodfellow et al. 2014) are a deep learning technology designed to create realistic never-before-seen data by mimicking the patterns it finds in a training set. While it may seem to the machine learning practitioners that they are receiving novel inputs to their downstream classifiers (as opposed to what they would receive from traditional data augmentation techniques, such as affine transformations (O'Gorman and Kasturi 1995) which seemingly duplicate training data) for little cost, it is not clear that they recognize the inherent dangers of this approach caused by inbuilt failures of the original GAN design.

As is the case with any machine learning technology, GANs are intended to capture the patterns in the data they receive, and thus cannot avoid also capturing any biases inherent in the data. Even in this (imaginary) best-case scenario where they work exactly as designed, this is an area for practitioners to tread lightly. The biases which exist in the original data will still be present – in the same capacity – in the generated data. In an average-case scenario, the biases in the original data are actually worsened in the generated data. A necessary condition of effective data augmentation is to sample data from the same distribution p_{data} as the train and test data, so that classifiers trained on data augmented with these samples will have an increased ability to generalize to a learned underlying real-world distribution. To illustrate, given a limited dataset of cat images, the traditional data augmentation technique of performing simple rotations and translations on a single image does not make it any less cat-like; an upside-down cat is still a cat. GANs on the other hand, while theoretically guaranteeing that their generated distribution p_g converges to p_{data}, do not meet this criterion in practice. One of the most pernicious and pervasive failure cases of GAN non-convergence is *mode collapse*: the generated distribution is far less diverse than the training distribution. GANs will create realistic images which appear to be sampled from the original data's underlying distribution, but actually are only representative of a small subset of the types of values present in the training data.

This fact about GANs has especially unfortunate implications for modes of data which are underrepresented in the augmented data. In this thesis, the work uses a facial image domain to investigate social biases, because we as a society are attuned to recognizing the underrepresentation of racial and gender minorities. We train an off-the-shelf DCGAN to "imagine" faces of engineering professors. The training set it sees is curated from university faculty directories of 47 of the best engineering schools in the country. This dataset is (unfortunately) already biased: the images are skewed toward white males. Exacerbation of bias is particularly harmful for minority classes. This work highlights how the DCGAN places far less priority on generating faces with a non-white skin color and feminine facial features, and thereby creates a new distribution of professors with even less representation of minorities than in the already-skewed real-world professor distribution.

That a machine learning model captures and perpetuates bias is already regrettable, though unsurprising to anyone who is familiar with machine learning. It must be stressed, though, that we aim to show that the GANs susceptible to mode collapse *amplify* bias. For any feature which is skewed in the original data, a GAN is likely to make the divergence between the modes along that dimension even more drastic. For example, if university professors have a, say, 3:1 ratio of men to women, then a GAN's imagination of university professors could have a worse ratio of 10:1. Augmenting the original distribution with the generated distribution will shift a downstream classifier's training distribution to make more extreme the biases of the real world. This is significant because real-world datasets are always skewed across crucial features, unless they are carefully constructed not to be. Data extracted from the web or from social media is bound to carry the same social biases from the real world.

Since a GAN only sees low-level features such as pixel values, it will not distinguish between any high-level features it ends up biasing. This exacerbation will occur for *any* skew in a dataset. For facial images, this could be pose, lighting, facial expression, or accessories. We choose to focus only on *sensitive* features to analyze bias. This thesis defines sensitive features to be social characteristics historically discriminated against, and which are or ought to be legally protected. As we shall see, though the GAN used in the experiment seems to pick up on the bias that professors often wear glasses, the thesis does not mark this as a problematic failing of GANs for facial data; there is no systemic problem where individuals with perfect vision face any more difficulties in navigating academia than individuals who wear glasses. This definition still leaves a large set of important characteristics (which can be reasonably deduced from images) to consider, such as age. However, the experiments conducted in this work specifically investigate biases along two axes: gender and race.

Chapter 2 opens the work with a technical description of the theoretical GAN objective, summarizes related works detailing why mode collapse occurs in practice, and provides an experiment to verify that the effects of the technical failure are more severe for GANs trained on highly non-uniform training sets than uniform ones. Chapter 3 describes the experiments conducted to showcase the implications of this GAN failure for social data bias in facial images. Chapter 4 shows that conditional variants of GANs, which solve image-to-image translation problems rather than pure image generation, are also susceptible to capturing social data bias. It draws attention to Snapchat's presumably GAN-based "My Twin" gender-swap selfie lens by demonstrating that it consistently lightens skin tones for women of color. It then presents a hypothetical "Engineering Professor" selfie lens to translate images of celebrities to engineers and notes that it reliably transforms feminine facial features to male ones and lightens skin tones for people of color. Chapter 5 offers a discussion on the ethical implications of exacerbating data bias from applying GANs or GAN-based data augmentation for downstream tasks by briefly analyzing the discrimination along two sensitive features, generalizing the implications of this work to another domain, and concludes the thesis.

CHAPTER 2

THE ADVERSARIAL GAME

1. The Loss Function

A GAN simulates a zero-sum game between two players, the discriminator D and generator G; each player's reward is the other player's loss. D and G are differentiable functions with respect to their inputs and parameters. D is a function over an observed variable x, and G is a function over a latent variable z. Both functions aim to minimize an individual cost (J^D or J^G) which depends on both players' parameters (θ^D and θ^G), while only being able to control their own. This is not an optimization problem, such as those solved by traditional classifiers. The solution to an optimization problem is a minimum, while the solution to a game is a Nash equilibrium (Goodfellow 2016).

D and G are both represented by neural networks. D is a binary classifier which aims to discriminate between images which come from a real-world data distribution p_{data} and those who do not. G aims to generate images from p_g that fool D into thinking they come from p_{data} (Figure 1). Thus, one can see the generator network's goal as sampling from p_{data}, given input of a random noise vector. It optimizes its weights according to a loss function representing how "real" its generated images look. The generator never directly compares the image from its generated distribution $p_g(x)$ to an image from $p_{data}(x)$. It instead optimizes its weights to minimize the reward of D, receiving feedback from the discriminator's cross-entropy through backpropagation.

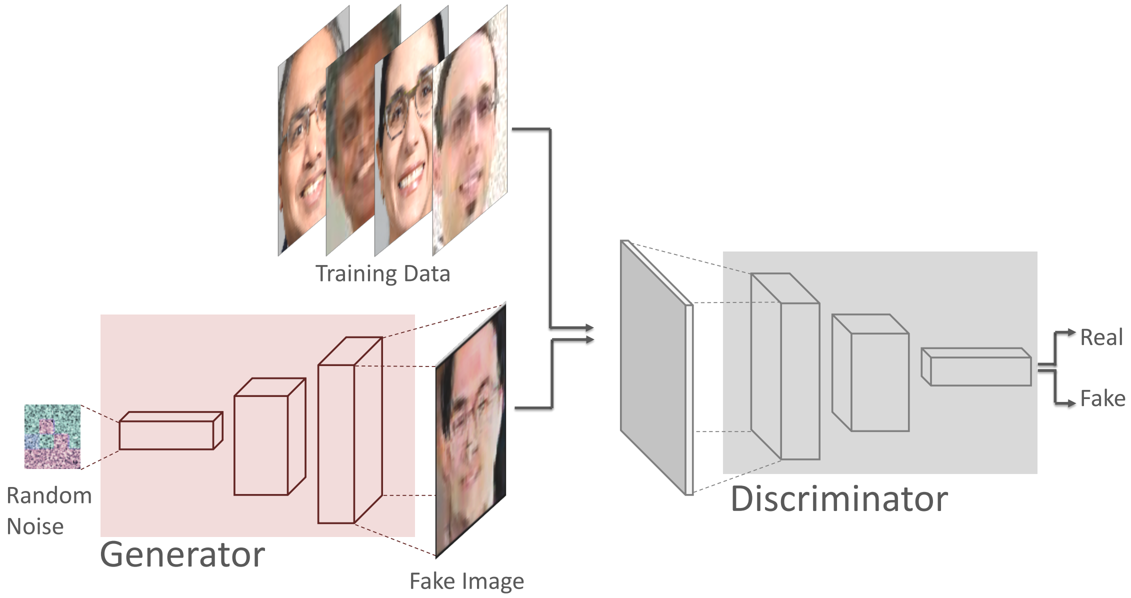


Figure 1: Generative Adversarial Network Architecture for images. Figure reproduced from Silva 2018.

A traditional cross-entropy loss is used in the discriminator for all GANs. Here, it is binary (Goodfellow et al. 2014):

$$J^D \;=\; -\frac{1}{2}\mathbb{E}_{x\sim p_{data}}\log\big(D(x)\big) - \frac{1}{2}\mathbb{E}_{z\sim p_z}\log\Big(1 - D\big(G(z)\big)\Big)$$

Minimizing this cost maximizes the likelihood of D marking inputs *x* as real, in expectation over the space of real images from a dataset p_{data}, and maximizes the *un*likelihood of D marking an input $G(z)$ as real, in expectation over the space of random noise (p_z, the prior of z). The generator's cost is simply the negative of the discriminator's, so $J^D + J^G = 0$.

$$J^G \;= -J^D = \;\frac{1}{2}\mathbb{E}_{x\sim p_{data}}\log\big(D(x)\big) + \frac{1}{2}\mathbb{E}_{z\sim p_z}\log\big(1 - D(G(z))\big)$$

Since both costs are associated with each other, the D's reward can be used to summarize the value of the two-player game.

$$V(G,D) = \mathbb{E}_{x\sim p_{data}}\log\big(D(x)\big) + \mathbb{E}_{z\sim p_z}\log\big(1 - D(G(z))\big)$$

$G(z)$ and $x{\sim}p_{data}$ are both values in an image space; they just come from different distributions. The second term can be equivalently written as $\mathbb{E}_{x\sim p_g}\log\big(1-D(x)\big)$. An expectation of a continuous random variable x is its integral with respect to its probability density function $f(x)$. $\mathbb{E}[x] = \int_{\mathbb{R}} x\, f(x)\, dx$, so the value of the game is rewritten:

$$V(G,D) = \int_x p_{data}(x)\log\big(D(x)\big) + p_g(x)\log\big(1-D(x)\big)\,dx$$

$p_g(x)$ refers to the probability that p_g assigns to some input x. This is the relative likelihood x appears in the dataset, compared to other images.

An equilibrium of the game occurs at $min_G\ max_D\ V(G,D)$, or expanded,

$$min_G\ max_D \int_x p_{data}(x)\log\big(D(x)\big) + p_g(x)\log\big(1-D(x)\big)\,dx$$

For G and p_g fixed, the cost function of D finds its minimum at $D^*(x) = \frac{p_{data}}{p_{data}+p_g}$. The equilibrium of this game $V(G^*,D^*)$ occurs when p_g converges to p_{data}, where $D^* = 0.5$ (Goodfellow et al. 2014). In other words, G samples from a distribution identical to p_{data} and produces images that are so realistic, in D's perspective, that it cannot tell the difference between real and fake images.

The training procedure to solve this game is to sample examples from the data and latent distributions and iteratively update their parameters simultaneously with respect to the gradients of their costs. While their costs aren't computed at the same time and using the same parameter values, their updates are considered "simultaneous" because their costs are being calculated in the same training iteration and on the same minibatch of z.

A generator's ability to learn depends on the gradients it receives. In practice, it works better to adjust the cost function so that G is not maximizing D's cross-entropy. In the theoretical minimax game, the term in J^G corresponding to the discriminator's classification over fake images is $\log\bigl(1 - D(G(z))\bigr)$. There is no substantive gradient sent to G when D can reliably distinguish between real and fake images, such as in the beginning of training when the generator has not yet learned how to make believable examples, because J^G approaches 0 as $D(G(z))$ approaches 0. Instead of maximizing J^D, G may minimize D's cross-entropy over fake images with flipped targets, (where fake images are labeled 1) in a non-saturating game (Goodfellow 2016). Then, $J^G = -\log\bigl(D(G(z))\bigr)$, which approaches ∞ as $D(G(z))$ approaches 0. This alters the game so it is no longer zero-sum, but the steep cost function ensures that a real-world generator will make sufficient parameter updates to learn. Intuitively, with the change G is maximizing the log-likelihood of D being fooled instead of minimizing the log-likelihood of D being correct. This is no longer a zero-sum game, and it cannot be represented with one cost function, but it works well in practice.

2. Mode Collapse

In practice, there is a common technical failure of GANs, called *mode collapse*, where the diversity of the generator's output images is far lower than the original training data. Complete mode collapse is rare; this is where the generator maps multiple inputs in the latent space to the exact same output, producing identical images for differing values of z. Partial mode collapse, on the other hand, is a largely unavoidable problem for most GANs: the generator maps different inputs in the latent space to similar outputs, such as those which have the same colors/textures or depict the same object (Figure 2). While the minimax game does have an optimal solution, the training procedure used to achieve this equilibrium in practice is not guaranteed to converge. For example, in some games, instead of Simultaneous Stochastic Gradient Descent (SGD) approaching equilibria, it will only orbit them.

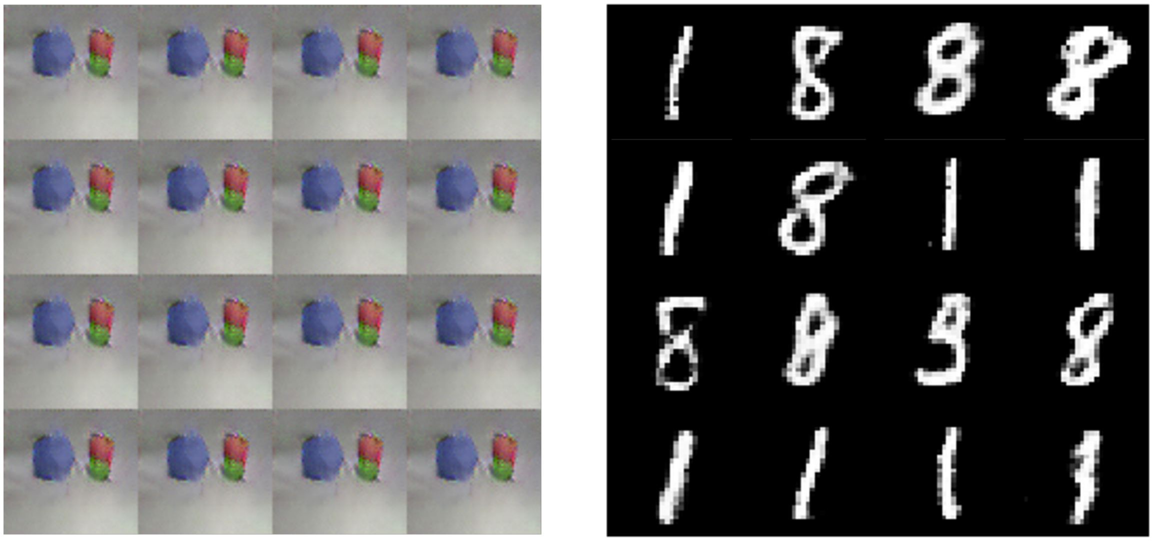


Figure 2: Complete and partial mode collapse

While the generated distribution converges to the original training distribution in the game presented, a training procedure making updates to D and G might approach a solution to an entirely different game. Simultaneous SGD is thought to be one of the reasons for the failure case of mode collapse, because it behaves like a minimax game at times and like a maximin game others (Goodfellow 2016, Grnarova 2018). G has extremely different optimal strategies for these two versions.

$$max_D \, min_G \mathbb{E}_{x \sim p_{data}} \log\big(D(x)\big) + \mathbb{E}_{z \sim p_z} \log\big(1 - D(G(z))\big)$$
$$\neq \, min_G max_D \log\big(D(x)\big) + \mathbb{E}_{z \sim p_z} \log\big(1 - D(G(z))\big)$$

In the first game, G only needs to learn to generate the image that will look most realistic to D. The maximin game over D focuses on the discriminator adjusting its strategy to maximize its worst-case value without knowing G's strategy (Shah 2017): D must spend its effort learning to detect this fake image, until G will learn to produce a different single realistic image. A cat-and-mouse game ensues. In other words, the generator's optimal strategy in the first game is complete mode collapse.

In the second game, G chooses its strategy over a worst-case scenario of D's actions, where D is free to learn the ratio of the two distributions' densities. The minimax game over G focuses on the generator adjusting its strategy to minimize D's best-case value without knowing D's strategy. Here, the generator's optimal strategy is to reproduce the entire training distribution, so D cannot tell the difference between any real and fake images.

There are several GAN architectures which reduce the effects of mode collapse. One of these is the Unrolled GAN (Metz et al. 2016), which ensures that the inner minimization over G is completed first in a minimax game by building a computational graph to describe the updates of D k steps into the future and optimize G over more than one copy of the discriminator. Other GAN architectures which perform well in converging to the original training distribution are the Wasserstein GAN (Arjovsky, Chintala, and Bottou 2017) and VEEGAN (Srivastava et al. 2017). These GAN architectures, while impressive in their coverage and diversity, do not create images of good quality (Lala et al. 2018), even for the MNIST dataset (LeCun 1998), so they are not usually used for practical applications.

3. Finite Settings

The theoretical guarantee for p_g to converge to p_{data} in Goodfellow et al.'s work (2014) is the result of assuming an infinite training set size and infinite network capacity for both G and D. In practice, a GAN using the original loss function will not converge, and mode collapse will occur as a result of insufficient examples and network parameters.

The problem of attracting sufficient gradients is an important consideration in understanding mode collapse. Theoretically, G will gradually adjust its mapping of noise inputs z, and as a result adjust p_g, according p_{data}: for the images x where G assigns too much probability mass, p_g contracts by mapping less inputs z from the latent space to those images. In the areas with too little probability mass, it maps more inputs z, so as to generate certain colors and textures more often. p_g will converge to p_{data} over infinite training samples given the assumption that both image distributions are non-zero everywhere. Che et al. (2016) explain that in practice with a finite dataset, G may not receive enough gradients from the discriminator for parts of the image space. This results in some modes of p_g in the image space, called major modes, with disproportionately more probability mass than other, minor modes (Figure 3); the intuition here is that G can only learn to generate the modes for which it receives feedback from the discriminator. If G does not originally visit certain areas of the image space, it will not receive feedback to learn to generate images in those areas and continue not to visit them. And so, a cycle begins, resulting in G barely learning to create the modes it did not originally create often.

Correcting this failure case is a challenge given the GAN design; the only way for G to map more noise inputs z to minor modes in p_g is to learn from gradients. These necessary gradients come from visiting the images $G(z)$ that correspond to the minor modes or to the boundary of minor and major modes. However, by definition, the generator creates images in these modes disproportionately less often, and so it receives these gradients disproportionately less than other gradients. This occurs regardless of training set distribution; mode collapse has proved an aggravating problem even with uniform datasets. The generator's not visiting the values of z which map to minor modes enough to learn to generate them is one of the reasons for mode collapse.

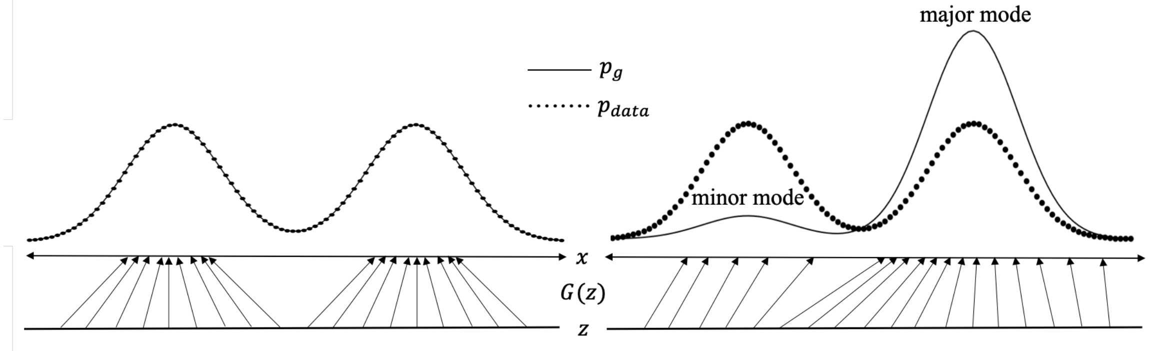


Figure 3: $G(z)$ trained on a uniform, bimodal p_{data} without and with mode collapse. Figure inspired by Goodfellow et al. 2014, Figure 1

The problem of partial mode collapse, where the generator maps different inputs to similar outputs, is equivalent to the problem of lack of diversity in the generated dataset. Diversity of a probability distribution can be quantified by its support (Arora et al. 2017, Arora and Zhang 2017, Arora 2017), or the combination of possible values the random variable can take on. For images, this would ideally correspond to individual pixel values, but since complete mode collapse is rare, it can be interpreted as corresponding to high-level features. The support, then, is the set of all possible combinations of the values that high-level features take on in the generated distribution, such as style of attire, hair color, complexion, etc. (Arora et al. 2017).

To explain how GANs perform in practice with finite network capacity, Arora et al. explain that for input images of dimension d, the number of peaks – or modes – in p_{data} can be exponential in the size of d (2017). The number of parameters p in the discriminator will, on the other hand, be much smaller. The authors present a theorem that the distance of the generated and training distributions is within ϵ when p_g is a uniform distribution over a random sample from p_{data} with size $\frac{Cp}{\epsilon^2 \log(p)}$, where C is a fairly small constant. In other words, when the GAN is near its equilibrium, p_g only selects a small set of modes from p_{data} from which to sample to fool D. Real-world discriminators do not prevent p_g from even having diverse samples, so they certainly do not encourage p_g to converge to p_{data}. Arora (2017) clarifies that this theorem also demonstrates that this is not a problem of training set size; adding more training samples can only improve the GAN objective by at most ϵ, and in a subsequent work, Arora and Zhang (2017) empirically verify that the support of the generated distribution is (only) near-linear in p.

4. Skewed Datasets

We hypothesize that mode collapse is prone to be more severe in applications using highly skewed datasets, under an assumption that the discriminator will perform differently for different modes. Let the skewed training dataset contain two modes with differing frequencies: m_1 is the mode which occurs less and m_2 is the one which occurs more. Let $G(z_1)$ and $G(z_2)$ map to m_1 and m_2 respectively. On a finite dataset, D will be worse at distinguishing between real and fake images that correspond to m_1 than m_2 (because it sees more examples of m_2), so it will be worse at understanding that $G(z_1)$ is fake than $G(z_2)$. In other words, D assigns $G(z_1)$ a higher likelihood of being real, so $D\big(G(z_1)\big) > D(G(z_2))$. For the non-saturating game used in practice, this results in $G(z_1)$ receiving smaller gradients than $G(z_2)$, since $J^G = -\log\big(D(G(z))\big)$ gets flatter when $D(G(z))$ approaches 1.

This thesis highlights that the modes that GANs collapse to align with the majority modes in the original data. This is an interesting claim because it is well-studied that the mode collapse in GANs depends on the prior distribution p_z of the latent input to the generator, which has nothing to do with the distribution p_{data} of the training set. Few related works studying mode collapse and the bias and generalization of GANs do not make a distinction between uniform and non-uniform datasets, presumably because mode collapse is a pernicious problem for all training datasets, not just skewed ones. The only work we know of is Mishra et al.'s experiments to showcase that the divergence between p_g and p_{data} does indeed get worse as the training data is more skewed (2018): they train various GAN variants on a dataset with two classes from the MNIST dataset at a 50:50, 70:30, and 90:10 distribution. Using three clustering metrics and the Fréchet Classification Distance to analyze how close the distributions are, they report a trend that the GANs' ability to match the training distribution decreases as the training set's skew increases.

Our work verifies this result with one GAN variant, but it uses a different dataset and metric to evaluate the divergence between distributions. Mishra et al.'s metrics are four scalar values which do not provide much insight on *how* the distributions differ; their work verifies that the effects of mode collapse are more drastic with skewed datasets but does not show which modes the generator collapses to. To qualify the difference in GAN performance across a uniform and skewed dataset, we follow the framework introduced by Zhao et al. (2018) to project the support of the original and generated distributions to a low-dimensional feature space (where, in this case, the number of features is just two: size and color).

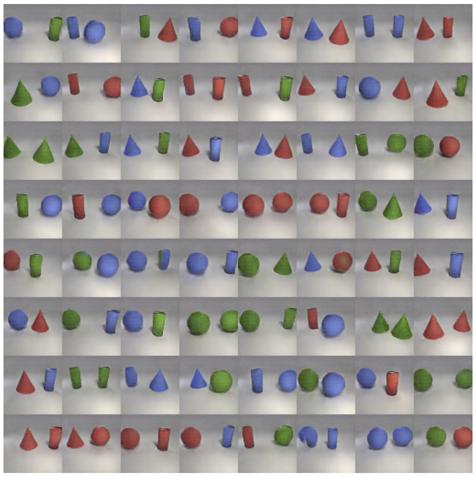
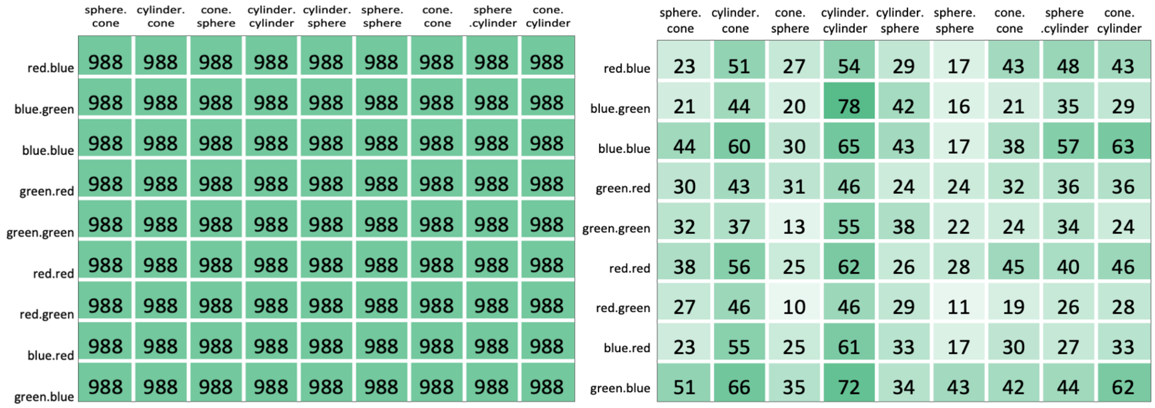


Figure 4: A random sample of a DCGAN-generated distribution trained on the two-object CLEVR dataset

One of their experiments uses a two-object CLEVR image dataset. CLEVR (Johnson et al. 2017) is a synthetic dataset of images of objects for visual reasoning, designed to artificially control biases. These objects may vary in size, color, shape, and material. Johnson et al. provide code to generate new CLEVR images using Blender, a 3D computer graphics programming tool, which we utilize for our experiment. The two-object CLEVR image dataset controls two properties of two objects: their shape and color. The characterization of the distributions and measurement of their divergence, then, are not done in the image space, but are only done for two high-level features: object shape and color (Figure 8). If two objects are each allowed three colors and three shapes to take their form, the projections to the random variable in this feature space have just $(3 \times 3)^2$ possible values; a support size of 81 can be easily visualized.

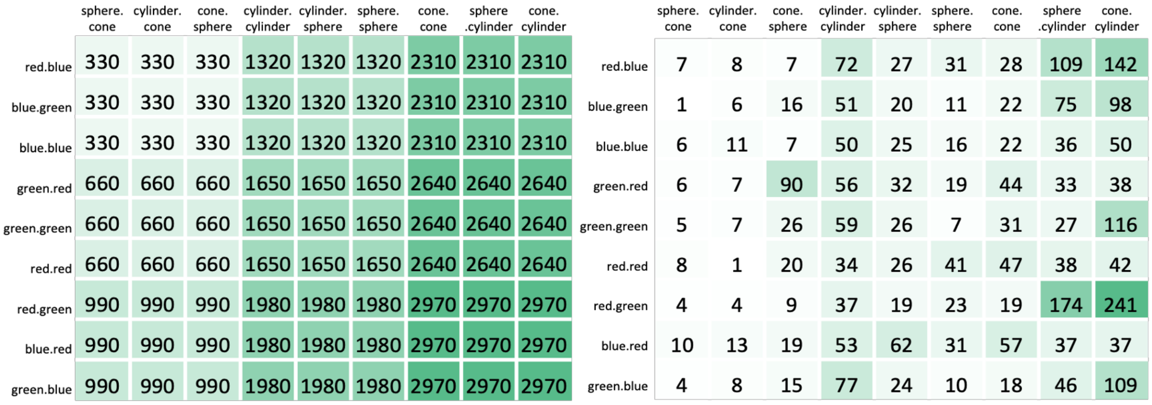


	sphere. cone	cylinder. cone	cone. sphere	cylinder. cylinder	cylinder. sphere	sphere. sphere	cone. cone	sphere .cylinder	cone. cylinder
red.blue	988	988	988	988	988	988	988	988	988
blue.green	988	988	988	988	988	988	988	988	988
blue.blue	988	988	988	988	988	988	988	988	988
green.red	988	988	988	988	988	988	988	988	988
green.green	988	988	988	988	988	988	988	988	988
red.red	988	988	988	988	988	988	988	988	988
red.green	988	988	988	988	988	988	988	988	988
blue.red	988	988	988	988	988	988	988	988	988
green.blue	988	988	988	988	988	988	988	988	988

	sphere. cone	cylinder. cone	cone. sphere	cylinder. cylinder	cylinder. sphere	sphere. sphere	cone. cone	sphere .cylinder	cone. cylinder
red.blue	23	51	27	54	29	17	43	48	43
blue.green	21	44	20	78	42	16	21	35	29
blue.blue	44	60	30	65	43	17	38	57	63
green.red	30	43	31	46	24	24	32	36	36
green.green	32	37	13	55	38	22	24	34	24
red.red	38	56	25	62	26	28	45	40	46
red.green	27	46	10	46	29	11	19	26	28
blue.red	23	55	25	61	33	17	30	27	33
green.blue	51	66	35	72	34	43	42	44	62

Figure 5: Uniform training distribution and total of three DCGAN-generated distributions

We use a Deep Convolutional Generative Adversarial Network (DCGAN, Radford, Metz, and Chintala 2015), because it is a stable architecture whose loss function is the same as the one presented in this chapter. We train three initializations of a DCGAN on a dataset of 80,028 images, uniformly distributed along combinations of shape and color. Each of the two objects can be red, blue, or green, and have the shape of a cone, cylinder, or sphere. A random sample of 1000 images from each initialization is manually assigned to one of the 81 classes (Figure 9). The DCGAN shows signs of having mode collapsed, even in this projected state; the total number of images created containing two cylinders over the three generated distributions is much higher than the total number of images containing two spheres. However, though the most-generated class appears in the generated dataset almost eight times more often than the least-generated class, most of the classes enjoy ~4-5% of the probability mass, and the degree of mode missing is slight.



	sphere.cone	cylinder.cone	cone.sphere	cylinder.cylinder	cylinder.sphere	sphere.sphere	cone.cone	sphere .cylinder	cone.cylinder
red.blue	330	330	330	1320	1320	1320	2310	2310	2310
blue.green	330	330	330	1320	1320	1320	2310	2310	2310
blue.blue	330	330	330	1320	1320	1320	2310	2310	2310
green.red	660	660	660	1650	1650	1650	2640	2640	2640
green.green	660	660	660	1650	1650	1650	2640	2640	2640
red.red	660	660	660	1650	1650	1650	2640	2640	2640
red.green	990	990	990	1980	1980	1980	2970	2970	2970
blue.red	990	990	990	1980	1980	1980	2970	2970	2970
green.blue	990	990	990	1980	1980	1980	2970	2970	2970

	sphere.cone	cylinder.cone	cone.sphere	cylinder.cylinder	cylinder.sphere	sphere.sphere	cone.cone	sphere .cylinder	cone.cylinder
red.blue	7	8	7	72	27	31	28	109	142
blue.green	1	6	16	51	20	11	22	75	98
blue.blue	6	11	7	50	25	16	22	36	50
green.red	6	7	90	56	32	19	44	33	38
green.green	5	7	26	59	26	7	31	27	116
red.red	8	1	20	34	26	41	47	38	42
red.green	4	4	9	37	19	23	19	174	241
blue.red	10	13	19	53	62	31	57	37	37
green.blue	4	8	15	77	24	10	18	46	109

Figure 6: Skewed training distribution and total of three DCGAN-generated distributions

To characterize how DCGAN performance varies with the training distribution, we also create a skewed dataset of two-object CLEVR images. This dataset of 133650 images is highly non-uniform, with some classes occurring almost nine times more often than some others. 1000 images are randomly sampled, again, from each of three initializations of DCGAN trained on the skewed dataset and classified manually. The divergence between p_{data} and p_g is much more pronounced for this skewed training distribution than for the uniform one (Figure 10). The phenomenon of missing modes is more prominent than that of actually collapsing to modes; several minority classes patently occur disproportionately less than in the original distribution. It should be noted that the third initialization of DCGAN undergoes an extreme case of mode collapse; 45 of the 81 modes are never visited even once, and the generator assigns over a fifth of the probability mass to one single class.

This experiment supports Mishra et. al's result that it is much harder for p_g to converge to p_{data} when p_{data} is non-uniform than when it is uniform, or, in other words, that mode collapse is worse when the modes are not balanced. We find that it is also easy to predict the direction of the skew, since the modes that DCGAN misses in p_g correspond to the minor modes in the original training distribution p_{data}.

In summary, mode collapse is a pernicious failure of GANs arising from the real-world constraints of the network and optimization algorithm design. It inhibits models from converging to a theoretically guaranteed equilibrium where the generator wins and successfully fools the discriminator with a realistic distribution. While a large area of research, it is still not such a well-understood phenomenon that it may be overcome by models that can scale to generating high-quality images. To summarize, there are four major causes of mode collapse in practice for skewed datasets. The first is that simultaneous gradient descent does not necessarily solve the correct minimax game and can make progress toward the generator learning to completely mode collapse. The second is that the prior of the latent space does not send enough inputs z to G which would correspond to minor modes in p_g. The third is that the diversity of p_g is limited by the capacity D. These first three are universal for all finite training sets, regardless of their distribution. The final cause that this work suggests is that D is less confident about generated images which correspond to small modes m_1 in the original dataset p_{data}, so it assigns a higher likelihood and sends smaller gradients for those generated outputs; this last cause only applies to skewed training datasets.

CHAPTER 3

IMAGINING AN ENGINEER

The purpose of this thesis is to empirically demonstrate that mode-collapsing GANs exacerbate social biases when trained on skewed datasets. The illustrative task of this work is to contrast the already-skewed real-world dataset of U.S. universities' engineering faculty faces with the generated distribution of faces the GAN imagines when asked to create engineer professors to highlight the differences in the proportions of minority classes in the two datasets, specifically surrounding feminine facial features and non-white skin tones. The hypothesis is that when trained on the engineering faculty images, the GAN will skew its generated facial images toward white males. This skew will not capture the same distribution of biases as the original data, which would actually be a best-case scenario, but will rather exacerbate them further. The proportion of white males will be higher in p_g than in p_{data}.

We again use a Deep Convolutional GAN, which incorporates design choices for the architecture to stabilize training. As expected, the discriminator and generator consist of deep neural networks which use convolutional layers. Notably, this architecture omits pooling layers (Springenberg et al. 2014) and applies batch normalization (Ioffe and Szegedy 2015) to every layer in the networks except the output of the generator and the input of the discriminator. The justification for using this GAN variant is that it is popular among machine learning practitioners; it is readily available as an off-the-shelf model. Since it uses the same objective function as the original GAN, it is well-known that it is susceptible to mode collapse, which we claim is the primary cause of social data bias exacerbation. Finally, it is the state-of-the-art model for facial images in the unconditional setting, which solves pure generation problems. In unconditional GAN variants, the generator receives random noise as input, as opposed to conditional variants where the generator receives an image as input, which solve image-to-image translation problems.

1. Experiments

The design of the experiments presented in this section revolve around the intentional use of a dataset known to have social biases: U.S. university engineering faculty headshots. The creation of a dataset ready to be used by a neural network involves three steps: collection, cleaning, and preprocessing. For data collection, we scrape images from the faculty directories of all engineering departments of 47 universities on the 2020 U.S. News “Best Engineering Schools” list. This process is automated by using a testing browser to iterate through a paginated directory and download all images on the page. Care is taken to filter the directories to exclude administrative staff and other non-faculty positions to ensure the integrity of the engineering professor dataset. To clean the dataset, each image was input to the Python face-recognition library, which uses an unsupervised object detection model, the Histogram of Oriented Gradients (Dalal and Triggs 2005), to act as an unsupervised face detector. This method divides the image into evenly spaced grids and computes the average gradients of each group by convolving the pixel values over a derivative mask. It normalizes the gradients for each group and compares the average to a known face pattern; if it finds a match, a face is detected. All images for which no face was detected are removed from the dataset. This automates the process of removing noisy images, such as ghost images or logos when a faculty directory lists a professor but has no image record, or images that are of poor quality. In a final step, the remaining images are manually examined, and noisy images with faces, such as those with multiple people, are deleted. Most images are faculty headshots with similar lighting and backgrounds. We crop the images to the face to prevent the discriminator from learning to recognize patterns in the dataset semantically unrelated to the face itself. This task is again accomplished using the face-recognition library, which returns the coordinates of the pixel groups bounding the gradients which matched to the known face patterns. We crop the images and resize them all to a consistent 64 × 64 pixels so they can be input into a neural network. This resizing is a decrease in resolution for most of the images. The final dataset consists of 17,245 facial images.

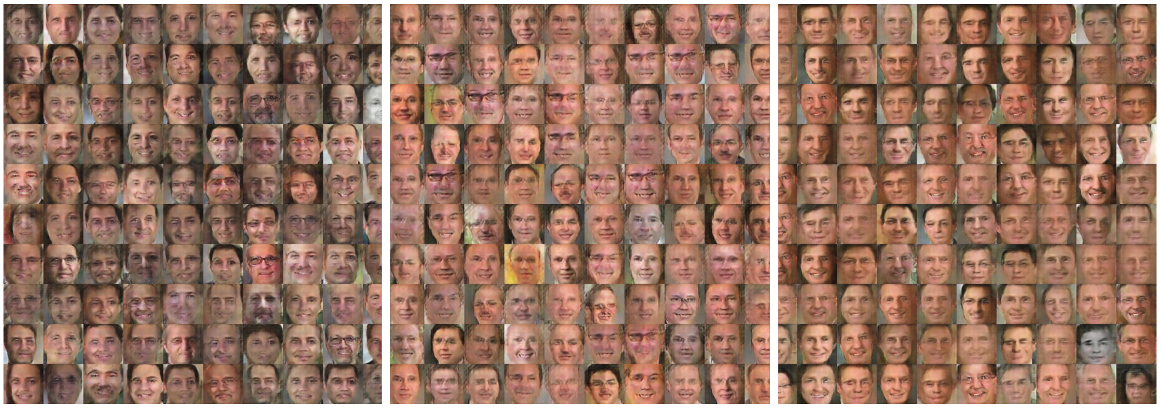


Figure 7: DCGAN-generated samples after training for 50 epochs on the engineering faculty dataset on three random initializations

We train a DCGAN for 50 epochs on three random initializations to generate three separate synthetic datasets (Figure 4). It is immediately apparent that the generator does not learn to create diverse skin colors. To evaluate the potential social biases that the generator learns to fool the discriminator, we annotate a random sample of images from each of the four distributions: the original faculty dataset p_{data} and each of the three generated distributions p_{g_1}, p_{g_2}, and p_{g_3}. We are interested in the proportion of feminine and non-white faces in the original dataset, and the average proportion of feminine and non-white faces of the three generated datasets. We do this in two ways: we ask humans through Amazon Mechanical Turk to each annotate gender and race on 50 images, and we use a commercially available classifier to annotate gender.

We recruit 132 master Turkers for a seven-minute study. Workers receive a master qualification when they earn a high reputation by completing multiple tasks. All Turkers in our study are paid $1.20 for their work. Each Turker labels a set of images uniformly sampled from one of the four distributions (with no mix-and-match). The annotation process is designed as a between-subject study: each Turker only sees images from one distribution. The Turker must select the most appropriate option from two sets of multiple-choice answers. One question is used to qualitatively assess gender appearance: “face has mostly masculine features,” “face has mostly feminine features,” and “neither of the above is true.” The second question is used to qualitatively assess race: “skin color is mostly white,” “skin color is mostly non-white,” and “can’t tell.” Each Human Intelligence Task (HIT) took an average of five minutes to complete, though a few of the workers did utilize their full seven minutes. To ensure the quality of the annotation, each Mechanical Turk worker receives 52 images to label: 50 are from one of the distributions, and two are images of celebrities – Scarlett Johansson and Idris Elba – for which the answers to both questions are trivial (mostly feminine and white for Johansson and mostly masculine and mostly non-white for Elba). The Turkers’ answers to these two questions allowed us to prune meaningless data which may have risen from the use of bots randomly selecting answers or humans hurriedly completing the task without paying attention. We pruned 18 such workers' answers, leaving 114 valid completed HITs. This resulted in 25 valid sets of annotations for images sampled from p_{data}, 30 valid sets of annotations for images sampled from p_{g_1} and p_{g_3}, and 29 valid sets of annotations for images sampled from p_{g_2}.

2. Results

The results of this experiment support our hypothesis (Figure 5). Using majority voting for classification, the percentage of images classified as having feminine features decreased from 20% in the original distribution to just 6.67% on average across the three generated distributions. The hypothesis was also supported along the axis of race: the percentage of images classified by the majority of Turkers as having a non-white skin color decreased from 24% in the original distribution to 1.33% on average across the three generated distributions. This result shows a drastic divergence, suggesting that it is likely that for one or two GAN initializations, this mode may have been missed entirely! We verify that both results are statistically significant by using a one-tailed two-proportion z-test with the following null and alternate hypotheses: $H_0: \hat{p} = p_0$ and $H_a: \hat{p} < p_0$. The p-values of this test are 0.0094 and 8.7×10^{-5} for the divergences across the dimensions of gender and race, respectively, showing statistical significance.

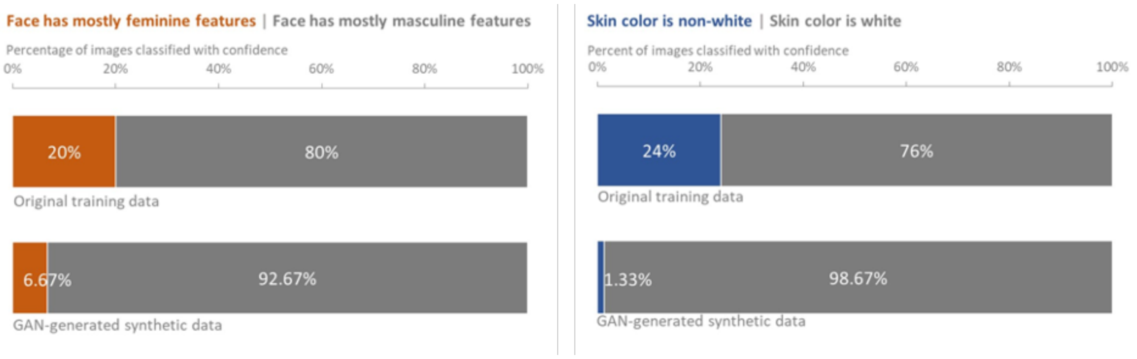


Figure 8: Distributions of classifications for gender and race from human annotations on a random sample from p_{data} and the average human annotations of random samples from p_{g_1}, p_{g_2}, and p_{g_3}, using majority voting.

We measure the confidence of the annotations by plotting how the dataset would be classified if we were to use a different thresholding technique than majority voting (Figure 6). For the original images, there is a consensus of which images depict masculine and feminine features and a consensus of which images depict white and non-white skin tones. For the generated images, the reaction of the crowd is different, presumably because DCGAN does not create photorealistic faces.

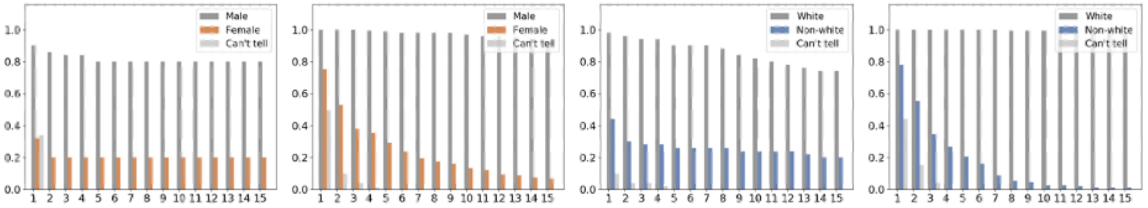


Figure 9: Classifications for gender and race on thresholding values

One interesting observation is that for every image in the generated dataset, there are at least three Turkers who label it as having mostly masculine features and at least seven Turkers who label it as having a mostly white skin color. The classification of images to having masculine or white features does not decrease considerably as the number of required votes increases, suggesting that the crowd is confident about the images they mark as masculine or white. However, the opposite is true for the classification of images to having feminine or non-white facial features. The proportion of images that the crowd classifies as having a minority feature decreases as the number of required votes increases. There are two valid but conflicting interpretations to this result. The first is that the generator is better at generating convincing masculine and white features than it is at generating convincing feminine and non-white features. This interpretation supports the idea that the generator has collapsed to the white and male modes; these images may be better quality because the generator received more gradients from these modes and had the opportunity to learn them well. The second is that humans, when asked to classify gender and race, may have a bias to default to male and female for ambiguous images. This second interpretation may be explained away by the fact that none of the images are labeled to the third, ambiguous category with confidence; the options corresponding to neither masculine or feminine for gender, and “Can’t tell” for race are not ever selected for any image by more than four Turkers, meaning that even when the workers become less decisive about whether images are female, they are not more decisive that the image is ambiguous.

The human annotation results were supplemented by using a commercially available classifier to annotate the images by gender. We use Microsoft Azure Cognitive Services’ Face API because it had the best overall accuracy in Gender Shades (Buolamwini and Gebru 2018), a study conducted to test three commercial classifiers’ performance across different groups from six countries in Europe and Africa. As this is an automated annotation process, we were able to drastically increase the number of images being labeled. We increased our sample size of each of the four datasets p_{data}, p_{g_1}, p_{g_2}, and p_{g_3} from 50 to 1000 and performed the analysis over five initializations of the DCGAN instead of three.

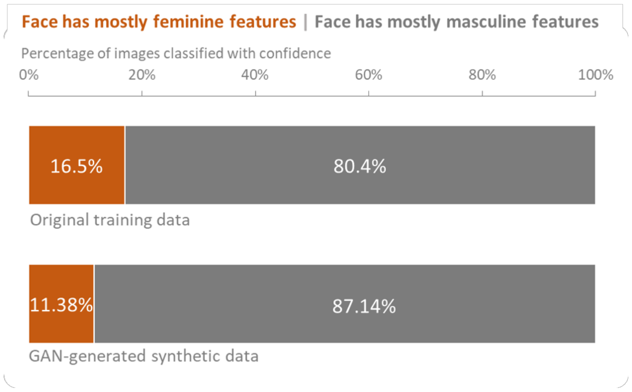


Figure 10: Distributions of classifications for gender from Microsoft Azure Cognitive Services Face API on a random sample from p_{data} and the average of random samples of five generated distributions.

The percentage of feminine features decreases from 16.5% in the original dataset to 11.38% on average over the five generated datasets (Figure 7). The results of this experiment do not show as harsh of a divergence between the original and average generated distributions as the human labeling did, but we still reject the null hypothesis that the proportions are the same, with a p-value of 6.4×10^{-5}.

The results of this experiment show that the GAN collapses along the latent dimensions of gender and race, and it learns to bias the synthetic faces toward masculine features and light skin tones when asked to "imagine" engineering faculty.

CHAPTER 4

PHOTO-EDITING LENSES: A CASE STUDY

1. Snapchat "My Twin" Lens

Thus far, this thesis has focused on just one class of GANs: the state-of-the-art unconditional variant, where the generator receives random noise from a latent variable as input. DCGAN is designed to use the same objective as the original GAN and solve the same problems. It is only the second-most popular GAN variant. Conditional variants of GANs do not solve the pure image generation problem, but rather an image-to-image translation problem to transform images from one domain to belong to another. The two domains have a close correspondence; a transformation can be found by changing the geometries minimally, if at all. Some examples of successful applications for image-to-image translation are conversion of horses to zebras, street photographs to their semantic segmentation, aerial photos to Google maps, and summer landscapes to winter landscapes. Conditional variants of GANs can also be used as an augmentation technique to abet the sparse data problem. In the domain of medical imagery, one could use these GANs to convert common images of healthy patients to rare images unhealthy patients. Conditional GAN variants can also be used with facial image data, such as to swap one's apparent gender or to edit a visage to have certain desirable attributes.

The most popular off-the-shelf GAN variant used by machine learning practitioners today (as measured by the number of stars on the most-used GitHub repositories for the model, carpedm20 2019 and junyanz 2020) is CycleGAN (Zhu et al. 2017). This conditional variant learns mappings between two unpaired datasets, where images do not exactly correspond between sets. CycleGAN minimizes several losses for two domains X and Y and two functions $G: X \rightarrow Y$ and $F: Y \rightarrow X$. The first two are GAN losses, which are traditional cross-entropy losses from the discriminator for fake images $G(x)$ and real images y, and fake images $F(y)$ and real images x. The second is a cycle-consistency loss to ensure that a mapping from one domain to another is revertible; it encourages that $G\big(F(y)\big) \approx y$ and $F\big(G(x)\big) \approx x$. Finally, to stabilize the training, a third is an identity loss. When the GAN is asked to map an image to the domain it already comes from, the GAN should not change the image, so $G(y) \approx y$ and $F(x) \approx x$.

One may expect that conditional GAN variants do not suffer the failures of mode collapse. To paint a picture, if a generator from a conditional GAN variant is trained on facial data and it sees some proportion of females to males in its training set, it is not obvious to expect that the gender proportions will change in the generated set. To do this, a GAN would have to actively work to change the gender appearance of a face; it seems it would be easier just to leave the masculine and feminine features alone. However, it is known that even conditional GAN variants whose generators have a chance to see real images from the training set are not immune to mode collapse (Ma et al. 2018). While this thesis does not perform a comprehensive evaluation of how CycleGAN-generated distributions differ along the axes of gender and race from their original training distributions as it did for DCGAN, it details a preliminary and speculative study on its applications and offers examples of potential mode collapse. The findings in the remainder of this work are intended to open discussion and further avenues for research.

Core to the brand of Snapchat is its computer-vision-assisted facial filters, or “selfie lenses” (Snapchat Support). These augmented reality lenses allow users to snap pictures of their faces with additional features warped onto their image, such as crowns or dog ears. In 2019, Snapchat released two lenses called “My Twin,” colloquially known as the gender-swap filters. These lenses aim to transform a user’s face into a quintessential male or female. The male filter noticeably squares the jawline and adds facial hair, while the female filter seems to make the chin pointed, slim the nose, and soften the facial features. Both versions of the filter are available to all users on the app. CycleGAN was known, before the release of this selfie lens, to be able to perform a “gender-swap,” or translate images of women to images of men and vice-versa. A CycleGAN used by this lens could have been trained to minimize cycle-consistency and identity losses for transformations between two domains: male faces and female faces. While Snapchat has not released its algorithm, dataset, or architecture specifications for these lenses, there seems to be common consensus in the machine learning community that this lens mostly likely takes advantage of the capabilities of conditional GANs (Yanjia 2019, Liang 2019, aDutchofMuch 2019). Of course, the lens may use traditional landmarking technologies in tandem, such as to enhance the jawline and add facial hair to images.

We present how this presumably conditional-GAN-based technology reacts to the sensitive features this work discusses. When a GAN for this lens is trained on a biased dataset of faces, it is susceptible to collapsing multiple selfie inputs to similar colors and textures. It seems that a pattern in the female “My Twin” selfie lens is that, when used on women of color, it lightens their skin tones. This is not the case for white women using the same filter. To quantify this perceived color change, we use the shades on L’Oréal Research’s skin color chart (“Expert in Skin and Hair Types,” Figure 11).

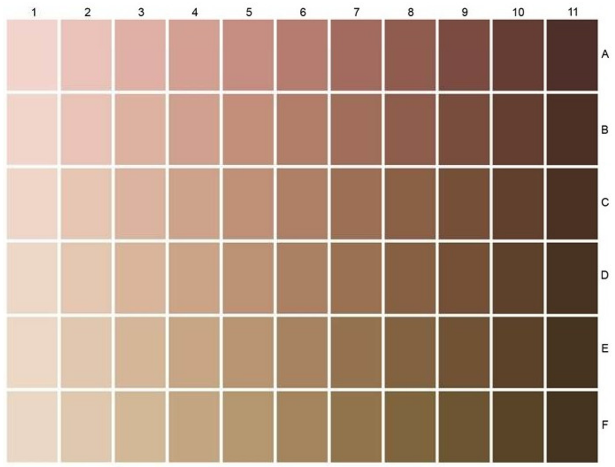


Figure 11: L'Oréal Research "A New Geography of Skin Color" chart

We collect before-lens and after-lens selfies of several women and manually discard noisy pairs of images where the position of the face changes, as this can affect lighting in the image. Machine analysis is conducted on the same section of each face, which is a manually-cropped region spanning both cheeks and falling between the eyes and mouth. An automated process finds the average pixel value of the region on this face and matches it to its closest shade in the skin color chart. Lightness in the skin color increases as one moves right in the chart, and warmth in the skin color increases as one moves down. This experiment discards the information about the skin *tone* (warmth) and is only interested in skin *color* (lightness). The selfie lens consistently lightens the skin color by one shade on the chart for women of color, while it acts arbitrarily for white women. Of the six women of color whose images we study, five faces are lightened one shade, and one remains unchanged. For the six white women whose images we study, two faces are lightened one shade, two remain unchanged, and two faces are darkened one shade.

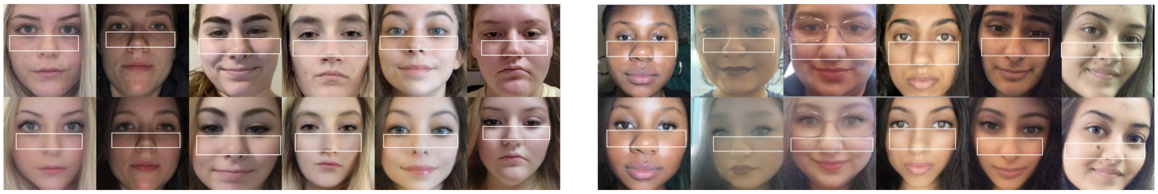


Figure 12: Transformations using the female “My Twin” selfie lens on 12 women, six white and six non-white, with the regions used for machine analysis highlighted

It is possible that lighting changes cause the arbitrary performance for white women, but this does not explain why the selfie lens has a consistent reaction to one class of inputs and an arbitrary reaction to another. A potential cause of lightening skin tones in women of color is that a GAN used by the selfie lens collapses all inputs in a region of the image space to output lighter colors. Indeed, if this technology uses a conditional GAN variant, mode collapse would be an unsurprising explanation for this phenomenon of lightening skin colors for a minority race, since we have already seen in the previous chapter that a DCGAN biased its outputs along the dimension of race by almost entirely missing non-white faces.

It must be noted that though the selfie lens was likely intended to be used across domains for gender-swap, we still have an expected performance for GANs used to translate images of women to images of women. Conditional variants of GANs typically are trained to preserve identity when asked to map images to their own domain with a regularizing term built into their loss functions. CycleGAN’s identity loss has already been described. In fact, it was inspired by the identity loss from the Domain Transfer Network (Taigman, Polyak, and Wolf 2016), a GAN-based architecture which places a heavy weight on minimizing the distance between $G(x)$ and x for all images in the dataset for a given distance metric. This suggests that a female selfie lens using a GAN which does not mode collapse should, at most, minimally change the appearance of women's faces.

There is not enough data to analyze distributions and diversity to conclude whether this would constitute exacerbation of social biases or simple perpetuation, but this observation does open an intriguing research problem. This can be studied further if Snap, Inc. confirms that it is using a GAN or provides insight into its data-collection process; white female users make up the single largest demographic of Snapchat's base (Newberry 2019). Snapchat's having trained a model on its own data skewed toward white females, for example, would explain mode collapse to light-skinned outputs from its non-white users. This work does not intend to be a final word on the matter, and concedes that much more research will need to be done in the area of mode collapse in skewed dataset for conditional GANs to make any convincing claims, but this case study opens up thought-provoking questions about social data bias perpetuation in social media applications.

2. Engineering Professor Lens

Stirred by the idea that Snapchat selfie lenses are using GAN-based image-to-image translation technology, we train our own "selfie lens" to convert regular faces to their engineering professor counterparts. Since we suspect from the previous case study that CycleGAN mode collapses in such a way that, given a skewed output domain, it actually changes the inputs in the direction of sensitive features to transform their gender and race, we hypothesize that a CycleGAN trained to translate to our engineering faculty dataset will lighten skin tones for people of color and make women's faces more masculine. Like the DCGAN, a CycleGAN will bias its outputs to make them lighter and more masculine.

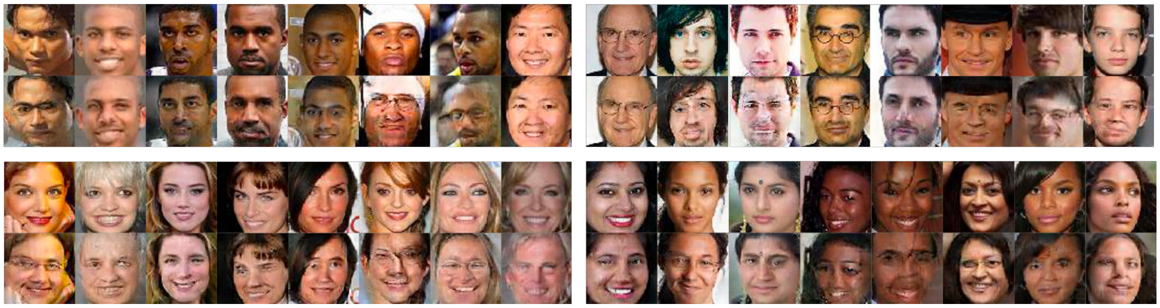


Figure 13: Random samples of the GAN transformations on the test sets, divided into four categories: non-white male, white male, white female, and non-white female.

One of our domains is the engineering faculty dataset, and the other is the CelebA dataset (Liu et al. 2018), which consists of over 200,000 annotated images of celebrities. We balance our datasets, creating two training sets of 16,500 engineering professors and 16,500 celebrities. We also hold out two test sets with four categories of 100 images each: nonwhite, white, male, and female, where there may be overlap among the categories. We run the off-the-shelf CycleGAN for 200 epochs with a batch size of 1, as is done in the original paper. We test the model to translate celebrities from the four categories to engineering professors.

Even when these transformations are randomly sampled, it is evident that the GAN learns to lighten skin tones for people of color, make women's faces more masculine, and largely leave white male faces untouched in comparison (Figure 13). We also present illustrative sets of transformations to emphasize the transformations made by the CycleGAN on female (Figure 14) and non-white (Figure 15) individuals. In addition to making changes along gender and race, the GAN also learns to add glasses, create smiling facial expressions, and dramatically increase the apparent age.

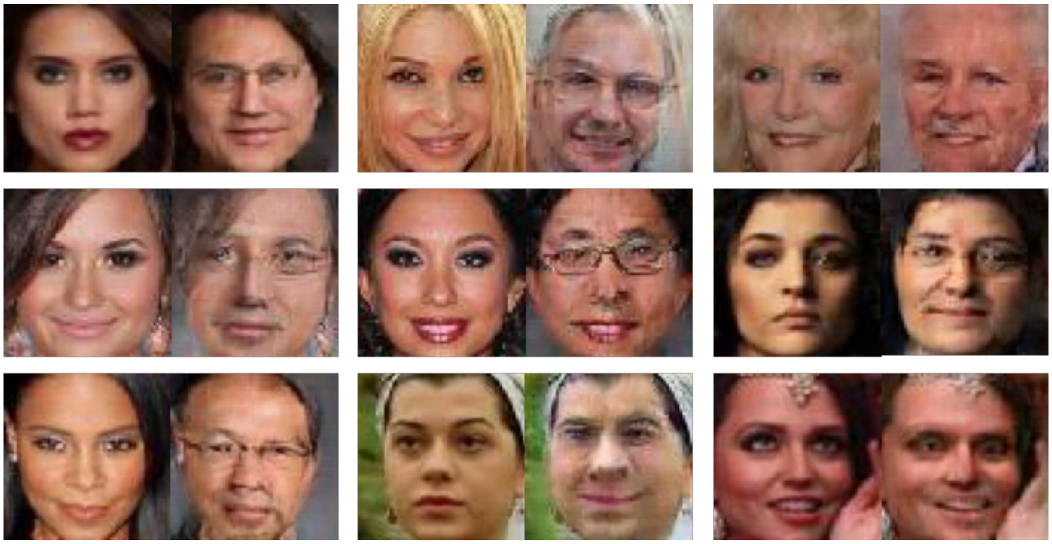


Figure 14: Illustrative test set of transformations on female celebrity faces

As discussed in the introduction, we can expect a GAN to perpetuate and exacerbate biases along *all* dimensions where there exists a skew in the training set; some axes' skews have innocuous explanations: the faculty directories comprise mostly headshots, where individuals usually smile wide, and members of academia are thought to be more likely to develop short-sightedness than those not in academia due to their heightened amount of close-up work and use of screens. This work highlights that the CycleGAN picks up these biases when it learns its mapping from celebrities to professors but reiterates that this kind of bias is not the focus of the work; machine learning systems are designed to find correlations in its attempt to learn patterns. This correlation-seeking is only problematic in social data when models perpetuate and exacerbate biases for minority groups that have faced systemic disadvantage or discrimination.

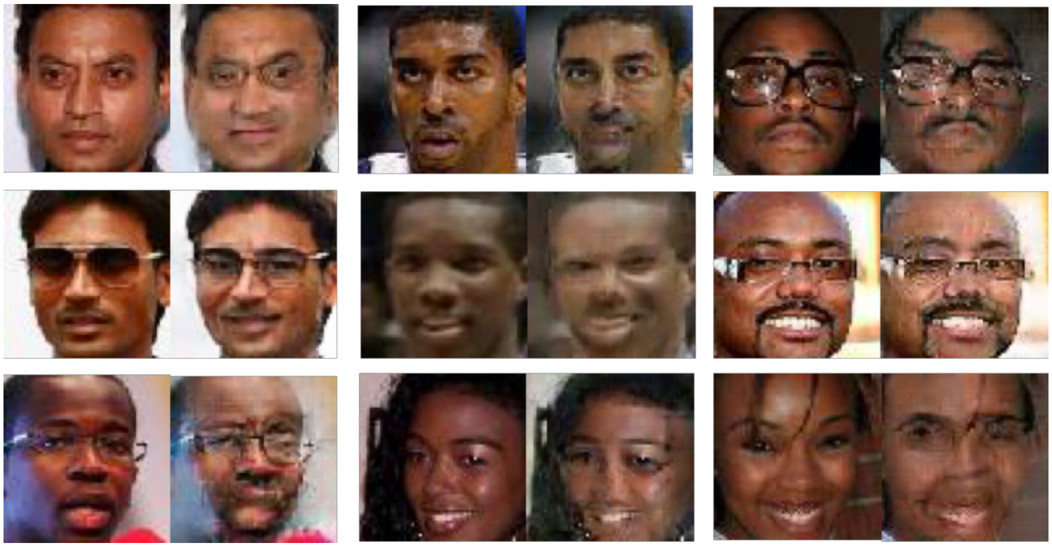


Figure 15: Illustrative test set of transformations on non-white celebrity faces

Again, this thesis has not completed exhaustive studies for CycleGAN to analyze and measure the diversity of its generated distributions and the original distributions from their training sets, nor has it conducted research to deduce whether the phenomena mentioned are result of exacerbation of bias or mere perpetuation. Both tasks would require annotating sample of the original real-world data and the generated data, either using humans or a commercial classifier. Rather, this work simply makes a reasonable case that conditional variants of GAN like CycleGAN suffer from all the previously mentioned implications of mode collapse – even though they are seemingly constrained by their inputs – because they have the ability to change the class of their input. The results of these three studies support the claim that most GANs (not just those who solve the pure generation problem) amplify biases along the direction of a skew in the training dataset.

CHAPTER 5

DISCUSSION AND CONCLUSION

Machine learning practitioners and software engineers should tread lightly when using GANs for data augmentation and data transformation. While GANs are an elegant solution to the problem of supervised generation and domain translation, they have their own drawbacks which must be carefully considered when designing experiments and applications. Their non-convergence cases are a subtle failure mode that can do more harm to the data than good. Most GANs amplify bias in skewed datasets because of this failure mode, and all datasets are skewed unless meticulously designed not to be. This work highlights how the technology exacerbates social biases, so social media applications ought to use GANs responsibly.

Colorism has pervaded across the corners of the Earth through the centuries (Hunter 2007). For Asian communities, its beginnings are found in early hierarchical structures: the wealthy “leisure” class who were fortunate enough to not need to work in the sun tended to have lighter skin. For Latin communities, its beginnings are found in European imperialism: in Latin America, creoles, mestizos, and mulattoes all had higher places in the social order. For African-American communities, its beginnings are found on U.S. plantations before the Civil War: lighter-skinned slaves were treated better than darker-skinned ones by being given jobs inside the house instead of working on the fields. When light skin is seen as being more “beautiful” today, it carries the weight of those historical, racial associations of light skin with power.

Since GANs have a technical failure which forces them to amplify these harmful biases along sensitive features, they should be used with the utmost care, if at all, in social applications. Snapchat is a social media application whose largest demographic is young users aged 13-24; they comprise 90% of its user base. Over 250 million users are active on the app daily ("Snapchat by the Numbers"). Snapchat caters mostly to impressionable users and holds the reins in controlling youth perceptions. The release of a GAN-assisted selfie lens adds fuel to the already-present narrative that Snapchat's beautification lenses lighten skin tones. The messaging that girls and young women receive when using this app, even subconsciously, is that beauty and femininity depend on skin color and superiority.

Women have seen uncountably many successes in legal battles for civil rights in the past century but still find themselves running an obstacle course of inequality in the workplace. The most well-known disparity is the wage gap: white women receive 82% of what white men do, and Hispanic women receive a meager 58% (Akhtar 2019). Women also undergo social motherhood penalties when men receive fatherhood bonuses; mothers face an additional pay gap to childless women because they are considered less committed to work when they have children, whereas fathers receive more pay compared to other childless men (Budig 2014). Women find they are valued less and are often blocked by their supervisors (Marcus 2016). Women face sexual harassment. These effects are especially destructive for women of color.

When a GAN uses social data characterizing a profession skewed against women, such as used by our engineer-imagining DCGAN, a technical failure will force it to amplify the same gender biases that already create a lack of opportunities for women. Applications using techniques which further bias along the dimension of gender are underrepresenting women even further, working to erase their presence in fields that they have struggled to climb their way into. The CycleGAN trained to transform faces to look like engineers learned a foolproof method to do so for women: convert them into men. Mode collapse is a technical failure of GANs to force it to diverge from the real world, but since it occurs along the same direction as the original data, it also sheds light to the social biases that exist in the real world. Women often find that their gender is the largest obstacle to their success.

As a final remark, this work highlights social biases as a means to understanding a technical failure of GANs, but it must be reiterated that GANs susceptible to mode collapse will amplify any bias in the original data. This work uses a facial dataset of engineering faculty to showcase this result because humans are better able to understand and visualize social biases than those which may exist in, say, chest x-rays. This result generalizes to all domains, and care must be taken that data augmentation does not further underrepresent minorities regardless of the application, especially in life-or-death scenarios; one area where this technique has been awarded accolades is in the domain of medical imaging.

Medical imaging is, of course, data collected from real humans, so any medical dataset can contain all the same social biases as a facial dataset, subject to the same bias exacerbation and mode missing from GANs. Machine learning practitioners use GAN-based data augmentation as a technique to “balance” their datasets between healthy and unhealthy examples. In their study, Salehinejad et al. use a DCGAN-based data augmentation technique to train a classifier to detect pathology from chest x-rays (2018). Frid-Adar et al. also use a DCGAN to generate synthetic medical images of liver lesions (2018). Though these machine learning practitioners are hopeful that the use of GAN-generated synthetic images will balance the biases in their data, they are sure to amplify latent biases along dimensions among the classes that they might not even be aware of. Unfortunately, since the medical image domain is so esoteric and inaccessible, it may be difficult to identify and analyze the biases created, perpetuated, and exacerbated by GANs; identifying the features along which medical biases can exist is not trivial. We urge machine learning practitioners to carefully consider the implications of most GANs' susceptibility to mode collapse, and even urge them not to use GANs in those applications where creating additional bias would pose a direct disadvantage for any minorities in the dataset, placing preference on data-augmentation techniques known to better align to the original distribution.

REFERENCES

aDutchofMuch. “r/MachineLearning - [D] Is the New Snapchat Gender Filter GAN-Based?” Reddit, 13 May 2019, www.reddit.com/r/MachineLearning/comments/bo4orw/d_is_the_new_snapchat_gender_filter_ganbased/.

Akhtar, Allana. “Lower Pay, More Harassment: How Work in America Failed Women of Color in the 2010s.” Business Insider, Insider Inc., 18 Dec. 2019, www.businessinsider.com/how-work-is-failing-women-of-color-2019-10.

Arora, Sanjeev, Andrej Risteski, and Yi Zhang. " Do GANs Learn the Distribution? Some Theory and Empirics." International Conference on Learning Representations. 2018.

Arora, Sanjeev, et al. "Generalization and equilibrium in generative adversarial nets (gans)." Proceedings of the 34th International Conference on Machine Learning, Vol. 70. 2017.

Arora, Sanjeev. “Generalization and Equilibrium in Generative Adversarial Networks (GANs).” Off the Convex Path, 30 Mar. 2017, www.offconvex.org/2017/03/30/GANs2/.

Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein gan." arXiv preprint arXiv:1701.07875. 2017.

“Best Engineering Schools.” U.S. News, 2019. https://www.usnews.com/best-graduate-schools/topengineering-schools/eng-rankings.

Budig, Michelle J. “The Fatherhood Bonus and The Motherhood Penalty: Parenthood and the Gender Gap in Pay.” Third Way, 2 Sept. 2014, www.thirdway.org/report/the-fatherhood-bonus-and-the-motherhood-penalty-parenthood-and-the-gender-gap-in-pay.

Buolamwini, Joy, and Timnit Gebru. "Gender shades: Intersectional accuracy disparities in commercial gender classification." Conference on Fairness, Accountability and Transparency. 2018.

carpedm20. “carpedm20/DCGAN-Tensorflow.” GitHub, GitHub, Inc., 12 Sept. 2019, github.com/carpedm20/DCGAN-tensorflow.

Che, Tong, et al. "Mode regularized generative adversarial networks." arXiv preprint arXiv:1612.02136. 2016.

Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Vol. 1. IEEE, 2005.

“Expert in Skin and Hair Types around the World - L'Oréal Group.” L'Oréal. L'Oréal Group, www.loreal.ca/en-ca/research-and-innovation/when-the-diversity-of-types-of-beauty-inspires-science/expert-in-skin-and-hair-types-around-the-world.

Frid-Adar, Maayan, et al. "Synthetic data augmentation using GAN for improved liver lesion classification." 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018). 2018.

Goodfellow, Ian. "NIPS 2016 tutorial: Generative adversarial networks." arXiv preprint arXiv:1701.00160. 2016.

Goodfellow, Ian, et al. "Generative adversarial nets." Advances in Neural Information Processing Systems. 2014.

Grnarova, Paulina, et al. "Evaluating Gans via Duality." arXiv preprint arXiv:1811.05512. 2018.

Mishra, Deepak, et al. "Mode matching in GANs through latent space learning and inversion." arXiv preprint arXiv:1811.03692. 2018.

Hunter, Margaret. "The persistent problem of colorism: Skin tone, status, and inequality." Sociology compass 1.1 (2007): 237-254.

Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." arXiv preprint arXiv:1502.03167. 2015.

Johnson, Justin, et al. "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.

junyanz. “Junyanz/Pytorch-CycleGAN-and-pix2pix.” GitHub, GitHub, Inc., 13 Apr. 2020, github.com/junyanz/pytorch-CycleGAN-and-pix2pix.

Lala, Sayeri, et al. "Evaluation of mode collapse in generative adversarial networks." High Performance Extreme Computing. 2018.

LeCun, Yann. "The MNIST database of handwritten digits." 1998, http://yann.lecun.com/exdb/mnist/

Liang, Lavinia. “The Dark Implications of Facial Swap Filter Technology.” PAPER, Paper Communications, 11 Sept. 2019, www.papermag.com/snapchat-gender-swapping-filter-2638765039.html?rebelltitem=12#rebelltitem12.

Liu, Ziwei, et al. "Large-scale celebfaces attributes (celeba) dataset." Retrieved 15 Aug. 2018.

Ma, Shuang, et al. "Da-gan: Instance-level image translation by deep attention generative adversarial networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

Marcus, Bonnie. “It's Obstacles Women Face In The Workplace - Not A Lack of Ambition - That Causes Them To Opt Out.” Forbes, Forbes Media LLC, 15 Aug. 2016, www.forbes.com/sites/bonniemarcus/2016/08/15/its-the-obstacles-women-face-in-the-workplace-not-a-lack-of-ambition-that-causes-them-to-opt-out/#7321e4a12667.

Metz, Luke, et al. "Unrolled generative adversarial networks." arXiv preprint arXiv:1611.02163. 2016.

Newberry, Christina. “Top Snapchat Demographics That Matter to Social Media Marketers.” Top Snapchat Demographics That Matter to Social Media Marketers, Hootsuite Inc., 17 Sept. 2019, blog.hootsuite.com/snapchat-demographics/.

Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434. 2015.

Salehinejad, Hojjat, et al. "Generalization of deep neural networks for chest pathology classification in x-rays using generative adversarial networks." 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018.

Shah, Nisarg. “CSC304: Lecture 5. Game Theory: Zero-Sum Games, The Minimax Theorem.” Algorithmic Game Theory and Mechanism Design, 25 Sep. 2017, University of Toronto. Microsoft PowerPoint presentation.

Silva, Thalles. “An Intuitive Introduction to Generative Adversarial Networks (GANs).” FreeCodeCamp, 7 Jan. 2018, www.freecodecamp.org/news/an-intuitive-introduction-to-generative-adversarial-networks-gans-7a2264a81394/.

Snapchat Support. “How to Use Lenses.” Snap, Inc., support.snapchat.com/en-US/a/face-world-lenses.

Springenberg, Jost Tobias, et al. "Striving for simplicity: The all convolutional net." arXiv preprint arXiv:1412.6806. 2014.

Srivastava, Akash, et al. "Veegan: Reducing mode collapse in gans using implicit variational learning." Advances in Neural Information Processing Systems. 2017.

Taigman, Yaniv, Adam Polyak, and Lior Wolf. "Unsupervised cross-domain image generation." arXiv preprint arXiv:1611.02200. 2016.

O'Gorman, L., and Kasturi, R. Document Image Analysis, volume 39. IEEE Computer Society Press Los Alamitos, 1995.

"Snapchat by the Numbers: Stats, Demographics & Fun Facts." Omnicore Agency, 7 Feb. 2020, https://www.omnicoreagency.com/snapchat-statistics/.

Yanjia Li, Ethan. "Gender Swap and CycleGAN in TensorFlow 2.0." Ethan Yanjia Li, 18 Dec. 2019, yanjia.li/gender-swap-and-cyclegan-in-tensorflow-2-0/.

Zhao, Shengjia, et al. "Bias and generalization in deep generative models: An empirical study." Advances in Neural Information Processing Systems. 2018.

Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." Proceedings of the IEEE international conference on computer vision. 2017.

APPENDIX A

GLOSSARY OF TERMS

conditional variant – GAN model which receives deterministic input. For images, a conditional GAN solves an image-to-image translation problem of adapting styles to convert images of one domain to another similar one, such as night to day, photograph to painting, or zebra to horse

high-level feature – a feature understood by humans but not directly given as input to a machine learning model

mode – a local maximum in a probability distribution. For images, a mode is a category of colors or textures which occur in the distribution

mode collapse/missing – a non-convergence case of GANs where the generator fails to capture some modes which were present in the original training distribution; equivalently, a lack of diversity in the generated distribution

minor mode – a mode that is assigned less probability mass in a distribution than another; when sampled uniformly from a distribution, values of a random variable which occur less frequently than other

sensitive feature – social biases with a history of discrimination or inequity, especially toward a minority class

skewed dataset – a dataset with minor and major modes; datasets are skewed along some dimensions of high-level features unless carefully designed not to be

support – the set of unique values a random variable takes with non-zero probability. For facial images, this can be understood as the combinations of position, lighting, expression, gender, race, age, accessories, hair color, attire, etc. present in a distribution; a metric for diversity which does not account for the probabilities assigned to values of a random variable

unconditional variant – GAN model which receives random noise as input. For images, an unconditional GAN solves a pure image-generation model; images that a generator outputs are not constrained to comprise any particular colors or geometries