Fair Generation Through Prior Modification

Eric Frankel*
Department of Mathematics
Stanford University
ericsf@stanford.edu

Edward Vendrow*
Department of Computer Science
Stanford University
evendrow@stanford.edu

Abstract

Machine learning models that learn from biased datasets often display the same bias in their results; this is particularly true for inherently unsupervised methods like generative adversarial networks. However, designing and evaluating fair generative model architectures is difficult due to the prohibitive computational requirements of training such models. In this project, we explore methods for post-hoc modification of generative models for producing fair dataset representations, with particular emphasis given to approaches that can be performed without human supervision.

1 Introduction

As machine learning models have become more common in daily life, their decisions can have significant implications for an individual's life. *Protected groups*, or individuals with certain *protected* or *sensitive* characteristics like race, gender, or religion [1], have historically been subject to biased treatment in machine learning models implemented in practice, including recidivism prediction [2], gender stereotypes in word embeddings [3], and skin cancer prediction failures for dark-skinned individuals [4]. As such, preventing discrimination, or non-favorable outcomes as a result of an individual's protected attributes [5], has become increasingly important in machine learning models in production.

Researchers have increasingly examined the data machine learning models are trained on as a source of bias and unfairness that can be propagated through the model [6]. This is particularly pressing for generative models, wherein its supervised learning process learns potentially sensitive attributes latently. As a result, it is more difficult to control for existing bias in the dataset, and data sampled from generative models will likely reflect existing biases from the dataset. This is problematic for applications of generative models requiring data generation, which include text-to-speech [7], pose-guided image generation [8], and text-to-image synthesis [9].

Initial methods of learning generative models for fair data generation focused on creating fair tabular datapoints through explicit supervision of protected attributes during training [10, 11]. Newer approaches have used weak supervision with unlabelled image datasets with latent sensitive attributes [12]. However, both approaches require training generative models from scratch rather than using pre-trained generative models trained on biased datasets, and [12] suffered reduction in generated sample quality. Indeed, generative models that have performed demonstrated high-quality image synthesis, like StyleGAN [13] or BigGAN [14], take on the order of weeks for training for high-resolution image generation, even when using up to 8 V100 Nvidia GPUs. Training generative models can also be prohibitively unstable [15], further increasing both the time and computational power required to train a fully expressive GAN. More generally, many image synthesis models exist that can produce high-quality but biased data.

In this work, used a neural network to adapt the prior of a generative model to generate fair data, the first described use of a post-processing fairness method for generative models. We tested our

*Equal contribution.

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

approaches on the datasets MNIST [16] and the CelebA-HQ dataset [17]. We first establish the efficacy of our methods on a DCGAN [18] trained to produce 0 and 1 digits from the MNIST data set. After demonstrating the success of our methodology on this test dataset, we implement our results in generating digits from a DCGAN trained on all 10 digits of MNIST, achieving high quality generated images. Finally, we implemented our results on a pretrained ProGAN model that generated CelebA faces and demonstrated equality across gender.

2 Preliminaries

2.1 Fair ML

There has been a growth in the amount of work that address fairness in machine learning, with particular focus given to statistical methods of enforcing different definitions of fairness [19]; these different notions of fairness include demographic parity, equality of odds, and equality of opportunity. Nonetheless, in the context of generative models and data generation, fairness can take on several different forms. For example, in [12], they used the *fairness discrepancy* for their generative model p_θ with respect to an unbiased distribution p_{unbias} and sensitive attributes u as

$$f(p_{\text{unbias}}, p_\theta) = |\mathbb{E}_{p_{\text{unbias}}}[p(u|x)] - \mathbb{E}_{p_\theta}[p(u|x)|]$$

Alternatively, in [10], the authors aimed to enforce the fairness criterion $\mathbb{P}(y = 1|u = 1) = \mathbb{P}(y = 1|u = 0)$ for a sensitive attribute u.

Notably, however, the fairness criteria used in these papers fall largely along the lines of one of the main methods for enforcing fairness: imposing fairness constraints during the training of a model. However, in our case, we aim to enforce some notion of post-processing fairness for our models; as a result, we must more carefully investigate the precise definition of fairness we wish to enforce.

2.2 Generative Adversarial Networks

Generative adversarial networks (GANs) are generative models with two components: a generator and discriminator. The generator $G(z)$ samples a prior distribution that is usually uniform or Gaussian noise to generate samples. The generator aims to learn a distribution p_θ that matches the data distribution p_{data} that is ideally representative of the population as a whole. The discriminator of the GAN $D(x)$ is a binary classifier that predicts whether its input x is generated from $G(z)$ or real data. Altogether, the value functions of a generative model can be summarized as a minimax game

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

When learning generative models, it is generally assumed that we have access to the requisite amount of unbiased data that can be sampled independently from a target unbiased distribution. However, in practice this is rarely the case, and the i.i.d. assumption rarely holds. Additionally, for high-dimensional data like images, the required sample complexity for learning a truly faithful representation can exceed practical capacity.

3 Methods

3.1 Investigating Bias

We begin by assessing the presence of bias in samples generated from generative models trained on face image datasets. In particular, we used the generative adversarial networks StyleGANm trainied on FFHQ, and ProGAN, trained on CelebA. We selected face generating models because of their public availability as well as the variety of potentially sensitive characteristics present in face images. We extracted binary features corresponding to the labels in the CelebA dataset (Young, Blond, Brown Hair, etc.) via a pre-trained MobileNetV2 architecture [20] fine-tuned on the CelebA dataset through the methods used in [21].

The different features present in the CelebA dataset occur with distinct frequencies, with "no beard" being the most common feature and "bald" being the least common feature. This disparity in feature

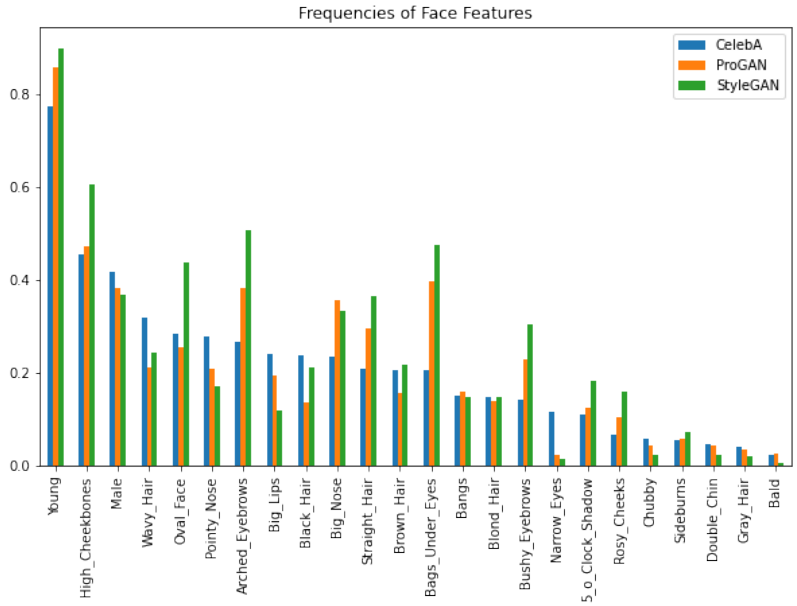

Figure 1: Frequencies of features identified using a trained MoblieNetV2 model. The frequencies of these features in samples generated from StyleGAN and ProGAN are similar to that of the original CelebA dataset, demonstrating the propagation of bias from the dataset to the output of the trained generative models.

	All	Chubby	Young	Blond	Black Hair	Brown Hair	Double Chin
Male	0.448	0.096	0.767	0.004	0.284	0.074	0.087
Female	0.552	0.027	0.952	0.128	0.212	0.260	0.017

Figure 2: Gender differences in features from images generated according to the standard distribution X in StyleGAN. There are clear gender disparities in StyleGAN; for example, men are much more likely to be 'Chubby', while women are much more likely to be blonde.

frequency is best seen in Figure 1. Some features occur with high frequency, while others are almost never seen.

In addition to varying feature frequencies overall, these models also demonstrate different feature frequencies across sensitive classes, and in particular across the example of male and female gender classes. To show this, we use the standard Gaussian prior $X \sim \mathcal{N}(0,\, 1)$ to generate 2500 images using StyleGAN. These generated images are passed through our feature recognition network to get feature frequencies. We can use the results to get preliminary information on biases present within StyleGAN, or rather the data set used to train it. As an example of this bias, we observe in Table 2 how features differ greatly in likelihood across male and female generated faces. For example, female faces are much more likely to be identified as "blond", while male faces are much more likely to be identified as "Chubby". While many of these relationships might make sense, some of the relationships result in undesirable biases that we would like to mitigate.

The frequency of generated features is also affected by the prior distribution used to generated images. To show this, we demonstrate that changing the prior distribution drastically affect class representation in generated images. We now sample our latent vectors from a second distribution $Y \sim \mathcal{N}(-2,\, 1)$ which has the same standard deviation as the original distribution X but with a different mean. Comparing images generated from X and Y show a remarkable difference, as demonstrated in 3. To visualize this difference, 4 shows images generated from both distributions. Samples generated according to Y tend to overwhelmingly be female and blonde compared to samples generated according to X.

By changing the prior distribution, we are able to change feature frequencies. This result suggests an important finding that it is possible to change the prior to affect feature frequencies. In particular,

Feature	Bald	Black Hair	Blond	Brown Hair	Male	Smiling	Young
Prior X	0.009	0.245	0.073	0.177	0.448	0.753	0.869
Prior Y	0.0	0.011	0.314	0.188	0.13	0.791	0.864
% Change	**-100.0%**	**-95.6%**	**+331.9%**	+6.3%	**-71.0%**	+5.0%	-0.6%

Figure 3: A table showing frequencies of various features in face images generated by prior distributions X and Y as well as the percentage change in feature frequency from X to Y.

Figure 4: Gallery of 25 images generated by $X \sim \mathcal{N}(0,\, 1)$ (left) and $Y \sim \mathcal{N}(-2,\, 1)$ (right). Note the remarkable difference in the kinds of faces each distribution generates.

it may be possible to change the prior distribution to meet a fairness criteria or generate images according to an arbitrarily desired distribution.

3.2 Fairness via Latent Perturbation

Traditionally, when learning generative models, the priors that are chosen tend to be simple, whether for tractability or ease of use[22]. However, both [23] and [24] propose learning complex priors at during training as to ensure better stability of learning generative models and greater expressivity of the underlying distribution. While we won't emulate the adaption of the prior at train time, we will similarly want to learn non-trivial (i.e. uniform or Gaussian) representations of the prior for fair generation.

We propose the use of a small neural network to implicitly learn the prior distribution to slightly perturb latent variables drawn from the prior distribution to a modified distribution. This network L accepts a latent variable z from the prior distribution and slightly perturbs it to $L(z) = z'$, following a modified distribution under which the desired fairness is achieved from the generative model. This approach eliminates the need for any further work once the latent network is properly trained: once this network is "attached" in front of a generator to form a combined network, the modified generator operates identically to the original, except with different outputs. Latent codes used to generate data may still be drawn from the same original distribution, since the latent perturbation network modifies these priors.

In addition to modifying the latent variables for fair generation, it is also important to preserve quality and variability in the final model so that the new prior is useful. To that end, we propose a three-headed training network combining three losses: $\mathcal{L}_{fairness}$ measuring fairness in the current model using a feature recognition network, $\mathcal{L}_{perturb}$ penalizing modified priors far from the original prior, and $\mathcal{L}_{quality}$ which uses the existing discriminator to ensure quality in the generated data. For clarity, we denote the latent network L, feature recognizer R, generator G, and discriminator D.

The fairness loss $\mathcal{L}_{fairness}$ measures the mean-squared error between the features generated by the batch and the desired, fair feature vector f. For instance, the feature vector for a fair binary

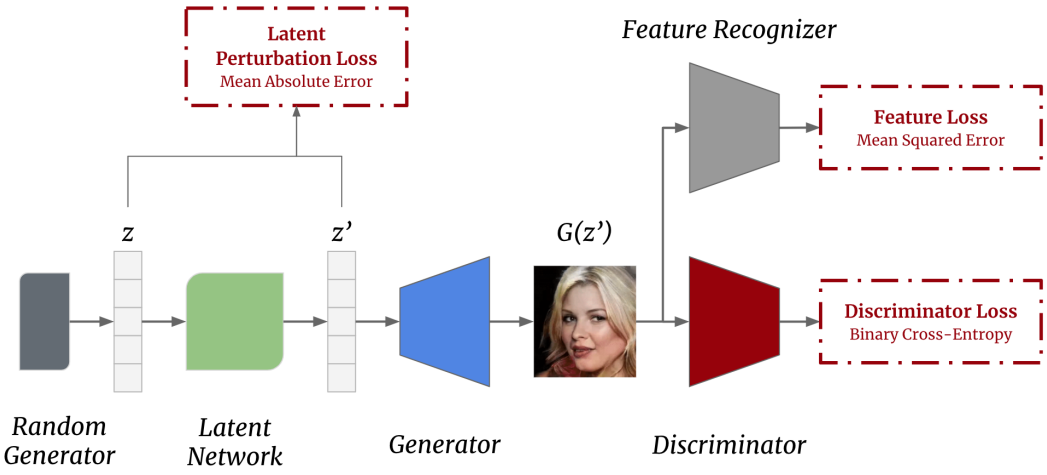

Figure 5: Our proposed architecture uses three heads to ensure fairness, quality, and variability.

classification would be $[0.5, 0.5]$.

$$\mathcal{L}_{fairness} = \frac{1}{n}\sum_{i=1}^{n} \|f - R(G(L(z_i)))\|_2^2 \tag{1}$$

The perturbation loss $\mathcal{L}_{perturb}$ ensures variability in the final output. Since the prior distribution used to train a generative model necessarily expresses the highest variability in generated data [source needed?], it suffices to penalize perturbed latent variables distant from the original latent variable. This loss is thus defined as the L_1 error between the original and perturbed latent variables.

$$\mathcal{L}_{perturb} = \frac{1}{n}\sum_{i=1}^{n} \|L(z) - z\|_1 \tag{2}$$

Finally, we take advantage of the existing discriminator model to penalize images of low quality, and thus likely low discriminator scores. Since we want all examples to fool the discriminator, binary cross-entropy loss reduces to

$$\mathcal{L}_{quality} = -\sum_{i=1}^{n} \log(D(G(L(z_i)))) \tag{3}$$

Our final loss is then a weighted sum of these losses, expressed as

$$\mathcal{L} = \mathcal{L}_{fairness} + \lambda_1 \mathcal{L}_{perturb} + \lambda_2 \mathcal{L}_{quality} \tag{4}$$

Figure 5 details the architecture using this loss to train the model. After passing through the latent perturbation network and generator, the resulting generated data is passed to both a discriminator to get $\mathcal{L}_{quality}$ and a feature recognizer to get $\mathcal{L}_{fairness}$. In the training phase, the weights for all networks are frozen except for the latent network. Note that although the discriminator and feature recognizer are needed at training time, they are no longer needed at inference time.

4 Experiments

4.1 MNIST Experiments

While many GAN models are trained on faces, running experiments on faces is inherently more difficult, as the distribution itself is significantly larger and the feature space is larger with three channels. Additionally, models trained to recognize specific "human" features of faces like race, gender, etc. have varying degrees of success due to both the dimensionality of the dataset as well as

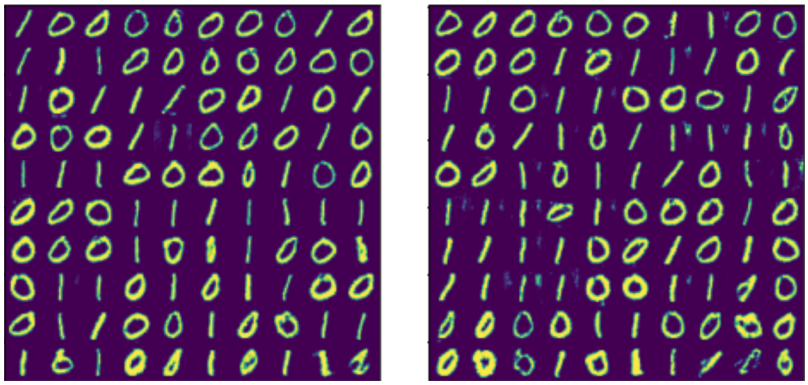

Figure 6: Samples of our DCGAN network trained on zero/one handwritten digits, before and after modification via our proposed method. Each image shows 100 randomly generated samples. The numbers are sorted by quality as determined by the discriminator network, from top left to bottom right.

the potentially subjective labelling of the different "human" features. Moreover, images created from generative models often retain artifacts after generation, which can "confuse" the model trained to recognize specific features. These challenges make it difficult to evaluate the efficacy of our designed model and the effects of our latent network.

Rather than using face images, we performed several initial experiments with generative models trained on the MNIST handwritten digit dataset. This dataset offers a much simpler image-based alternative with clear and reliable labeling, making it easier to work with. Furthermore, compared to face feature recognition models with varying accuracy, standard convolutional models perform nearly perfectly in digit recognition on MNIST. In order to generate high quality digit samples, we trained a DCGAN [18] on the MNIST dataset for 30. We also used a pretrained digit recognition model achieving neraly perfect classification accuracy. After training, the generator and decoder weights were frozen. For our first experiment, we trained a DCGAN to produce either 0's or 1's, and for our second experiment, we trained a DCGAN to produce all 10 MNIST digits.

In our first experiment, we found that an unaltered DCGAN trained to generate 0's and 1's generated 0's approximately 42.9% of the time, demonstrating the bias in the generative model over which digit is frequently produced. However, after implementing and training our proposed latent model to perturb the latent input to the DCGAN generator, the altered generator produced 0's approximately 50.2% of the time and correspondingly 1's approximately 48.2% of the time. These results are visualized in Figure 6. Examining the perturbations of the latent vectors reveals that the output distribution is only slightly different from that of the original; with randomly generated latents coming from the uniform distribution $U(-1, 1)$, the mean perturbation of the latents after being passed through the latent network was 0.073, while the median perturbation was 0.054. These encouraging results indicate that latents are only slightly altered in order to attain the desired distribution.

In our second experiment, we trained a DCGAN on all 10 digits found in the MNIST dataset. The unaltered DCGAN generated digits at unequal frequencies, as displayed by Figure 7. Indeed, digits 1 and 0 were generated most frequently, while the remaining digits were produced noticeably less frequently. After training our proposed latent model, we found that the altered generator produced digits approximately in line with a uniform distribution over the ten; these results are likewise found in Figure 7. Similar to before, we also computed the perturbation of the new distribution generated by the modified latents, and found that the mean perturbation of the latents was 0.185, while the median perturbation was 0.134. These results again indicate that the latent vectors are only slightly modified in order to attain the desired distribution. Note that these perturbations are larger than those from our two-digit experiments. We believe this occured because the latent mappings must now be learned for ten digits rather than for two digits.

Since the images produced from this GAN constituted an approximation, we were also able to calculate the FID of our output distribution from the original distribution of MNIST. Using an unmodified DCGAN, the FID score tends towards approximately 50; using our modified generator

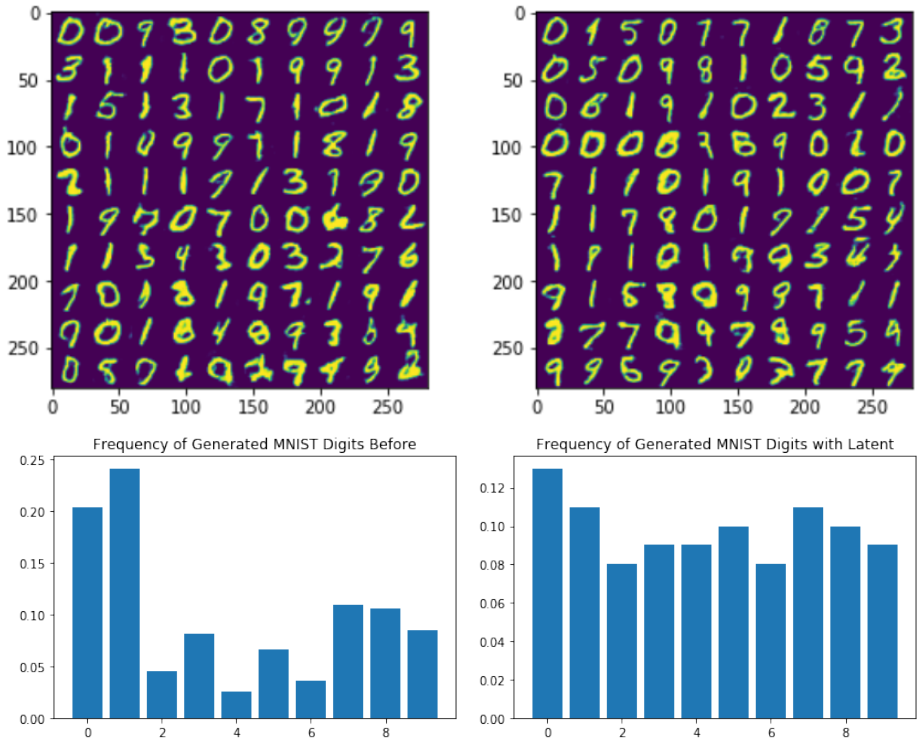

Figure 7: Top: Samples of our DCGAN network trained on the entire handwritten digit dataset, before and after modification via our proposed method. Each image shows 100 randomly generated samples. The numbers are sorted by quality as determined by the discriminator network, from top left to bottom right. Bottom: Frequencies of digits produced without and with the latent network.

including our latent network, our FID score was approximately 70 after 40 epochs. This result indicates that our output distribution closely reflects that of the original MNIST distribution, but retains a higher score because it does not precisely model the same frequencies of digits displayed in the original MNIST distribution.

4.2 CelebA Dataset

Following the effectiveness of our model's experiments using GANs trained on MNIST, we used a ProGAN model [25] pretrained on CelebA [17] to generate several samples from latents uniform on $U(-1, 1)$. We used the model used in [21] as the recognizer model, which is built off of MobileNetV2 [26], for face features labeling and implemented our latent model as described in the methods. We aimed to balance the images produced by the gender identified by the recognizer model. Using the pretrained ProGAN model alone, 63% of generated images were labeled as female; however, after generating samples using a combination of the latent model and the ProGAN generator, our the proportion of images identified as male by the recognizer model was 49.4%. Samples of this model can be found in Figure 8.

5 Discussion

In this paper, we discuss a novel post-processing fairness method for balancing the outputs of a generative model by training a network to perturb latent vectors. We evaluated this method on both the MNIST and CelebA datasets and found that the modified generator network was able to produce balanced datasets according to the identified class labels. Moreover, these balanced datasets were produced through only slight modification of the latent vectors originally sampled from a $U(-1, 1)$, indicating that the latent network learns precise mappings between relevant labels and areas of the latent space.

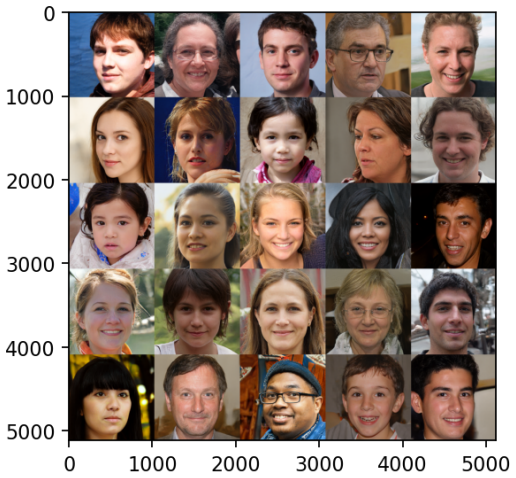

Figure 8: Samples of the pretrained ProGAN network in conjunction with our latent model. These 25 samples were randomly selected from 10,000 gender-balanced images.

Given the rapidly increasing training costs for state-of-the-art GANs, we believe that this approach will allow for highly customizable modification of generated images without needing to expensively retrain the model. Indeed, beyond enforcing notions of fairness, this method allows for the generation of images to fit the desired proportion of a particular dataset, which can improve methods for data augmentation across tasks with many class labels.

5.1 Future Work

In future work, we would aim to further reproduce the methods identified in [12] and [10]. In particular, in [12], the authors demonstrated the relationship between changing ϵ-satisfying fairness constraints with different GAN metrics like FID and Inception score; we would likewise want to demonstrate a similar relationship relating the latent network with the generator network to changing values of FID. Likewise, in [10], the authors explored the effects of classifier accuracy when trained off of their fair dataset. It would be interesting to implement similar results for the samples generated by our original generator with our latent network attached.

References

[1] Reuben Binns. Fairness in machine learning: Lessons from political philosophy. *arXiv preprint arXiv:1712.03586*, 2017.

[2] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, 5(2):153–163, 2017. PMID: 28632438.

[3] Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 4356–4364, Red Hook, NY, USA, 2016. Curran Associates Inc.

[4] Adewole S. Adamson and Avery Smith. Machine Learning and Health Care Disparities in Dermatology. *JAMA Dermatology*, 154(11):1247–1248, 11 2018.

[5] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*, 2019.

[6] Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. *A Deeper Look at Dataset Bias*, pages 37–55. Springer International Publishing, Cham, 2017.

[7] Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C Cobo, Florian Stimberg, et al. Parallel wavenet: Fast high-fidelity speech synthesis. *arXiv preprint arXiv:1711.10433*, 2017.

[8] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Advances in Neural Information Processing Systems*, pages 406–416, 2017.

[9] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

[10] D. Xu, S. Yuan, L. Zhang, and X. Wu. Fairgan: Fairness-aware generative adversarial networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 570–575, 2018.

[11] Prasanna Sattigeri, Samuel C Hoffman, Vijil Chenthamarakshan, and Kush R Varshney. Fairness gan: Generating datasets with fairness properties using a generative adversarial network. *IBM Journal of Research and Development*, 63(4/5):3–1, 2019.

[12] Aditya Grover, Kristy Choi, Rui Shu, and Stefano Ermon. Fair generative modeling via weak supervision, 2019.

[13] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[14] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

[15] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[16] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[17] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

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

[19] Sahil Verma and Julia Rubin. Fairness definitions explained. In *2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*, pages 1–7. IEEE, 2018.

[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[21] Luca Anzalone, Paola Barra, Silvio Barra, Fabio Narducci, and Michele Nappi. Transfer learning for facial attributes prediction and clustering. In *International Conference on Smart City and Informatization*, pages 105–117. Springer, 2019.

[22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[23] Hui-Po Wang, Wen-Hsiao Peng, and Wei-Jan Ko. Learning priors for adversarial autoencoders, 2019.

[24] Thomas Goerttler and Marius Kloft. Learning a multimodal prior distribution for generative adversarial nets. 2019.

[25] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[26] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.