Kenfack, P. J., Arapov, D. D., Hussain, R., Kazmi, S. M. A., & Khan, A. M. (2021). *On the Fairness of Generative Adversarial Networks (GANs)*. https://doi.org/10.48550/arXiv.2103.00950

University of Bristol - Explore Bristol Research
General rights

On the Fairness of Generative Adversarial Networks (GANs)

Patrik Joslin Kenfack*, Daniil Dmitrievich Arapov†, Rasheed Hussain‡, S.M. Ahsan Kazmi§, and Adil Khan¶,
{*p.kenfack, †d.arapov}@innopolis.university
{‡r.hussain, §a.kazmi,¶a.khan}@innopolis.ru
Innopolis University, Innopolis, Russia.

Abstract—Generative adversarial networks (GANs) are one of the greatest advances in AI in recent years. With their ability to directly learn the probability distribution of data, and then sample synthetic realistic data. Many applications have emerged, using GANs to solve classical problems in machine learning, such as data augmentation, class imbalance problems, and fair representation learning. In this paper, we analyze and highlight fairness concerns of GANs. In this regard, we show empirically that GANs models may inherently prefer certain groups during the training process and therefore they're not able to homogeneously generate data from different groups during the testing phase. Furthermore, we propose solutions to solve this issue by conditioning the GAN model towards samples' group or using ensemble method (boosting) to allow the GAN model to leverage distributed structure of data during the training phase and generate groups at equal rate during the testing phase.

Index Terms—Generative Adversarial Networks, Fairness, Group Imbalance, Representation Bias.

I. INTRODUCTION

Machine learning (ML) aims at reproducing human intelligence, for example by learning patterns in data and trying to generalize them. When data are labelled the training process is supervised, i.e. each training instance has a label and the goal of the machine learning model is to learn patterns from these data and predict labels of new instances. When labels are not provided, the training process is said to be unsupervised and the model will try to discover patterns in data. ML models are widely used nowadays to solve different tasks, such as pattern recognition [1], natural language processing [2], anomaly detection [3], action recognition [4]–[6], scene classification [7], etc. However, in general the success of these models depend on features used into the training process [8], [9]. Although these features might be provided via features engineering, which requires a lot of work and in certain domains difficult to extract manually, the need for the ML models to extract meaningful features from the data by themselves has become a hot topic within the Artificial Intelligence (AI) community [1], [8]. This is where *representation learning* (typically Deep Leaning) come in. Basically, the goal is to identify and extract useful information from data to facilitate or improve the predictions, or even generate new data. Generative model is one of the most promising approach in this direction. The main idea behind generative model, is to capture the underlying distribution of the data in order to be able to sample most realistic examples from that learned distribution.

Among the most popular generative approaches, we have Variational Auto Ecoders (VAEs) [10], [11] that consist in two neural networks: an encoder and a decoder trained jointly. The encoder tries to encode the prior distribution of the data into a latent space. The decoder tries to reconstruct original examples using examples sampled from the learned distribution with the goal to minimize the reconstruction loss. Although VAEs are simple to train, they suffer from the assumption of an approximate posterior distribution of the observed data. Thus, this makes VAEs less efficient in generating data that most reflect the real distribution of the observed data (e.g poor images quality). Generative Adversarial Networks (GANs) [12] solve the shortcoming of VAEs by trying to identify the real distribution of the observed data using an approach inspired by the Game Theory, the goal to find Nash equilibrium between the two networks, Generator and Discriminator. The generator generates most realistic examples as possible in order to deceive the discriminator, which the aim is to distinguish between the real and the generated (fake) data. Generator and discriminator are trained in competition until Nash equilibrium is reached, which means that the generator was able to generate data similar to the actual observed data such that the discriminator wasn't able to make the difference between them. Indeed, GANs and its different variants are considered as the most interesting breakthrough in AI for the last decade, because of its outstanding data generation capacity, and and its ability to generate diverse types of data. i.e it allows to generate most realistic images [13], natural language generation [14], music generation [15], medical data generation [16], fair representation learning [17] etc.

More importantly, the wide use of AI in different domain in real life has shown that ML models might exhibit unintended behaviour such as learning biases in data, even amplified them or generate new biases. i.e a model trained using data containing sensitive attributes such gender with disproportionate representation, will tends to be biased toward one groups [18]. Thus, fairness has become another important research focus within the ML community. The fairness of important technologies such as GANs must be particularly analysed. In general, dataset may contain different groups of data with different distributions, so it is important to analyse and ensure that rather than prioritizing a single group, the generator of GANs models is capable of generating examples of different groups at an equal rate. Several fairness notions have proposed

in the literature. Among them, *statistical parity* and *individual fairness* [18] are widely used. Individual fairness requires that similar individual must receive the same outcome, while statistical parity requires that classifier predictions outcome must be independent to the protected attributes. Bias in data can exist in many shapes and forms, some of which can lead to unfairness in different downstream learning tasks [19]. Representation bias is a common form of bias that leads to unfairness decisions, it happens from the way we define and sample from a population. i.e. if certain groups in the population are not represented or underrepresented.

In paper this we extensively analyse GANs models with respect to their ability to generate examples from different groups present in dataset at an equal rate. To do that, we first present experiments that show that is in most cases, GANs are unable to generate example from different groups and always converge to a single group distribution.

The key contributions of this paper are the following:

- We highlight and demonstrate the shortcomings of GANs in reproducing data from different groups, which may lead to fairness concerns i.e generate biased synthetic data.
- Point out the negative impact of inherent biases existing in GANs generation process, on downstream tasks such as classification.
- We propose a GAN boosting strategy to generate data samples from different groups at equal rate, by creating diverse ensemble of generators.

The rest of this paper in organised as follows: Section II introduces different GANs models and the theory behind. Section III provides the experiment results, that highlight the inability of GANs to generate examples belonging to different group distributions even thought they are were well balanced in the dataset. Section V presents possible solutions and research directions to fix this issue and finally the discussions and the conclusion.

II. OVERVIEW OF GENERATIVE ADVERSARIAL NETWORKS

Inspired by the game theory, GANs model consist in two neural networks (generator and discriminator) trained in adversarial manner. The generator's (G) goal is to learn the underlying distribution (p_g) of the dataset by sampling as most realistic examples as possible such the discriminator (D), which the goal is to distinguish between generated examples and real examples, will not be able to make to difference between examples $G(z)$ sampled from G using the random noise vector z and real examples x. Thus, G and D play a minimax game with value function $V(G, D)$:

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

The parameters in G and D will be updated simultaneously during the training until $D(G(z)) = 0.5$, which means that the global optimal solution in achieved and D cannot identify

the different between these two distributions. However, this original GANs model has presented some limitations and several extended variants have been proposed. Among these limitations, the problem of mode collapse is the one that has attracted the most attention. This problem occurs when the model can only fit a few modes of the data distribution, while ignoring the majority of them. In this regards, Arjovsky et. al [20] proposed Wasserstein GAN (WGAN) to improve stability of traditional GANs and resolve problems like mode collapse.

The original GANs used Multi Layers Perceptron to train the generator, since Convolutional Neural Networks (CNN) have shown the most interesting results in extracting feature from images, [21] proposed Deep Conditional Generative Adversarial Networks (DCGANs). Although DCGANs is able to learn good representation of images and provides great performances in image generation tasks, it suffers from the fact that the generator can collapse in training mode, and thus produces limited varieties of samples [21]. Mirza and Osindero [22] proposed Conditional Adversarial Networks (CGANs), which extends GANs by adding extra information c in both the generator and the discriminator, such that the generator could generate examples constrained by c e.g. class conditional samples. c can be class label, or any kind of auxiliary information. Therefore, the objective function is defined as for GANs as follow:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x|c)] \\ + \mathbb{E}_{z \sim p_z}[1 - \log D(G(z|c))]$$ (2)

It has been shown adding side information can significantly improve the quality of generated samples. Several extensions of CGANs have been proposed for diverse purpose, for instance Auxiliary GAN (ACGAN) [23] in which the extra information c is not added to the discriminator but instead the discrimination perform additional tasks such as predicting the side information c i.e outputs the class label of training data, therefore ACGAN use modified version of the loss function to increase the predictive performance for the class label.

Stacked GANs (SGAN) [24] uses a particular architecture, in which several encoders (DNN), generators and discriminators are put together. Encoders are stacked and pre-trained in a bottom-up manner. The input image is passed through the first encoder, which output a high-level representation and pass it to next encoder and so on. The uppermost encoder will output the class label. Provided with pre-trained encoder, generators are stacked and trained independently and jointly [1] in top-down manner. Each generator takes as inputs high level representation (from the encoder) and a noise vector and outputs lower-level representation. At each level, one discriminator distinguishes generated representation from "real" representation. The main advantage of SGAN is that it introduces intermediate supervision using representation discriminator at different representation level.

---

[1]*Samples generated by a preview generator is used as input of the next generator*
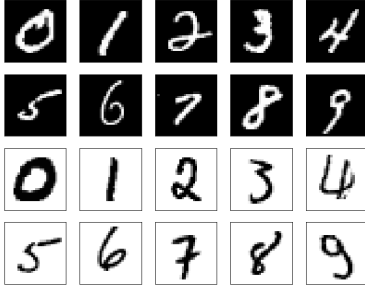
Fig. 1: Example of 2 groups created from MNIST dataset. *White digits* having white background and *black digits* with black background

Fig. 2: Example of groups created from SVHN dataset. First two rows show samples from the first group (normal samples) and last two rows the second groups (darkened)

III. THE SHORTCOMINGS OF GANs IN GENERATING DATA FROM DIFFERENT GROUP

In the section we discuss applicability of GANs model in generating data in a settings where the training data contains different groups with balance or imbalanced representation. We first describe the experimental setup and then present the results on different dataset.

A. Experimental setup

This study focuses on two widely used dataset: The MNIST [25] dataset, which contains 70,000 labelled examples of 28 x 28 images of hand-written digits, and the SVHN [26] dataset, which contains 73,257 labelled examples (10 classes) of 32 x 32 colored images of house number digits. CelibA is a large-scale and diverse face attribute dataset, which contains 202,599 images of 10,177 identities.

To create different groups, in MNIST dataset, we artificially inverted colors of certain images from black to white (Fig. 1) with different white-to-all ratio (30%, 50%, 70%). We trained different generators on the datasets with different representation ratio of groups, and analyse their ability to recover the distributions of both groups at the same rate. Similarly, for

SVHN dataset, we darkened certain images (Fig. 2). We have considered samples brightness as a sensitive attribute, forming new group out of SVHN samples, divided by 2.

B. Results on MNIST dataset

On MNIST dataset, we trained SGAN model with equal representation of different groups in the dataset (black and white digits) and analysed the distribution of groups inside the data generated by the generator. We used mean of the sample to decide which group each example belongs to: if the mean is greater than 0.5, then the sample is considered white and black otherwise. On overall, even thought groups were equally represented in the training dataset, the model was not able to generate white and black digits at equal rate. Figure 3 presents joint distribution of generated digits means. We trained generators 10 times, sampling 100 images per each of 10 classes. Figure 3 shows distribution of samples over all 10 experiments.

For instance Fig. 3(a) shows that majority of samples are white, while in Fig. 3(b) majority are black. In most cases the group distribution is not well balanced. The desired case is the one in which the group distributions are the same (as in Fig. 3(j)). However throughout our experiments, this case has rarely occurred.

The discrepancy is even more serious if the groups' representations are not well balanced in training set. We designed different experiment settings where, the GANs model was trained with imbalance groups representations. In the first setting, the model was trained using a dataset containing 70% of white digits and 30% of black digits, and the opposite in the second setting. In most cases, out of a thousand examples generated, most are from the well represented group. This behaviour shows groups are not treated equally by the model. Figure 4 shows the group distributions of examples generated using the trained model with the imbalanced group representation.

C. Results on SVHN dataset

On SVHN datset we trained almost the same SGAN, as reported in the Fig. 5, we trained 5 different generators using the same dataset. Group representations were balanced within the training data. However, during the testing phase, the examples sampled from the trained generator had a different distribution. This suggests that the generator converged to a particular group during training (Fig. 5).

D. CelibA dataset

CelibA dataset contains 61.4% female faces versus 38.6% male faces. We generated different images using Progressive Growing of GANs (PGAN) model, which is good in generating high resolution images [13], and it turns out that well-represented groups (female faces) are the most generated by the generator, while there is a lack of dark-skinned faces throughout several runs of the trained generator (Figure 6). Moreover, most of the few black faces generated were of worse quality.
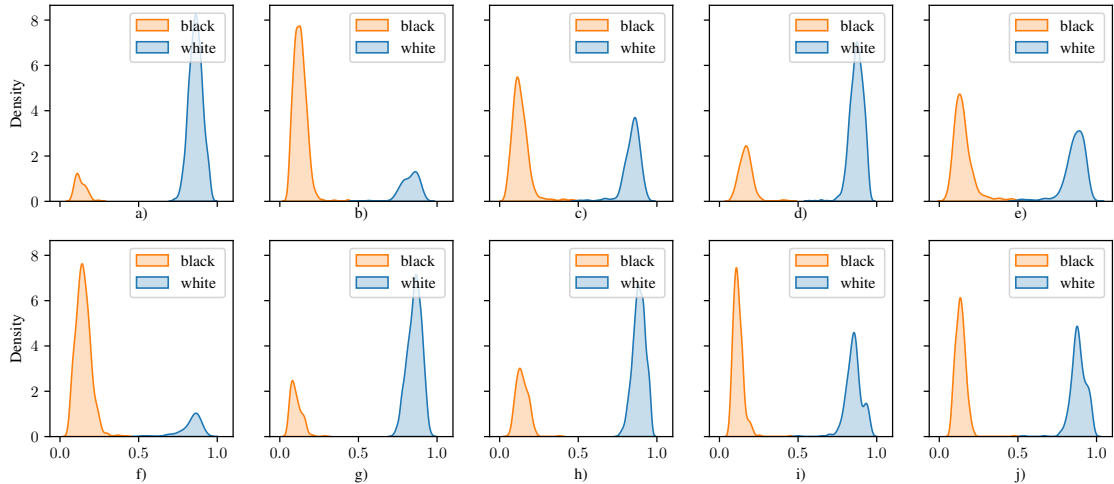
Fig. 3: Distribution of digits generated by Stacked GAN trained on the modified MNIST dataset with balanced group representations: black digits ($mean < 0.5$); white digits ($mean > 0.5$). The generator evenly produces digits from both groups, but not in most cases (a, b, c, d, f, g).
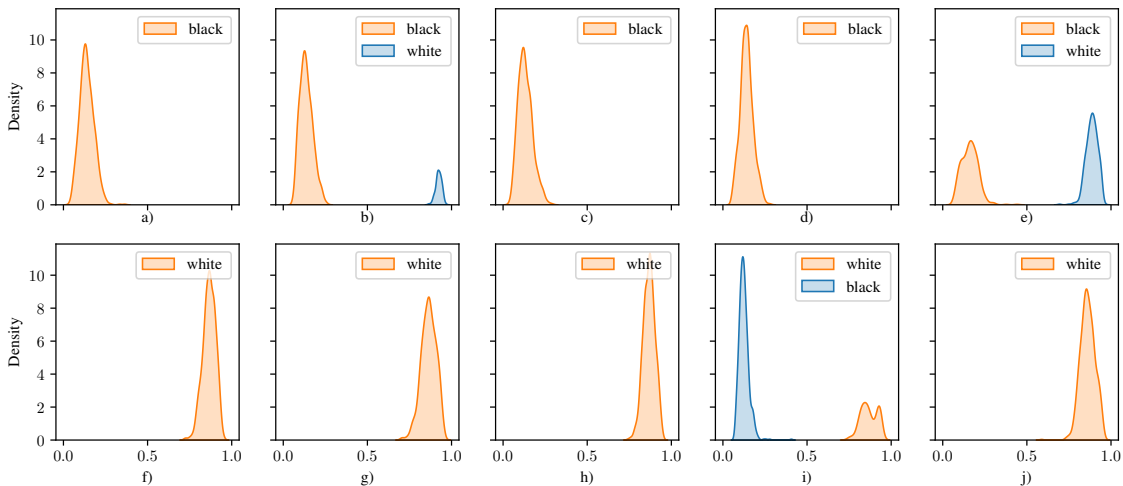

Fig. 4: Distribution of digits generated using GAN trained on MNIST dataset with imbalance group representation. (a, b, c, d, e) dataset with 70% of black digits vs 30% of white digits. (f, g, h, i, j) dataset with 30% of black digits vs 70% of white digits. The generator mainly generates digits that belong to the well representation group.

IV. DISCUSSIONS

In this section, we examine the impact that the inherent biases produced by GANs models can have when they are used to generate data for other downstream tasks, such as data augmentation, class imbalance problem and fair representation learning.

A. *Class imbalance problem & Data augmentation*

GANs model can be used to increase the size of training dataset when there is not enough data available. For example,

dataset of credit card fraud will normally contains more records of normal transactions than fraudulent ones. Similarly, database of medical information will contains higher number of healthy patients than affected ones. For classification tasks, when different classes are not uniformly distributed within the dataset, the accuracy can be significantly impacted negatively. Indeed, classifiers using this kind of dataset are biased towards the underrepresented classes. The good generative properties of GAN models gives the possibility to restore the dataset balance by generating examples from minority-class (oversam-

synthetic dataset, that could be then used for a given task e.g classification.

There are various applications of GANs in this way. Tanaka et. al [27] analysed the performances of GANs model when used to balance the dataset and generate synthetic data for classification tasks, they showed that synthetic data generated by GAN can in some cases lead to better accuracy performance than real dataset. For instance, Giovanni et. al [28] used to restore the balance of an imbalanced dataset. Other applications to balance dataset for Speech Emotion Recognition [29].s

In this paper, we argue that despite the fact GAN provides interesting property for data augmentation, their inability to generate different groups that are uniformly distributed may rise fairness concerns. As the results of the previous section showed, GAN model can inherently tends to converge to prefer certain groups in the dataset and therefor generates a much larger number of instances from these groups. Thus, when used for dataset augmentation, this may leads to the creation of biased dataset, i.e. dataset in which different groups are not equally represented. It has been shown that classifiers trained using this kind of dataset may lead to unfair decisions [19].

B. Using GANs for fair representation learning

Adversarial training has been used to learn "fair" representation of data. The goal of this approach is to mitigate unfairness by obstructing information concerning a protected attribute (Gender, age, religion etc...) as much as possible, while preserving other useful information [30]. The objective of the leaned representation is to achieve desired fairness notion such as statistical parity or individual fairness in downstream tasks as classification. The potential problem with approaches that use adversarial learning to mitigate unfairness is the bias toward subgroups. In fact, when the adversarial model attempts to mitigate unfairness by removing all information about a given sensitive attribute, there is a risk that the generator will only converge to certain subgroups. For instance, if the sensitive is gender, with each group having subgroups (for example, pregnant women) then the generator may focus on the main subgroups (non-pregnant women) and then unfairness will shift away from a male-female problem to a pregnant and non-pregnant problem.

V. PROPOSED SOLUTIONS

In this section, we discuss possible solutions that could help solve the problem presented in the previous sections.

A. Using Conditional GANs

One possible solution to solve the stated issue is to use Conditional GANs [22]. Instead of conditioning the generator and discriminator on the class label y, one can condition them with the groups label and necessary class label. The trained generator will then allows to generate samples conditioned by group label. e.g. generate images of imaginary actors of given gender (group) like male or female. This solution needs to be explored as quality of generated data. A limitation of this approach is that biases may shift to subgroups.
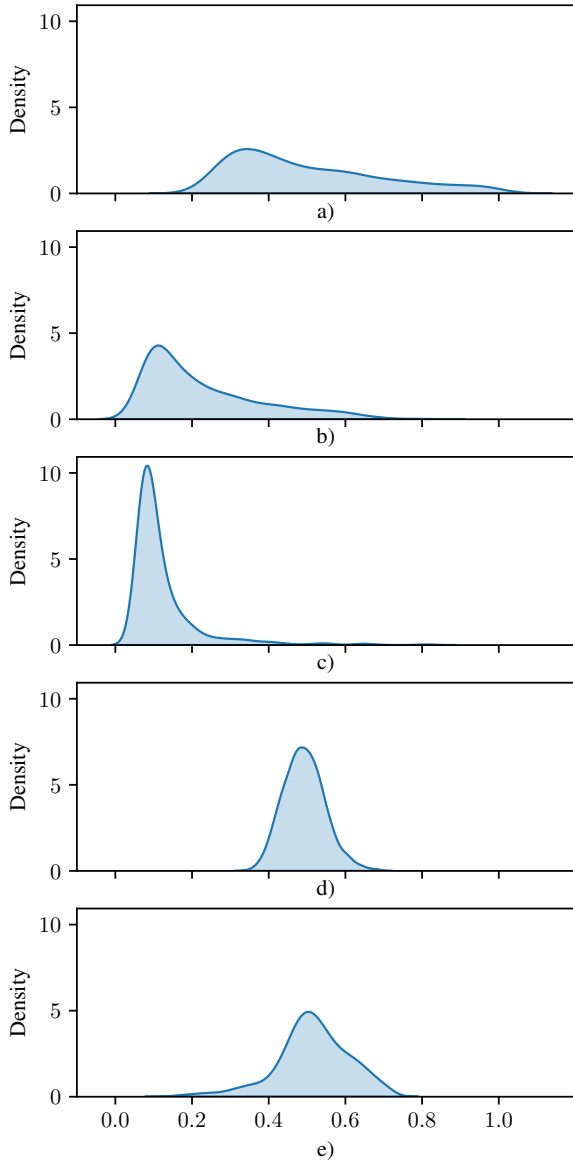

Fig. 5: Results on the SVHN dataset. In most cases the generated examples are concentrated around one group.

pling).

Another useful application of GAN models for data augmentation is when due to privacy concerns, real dataset cannot be used directly. For example, medical records could contain a lot of sensitive information about patients that cannot be shared or are protected. GAN can be use in this case to generate

Fig. 6: Images generated using PGAN. Overall the generator predominately produces pictures of females (well represented groups) with lighter skins tones.

B. Using model ensemble

Another interesting direction to explore is ensemble learning. The main idea behind ensemble model is to train different model and combine them to produce one model that outperform all of them. Boosting is an ensemble method technique widely for classification tasks. The idea is to create successive model during the training process, in which a newly created model attempts to correct the errors from the preview model. Models are added to the ensemble until errors are completely eliminated on training set or the maximum number of models is reached. In section this subsection, we describe a technique works similarly to boosting algorithm, to build ensemble of generators that combined together, could generate data with an equal representation rate of different groups. During the training process, different generators can be created base on the previous ones and added to the ensemble. More precisely, while creating new generator G_i based on the set of the already existing generators $\mathbf{G} = \{G_1, G_2, ..., G_{i-1}\}$, one should consider the set of corresponding samples generated by each existing generator G_i. i.e $\mathbf{Q} = \{Q_1, Q_2, ..., Q_{i-1}\}$, where Q_i is the set of samples generated by the generator G_i i.e. $Q_i = \{x_1, x_2, ..., x_m\}$, $\forall j \in \{1, ..., i-1\}$; where m is a hyper-parameter.

We can then add a regularization term to the loss function of generators, that will enforces the distance between the newly and previously generated samples. For a given set of fake samples $\mathbf{F} = \{f_1, f_2, ..., f_m\}$, sampled by the generator at the current batch, the regularization term Θ can define as following:

$$\Theta = \frac{\Phi(\mathbf{Q})}{\sum_{Q_i \in \mathbf{Q}} \sum_{x \in Q_i} \sum_{f \in \mathbf{F}} \| x - f \|} \quad (3)$$

Where Φ is a given function that depends on $\mathbf{Q}$ used as normalization term.

Thus the new loss of a generator G_i is define as follow:

$$\mathcal{L}^{(G_i)} = -\mathbb{E}_z \log\left(1 - \mathrm{D}(\mathrm{G}(z))\right) + \Theta_i \quad (4)$$

During the testing phase, the objective is to generate samples from all existing groups (and possibly subgroups). To do so, we should not use generators, whose generation space is already covered by already generated samples ($\mathbf{g} = \{g_1, g_2, ..., g_p\}$), but rather generators that can generate samples different from those already generated. So, given a probability density functions $\{P_1, P_2, ..., P_n\}$ for all the generators $\{G_1, G_2, ..., G_n\}$ and probability $\mathbf{P}$ for the generator to be chosen, we have $\mathbf{P}(G_i) < \mathbf{P}(G_j)$ iff $\sum_{g \in \mathbf{g}} P_i(g) > \sum_{g \in \mathbf{g}} P_j(g)$.

Although we can not be sure about the form of the probability density function for the different generators, we may use a Normal distribution as an estimator for the function. In this case, $P_i = \mathcal{N}(E[X_i], \sqrt{Var[X_i]})$.

More specifically, probability for some generator G_i to be chosen is given by the following equation:

$$\mathbf{P}(G_i) = \frac{\prod_{f \in \mathbf{f}} e^{1 - P_i(f)}}{\sum_{P \in \mathbf{P}} \prod_{f \in \mathbf{f}} e^{1 - P(f)}} \quad (5)$$

In other word, $\mathbf{P}(G_i)$ is defined as the softmax over probability that there is no point at f.

VI. RELATED WORK

The existing works, which are similar to ours, are those that seek to solve the GAN mode collapse problem in GANs, by improving the generalization performance of the model,

in order to generate diverse samples. For instance, Qi et. al [31] proposed a regularization method for mode seeking to address the mode collapse problem in CGANs. However, diversity of generated samples is different from the goal of fairly generating data from different groups. In other words, a diverse set of samples may also contain underrepresented groups samples.

Xu et. al [32] proposed FairGAN, a GAN architecture to generate "fair" data with respect to the statistic parity, i.e remove all the information about sensitive attribute from generated data, while retaining as much information as possible. In their architecture, they have included another discriminator, that distinguish whether the generated samples are from protected group or unprotected group. Although their work also focuses on fairness in GANs, the goal is different from ours in that we are trying to alleviate the problem of representational bias, rather than achieving a certain notion of fairness. However, in future work, we will conduct comprehensive empirically comparisons of our boosting approach and FairGAN approach in terms of accuracy-fairness tradeoff.

CONCLUSION

In this paper, we highlighted fairness concern of GANs model. We have shown that when different groups are present in the dataset, the GAN model is not able to generate them homogeneously during the test phase, which can lead to representation bias. We discussed existing applications of GANs and argued that techniques using GANs to generate data for downstream tasks should consider this issue in order to avoid generating biased data, inherently introduced by GANs. We proposed a boosting strategy that can help to solve this issue. This strategy will be further explored in our future work and will be compared to other approaches in terms of fairness and accuracy.

REFERENCES

[1] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[3] A. R. Rivera, A. Khan, I. E. I. Bekkouch, and T. S. Sheikh, "Anomaly detection based on zero-shot outlier synthesis and hierarchical feature distillation," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[4] Y. Gavrilin and A. Khan, "Across-sensor feature learning for energy-efficient activity recognition on mobile devices," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–7.

[5] K. Sozykin, S. Protasov, A. Khan, R. Hussain, and J. Lee, "Multi-label class-imbalanced action recognition in hockey videos via 3d convolutional neural networks," in *2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*. IEEE, 2018, pp. 146–151.

[6] A. M. Khan, Y.-K. Lee, S. Lee, and T.-S. Kim, "Accelerometer's position independent physical activity recognition system for long-term activity monitoring in the elderly," *Medical & biological engineering & computing*, vol. 48, no. 12, pp. 1271–1279, 2010.

[7] S. Protasov, A. M. Khan, K. Sozykin, and M. Ahmad, "Using deep features for video scene detection and annotation," *Signal, Image and Video Processing*, vol. 12, no. 5, pp. 991–999, 2018.

[8] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[9] A. Khan and K. Fraz, "Post-training iterative hierarchical data augmentation for deep networks," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

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

[11] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *International conference on machine learning*. PMLR, 2014, pp. 1278–1286.

[12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

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

[14] J. Guo, S. Lu, H. Cai, W. Zhang, Y. Yu, and J. Wang, "Long text generation via adversarial training with leaked information," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[15] O. Mogren, "C-rnn-gan: Continuous recurrent neural networks with adversarial training," *arXiv preprint arXiv:1611.09904*, 2016.

[16] C. Esteban, S. L. Hyland, and G. Rätsch, "Real-valued (medical) time series generation with recurrent conditional gans," *arXiv preprint arXiv:1706.02633*, 2017.

[17] D. Madras, E. Creager, T. Pitassi, and R. Zemel, "Learning adversarially fair and transferable representations," in *International Conference on Machine Learning*. PMLR, 2018, pp. 3384–3393.

[18] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *Proceedings of the 3rd innovations in theoretical computer science conference*, 2012, pp. 214–226.

[19] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *arXiv preprint arXiv:1908.09635*, 2019.

[20] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.

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

[22] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[23] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *International conference on machine learning*. PMLR, 2017, pp. 2642–2651.

[24] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie, "Stacked generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5077–5086.

[25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[26] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.

[27] F. H. K. d. S. Tanaka and C. Aranha, "Data augmentation using gans," *arXiv preprint arXiv:1904.09135*, 2019.

[28] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi, "Bagan: Data augmentation with balancing gan," *arXiv preprint arXiv:1803.09655*, 2018.

[29] A. Chatziagapi, G. Paraskevopoulos, D. Sgouropoulos, G. Pantazopoulos, M. Nikandrou, T. Giannakopoulos, A. Katsamanis, A. Potamianos, and S. Narayanan, "Data augmentation using gans for speech emotion recognition." in *Interspeech*, 2019, pp. 171–175.

[30] R. Feng, Y. Yang, Y. Lyu, C. Tan, Y. Sun, and C. Wang, "Learning fair representations via an adversarial framework," *arXiv preprint arXiv:1904.13341*, 2019.

[31] Q. Mao, H.-Y. Lee, H.-Y. Tseng, S. Ma, and M.-H. Yang, "Mode seeking generative adversarial networks for diverse image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1429–1437.

[32] 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)*. IEEE, 2018, pp. 570–575.