



Fairness in generative modeling

Mariia Zameshina, Olivier Teytaud, Fabien Teytaud, Vlad Hosu, Nathanael Carraz, Laurent Najman, Markus Wagner

► To cite this version:

Mariia Zameshina, Olivier Teytaud, Fabien Teytaud, Vlad Hosu, Nathanael Carraz, et al.. Fairness in generative modeling. GECCO '22: Genetic and Evolutionary Computation Conference, Jul 2022, Boston Massachusetts, France. pp.320-323, 10.1145/3520304.3528992 . hal-03793686

HAL Id: hal-03793686

<https://hal.science/hal-03793686>

Submitted on 5 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Fairness in Generative Modeling: do it Unsupervised!

M. Zameshina^{1,2}, O. Teytaud²

1. LIGM, Univ Gustave Eiffel, CNRS,
ESIEE, Paris

2. Facebook AI Research
France

{mzameshina, oteytaud}@fb.com

Nathanael Carraz

Univ. d'Antananarivo

Madagascar

carraznathanael@live.fr

Fabien Teytaud

Univ. du Littoral Cote d'Opale
France

teytaud@univ-littoral.fr

Laurent Najman

LIGM, Univ Gustave Eiffel, CNRS,

ESIEE Paris

France

laurent.najman@esiee.fr

Vlad Hosu

Univ. of Konstanz
Germany

vlad.hosu@uni-konstanz.de

Markus Wagner

The University of Adelaide

Australia

markus.wagner@adelaide.edu.au

ABSTRACT

We design general-purpose algorithms for addressing fairness issues and mode collapse in generative modeling. More precisely, to design fair algorithms for as many sensitive variables as possible, including variables we might not be aware of, we assume no prior knowledge of sensitive variables: our algorithms use unsupervised fairness only, meaning no information related to the sensitive variables is used for our fairness-improving methods. All images of faces (even generated ones) have been removed to mitigate legal risks.

KEYWORDS

Generative modeling, neural networks, fairness

ACM Reference Format:

M. Zameshina^{1,2}, O. Teytaud², Fabien Teytaud, Vlad Hosu, Nathanael Carraz, Laurent Najman, and Markus Wagner. 2022. Fairness in Generative Modeling: do it Unsupervised!. In *Proceedings of The Genetic and Evolutionary Computation Conference 2022 (GECCO '22)*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3520304.3528992>

1 INTRODUCTION

Fairness has become prevalent at the intersection of ethics and artificial intelligence. Various forms of fairness are critical in online media [6]. In the present paper, we consider fairness in the context of generative modeling. More precisely, when modeling the probability distribution of faces, we typically observe that classes already rare in the dataset become even rarer in the model. This phenomenon is called Mode Collapse (MC) [20], and for sensitive variables, it is one of the fairness issues. We propose tools based on statistical reweighting (Sections 3.1 and 3.2) or on user feedback (Section 3.3) for mitigating fairness issues (such as MC) in generative modeling.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '22, July 9–13, 2022, Boston, USA

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-392-686...\$15.00

<https://doi.org/10.1145/3520304.3528992>

1.1 Fairness

There are many facets to fairness. An algorithm may be considered to be fair if its results are independent of some variables, particularly for sensitive variables. Fairness [18] can be measured in terms of separation, i.e., whether the probability of a given prediction, given the actual value, is the same for all values of a sensitive variable. The measurement can also be rephrased in terms of equivalent false negative and true negative rates for all classes. A distinct point of view is sufficiency: sufficiency holds if the probability of actually belonging to a given group is the same for individuals from that group and with different sensitive variables. Another point of view is independence, i.e., when the prediction is statistically independent of sensitive variables. Because it is known that the many criteria for fairness are contradictory, it is necessary to design criteria depending on the application. In the present paper, we consider the case in which the goal is to preserve some frequencies.

Here, we consider the context of generative modeling. There is a model trained on data, and we want this model to satisfy some requirements on frequencies: for every class, we would like the frequency to match some target frequency. Typically, for simplicity in the present paper, the target frequency is the frequency in the original dataset: however, the methods that we propose can be adapted to other settings.

1.2 Generative modeling: fairness and mode collapse

There are many measures of fairness, even in the specific case of generative modeling [29]. The main criterion is whether all classes are correctly represented. It is known that modeling frequently decreases the frequency of rare classes (i.e., mode collapse). In addition, improving the image quality (for each image independently) aggravates the diversity loss [24]. For a conditional generative model, there is sometimes a ground truth. For example, in super-resolution, we want the reconstructed image to match the sensitive variables of the ground truth as closely as possible. This case became particularly critical since, e.g., [30]: a pixelized version of Barack Obama can be “depixelized” to be that of a white man. [31] points out the importance of fairness in the design of Generative Adversarial Networks (GANs) before applying them, for example as an early stage before supervised training. For addressing fairness issues, a

possibility is to increase editability: [9] disentangles latent variables for separating editable and sensitive parts. Some works focus on measuring fairness, for example, [14] uses causal methodologies for measuring fairness in a counterfactual manner. Fairness can be integrated directly into the training: [26] focuses on training a GAN while protecting some variables.

1.3 Related work

[3] increases fairness in GANs in a supervised manner, i.e., given the sensitive attributes. [27] targets and improves the fairness of generated datasets. More similar to our work, [10] focuses on uncertain sensitive variables, and [13] adds a bias in a GAN for mitigating fairness issues. In the same fashion as the present work, [28] considers biasing a GAN without any retraining. We focus on generically (i.e., independently of the application, data, and model) correcting for potential bias present in a generative model, *without knowing the sensitive variables*. The critical point is that sensitive variables seem to often come up as a surprise: typically, people do not decide to create an unfair algorithm actively. For example, in [19], the designers of the faulty soap dispenser had just not imagined that it might fail on black skins. Also, there may be relevant sensitive variables that have not been initially considered: ethnicity or gender are obvious sensitive variables, but aesthetics, body mass index, social origin, or even the quality of the camera, geographical origin, also matter.

Our goal is to have a generic correction independent of the sensitive variables. The first proposed method (Sections 3.1 and 3.2):

- is not only for the fairness issues regarding sensitive variables: we also preserve diversity for more classical diversity issues such as MC.
- does not need any retraining.
- is more or less effective depending on cases but is designed for (almost) never being detrimental (Section 4.2).

The second proposed method, which can be combined with the previous one, proposes several generations and then lets the user choose. Therefore, the user experience is modified: we expect the user to assist the method by actively selecting relevant outputs. Contrary to the generic method proposed above, which we will implement thanks to reweighting, the new approach is not a drop-in replacement. Moreover, this also does not need retraining.

1.4 Outline

Section 2 presents tools useful for the present work:

- Use of Image Quality Assessment (IQA) to improve image generation (Section 2.1): we connect this method to our research by investigating how much this quality improvement degrades fairness and how our proposed methods can mitigate such issues.
- Reweighting via simple rejection sampling to improve fairness and reduce MC when the variables used for computing the reweighting values are correlated to the target sensitive variables (Section 3.1).

Section 3 presents our proposed algorithms:

- Reweighting as above, but with reweighed variables unrelated to target classes (Section 3.2). This second context is

Class	A	B	C	D
Frequency	17.8%	52.2%	17.5%	12.4%
Rank-correlation AvA	-0.07	0.22	-0.11	0.06
Rank-correlation K512	-0.02	0.16	-0.08	0.02

Table 1: For four distinct classes of individuals A, B, C and D (obtained using R), we present the rank-correlation of the frequency of that class with AvA and K512 scores respectively. AvA and K512 are visual quality estimators, dealing with aesthetics and technical quality respectively. Visual quality assessment is a task fairly independent of semantics and therefore should exhibit little if any ethnicity-related biases. Dataset: faces generated by StyleGan2 (see thispersondoesnotexist.com). Classes: ethnicity evaluated by R (see R in Table 2). Observation: the biggest class has the strongest, positive correlation.

therefore applicable when we do not know the target classes. We propose a method which is a drop-in improvement of an arbitrary generative model: as soon as we have features and a generative model, we can apply Alg. 1.

- Multi-objective optimization, through computation of several solutions (typically Pareto fronts), to mitigate diversity loss by providing more frequently at least one output of the category desired/expected by the user.

Section 4 is a mathematical analysis. Section 5 presents experimental results.

2 PRELIMINARIES

2.1 Correlations image quality / sensitive variables

We investigate the known correlation between the estimated quality of an image and its membership to a frequent class [15, 24].

In order to demonstrate that this is easily observable, Table 1 presents the rank correlation between the aesthetic quality of an image and the logit of that image for each of four classes of individuals. We note that the most positively correlated class is the most frequent. Our interpretation is that the technical quality of generated images is higher for the most frequent classes, influencing the aesthetics score.

2.2 Image generation: GAN, PGAN, and EvolGan

Our work specializes in image generation, and in particular on faces. We use the following image generation tools. Our baseline GAN is Pytorch GAN Zoo ([21], based on progressive GANs (PGANs) [11]). We also use EvolGan [23], which improves Pytorch GAN Zoo by biasing the random choice of latent variables z using K512 [8]. We use three configurations of EvolGan, as it uses as a budget the number of calls to the original GAN; the three configurations then correspond to budgets 10, 20, and 40 (named *EG10*, *EG20*, and *EG40* respectively). Besides the one based on a random search, EvolGan has an option for CMA search [5] and PortfolioDiscrete-(1 + 1) (i.e. the variant of the Discrete (1 + 1)-ES as in [4]): we also employ these variants, with notation respectively *EG-CMA-10* and *EG-D(1 + 1)-10* for budget 10, and similar variants for budget 20 and

Name	Notation	Domain	Note
Variables to be protected			
R	R	$\{A, B, C, D\}$	Ethnicity [1]
AvA	AvA	$\{F, E\}$	Aesthetics[7]
Related auxiliary variables			
Koncept512	R'	\mathbb{R}^4	Logits of R
	$K512$	\mathbb{R}	IQA
Unrelated auxiliary variables			
Emotions	E	$\{1, 2, 3, 4, 5, 6, 7\}$	facial expression in [1]
	E'	\mathbb{R}^{100}	final layer of E
VGG-Face final layer	VF	$\{0, 1\}^{128}$	Binarized VGG-face

Table 2: Feature extractors used in the present article. All data are faces, typically generated by StyleGAN2 or other methods in Section 2.2.

40. Therefore we have nine flavors of EvolGan, corresponding to different algorithms and budgets.

2.3 Diversity loss in generative modeling

Usually, modeling decreases the frequency of rare classes. With StyleGan2, we get 71.55% white people and 4.64% black according to R (close to [24]). EvolGan, which is built on top of StyleGan2 with a budget of 40 decreases the percentage of black people to 0% while increasing the frequency of white to 81.25%.

2.4 Measuring the diversity loss

We assume that there exist target frequencies for each sensitive class. In the present paper, we focus on preserving the diversity in the sense of “having the same frequencies as the frequencies in the original data used for creating the model”, so the target frequencies are the frequencies in the original dataset. If we consider the diversity loss associated with optimizing a model, such as EvolGan, we assume that target frequencies are those of the original model.

Given classes $\{1, \dots, n\}$ with target frequencies f_i ($\sum_{i=1}^n f_i = 1$), and real frequencies f'_1, \dots, f'_n : the diversity loss Δ is defined as $\Delta := 1 - \inf_{f'_i > 0} f'_i / f_i$. $\Delta = 0$ if the target frequencies are reached, and $\Delta = 1$ if one of the classes has disappeared. Throughout our paper, we consider diversity loss for classes, and not inside each class: this other important case is left as further work.

2.5 Feature extractors

We use various feature extractors (Table 2). E and R use VGG-Face [17]. The goal of these feature extractors is to have auxiliary classes for reweighting: these values, after discretization, provide classes. These classes, termed strata, are used in Section 3.2.

3 METHODS

Section 3.1 presents a simple rejection method for ensuring target probabilities in generative modeling. Section 3.2 shows how to build classes in order to apply that method without knowing what the sensitive variables are. Section 3.3 then presents a methodology based on multi-objective optimization for improving fairness.

3.1 Reweighting: stratified rejection

Consider a generative model on some domain D . Consider a partition D_1, \dots, D_m of D into m disjoint strata. Assume that some unknown random variable ω has probability $p_i = P(\omega \in D_i)$ and

$\sum p_i = 1$. We have another random variable g also living with probability one in the union of the D_i . Assuming that $P(g \in D_i) = p'_i$, a simple tool for building g' such that $P(g' \in D_i) = p_i$ is rejection (see Alg. 1). This simple algorithm generates $g' \in D_i$ with probability p_i .

Algorithm 1 Given a generative model g , bins D_1, \dots, D_m and their target probabilities p_1, \dots, p_m . This algorithm assumes that none of the D_i has probability 0 for the original generative model g .

```

Generate  $x$  a (new, independent) output of  $g$  // random gen
Find  $i$  such that  $x \in D_i$ .
With probability  $1 - \frac{1}{\max_j \frac{p_j}{p'_j}} \frac{p_i}{p'_i}$ , go back to random gen.

return  $g' := x$ .
```

3.2 Creating strata: reweighting without knowing the target classes

We have classes corresponding to sensitive classes. We consider four sensitive classes of faces (A, B, C, D) using R [1] and two classes using AvA [7] (class F = bottom 20% of the aesthetics variable). However, we also want (possibly non-sensitive) classes used as auxiliary classes for reweighting: our goal is for our method to work for unknown target classes, so we need auxiliary classes. The idea is to investigate how much we can improve fairness for variables A, B, C, D without using those classes in our algorithm. Our auxiliary classes (Section 2.5), unrelated to our sensitive classes, will be called strata in the present work: the strata are the D_i used in our reweighting algorithms.

The key point in our experiments “preserving the diversity of unknown target variables” is that we do not use the target variables in our algorithms: our method is unsupervised in this sense. When we try to maintain diversity for class F, we can use auxiliary variables which are unrelated to F: so, we can use A, B, C and D. And when we try to maintain diversity for classes A, B, C and D, we can use F as an auxiliary variable.

Some attributes (final layer of an emotion classifier, or technical quality of the photo) can be used for all classes as they are not directly related to any of our sensitive variables. We will use two parameters d and M in our experiments. Given a possibly large number of auxiliary variables (not the target variables), we select d variables. Each of these d variables is discretized in M values, where M is called the arity: thresholds are chosen so that the M values are equally frequent.

3.3 The user-assisted context: generating multiple solutions

Whereas in Section 3.2 we have considered a drop-in replacement of the baseline, which generates one image per instance, we now consider the case in which we generated several instances, and the user can select one of them (see Alg. 2). There are two parts: how to generate multiple contexts, and, for some methods which generate way too many solutions for being manually searched by a human user, how to sample the obtained Pareto front.

Algorithm 2 Different contexts for image generation, without or with human assistance. Left: unassisted context, generative model. Middle: generative model with target class (case in which there is an expected class, e.g., super-resolution in which the ethnicity is supposed to be preserved statistically). Right: user-assisted method. Not all unsupervised fairness methods can be applied in all cases. The reweighting method in Sections 3.1 and 3.2 can be applied to the two first columns. In contrast, the multiple generation such as the one in Section 3.3 can be applied to the third column only.

No context, no user assistance	Context, no user assistance	Context, user assistance
Repeatedly, generate one individual per request. Check that their frequencies match the expectation: compute a DL.	Repeatedly, generate one individual per request. Requests have a context (e.g., low-resolution image). Check that their frequencies match the frequencies of the context (e.g., same ethnicity as low-res image): compute a DL.	Repeatedly, generate k individuals per request (e.g., by Pareto-based MOO, or by diversity-based MOO, or by MSR): the user chooses one of them. Check that their frequencies match the contextual expectations: compute DL.

3.3.1 How to generate multiple solutions. We consider a fixed limit on the number of generated images allowed so that the tool remains manageable for the user. Several approaches can generate a targeted number of outputs; we consider (i) multi-objective optimization (MOO: splitting the original criterion into several and optimizing them jointly) and (ii) multiple runs. Doing multiple runs is a simple and intuitive solution for generating multiple images. Regarding MOO, our solution is not compatible with all generative models: we consider that images are obtained by numerical optimization of a linear combination of criteria [22]. Instead of aggregating them, [2] proposed to preserve diversity by optimizing several numerical criteria by MOO, and we include this technique (as well as the previously mentioned reweighting techniques) in our fairness context. MOO naturally generates several solutions instead of one so that we are (presumably) more likely to have at least one satisfactory solution.

3.3.2 How to sample the obtained solutions. When we do multiple runs, we can choose their number to control the number of generated images. However, in MOO, we typically get a Pareto front. This Pareto front might be huge. Therefore, we have to sample this Pareto front. There are many tools for this:

- Optimizing this sampling for some representativeness criterion in the fitness space (hypervolume and others, see Appendix A).
- Or maximizing some diversity criterion in the original domain, regardless of fitness values.

4 METHODS ANALYSIS

4.1 Multi-objective diversification

Generating several solutions and letting the user choose among those proposals is a simple workaround for partially mitigating diversity loss.

However, not all methods are equal: we would like to have as much diversity as possible for a given fixed number of proposals. Also, Fig. 1 shows that it is not obvious that this will work: though this might not be intuitive, one can design counter-examples in which focusing on the Pareto-front and even more on a few key elements representing the Pareto front can actually decrease the diversity, compared to generating just one image at a time, because

the Pareto frontier might be entirely covered by a single class (in particular the biggest class, for which values are usually greater in machine learning models, as explained in Section 2.1). The simplest, and maybe most robust solution is to run multiple independent (randomized) runs: if the probability $P(g \in C)$ of generating a point in C is low, then the probability $1 - (1 - P(g \in C))^k$ of having at least one of k generated image inside C is greater: $1 - (1 - P(g \in C))^k \geq P(g \in C)$ (strict if $P(g \in C) \notin \{0, 1\}$). If the user needs an image of class C , generating k images is more likely to have at least one in C unless the original probability is 0 or 1.

The question is now how to do better than this baseline. We consider the following ideas:

- the k runs are not using the same weights: e.g., we use random weights in the optimization runs, and they are randomly drawn at each run.
- we run a MOO algorithm which tries to maximize some quantity, e.g., the hypervolume of the obtained solutions, or their diversity in the loss space, or the coverage in the domain space.

Consistent with the credo of the present paper (not using target classes in the algorithm), these algorithms are independent of the target classes.

4.2 Stratification by rejection is rarely detrimental

The reweighting method in Section 3.1 works in the sense that, by design, when we use it, we switch back to the exact probabilities for each stratum, i.e., $p' = p$. This implies that, unless a target class has entirely disappeared in the model, reweighting using strata based on the target classes recovers the frequencies of all target classes. However, the point of the present paper is to fix frequencies of unknown target classes. So, now, consider a target class C , which is not necessarily one of the strata. If C is one of the D_i (or a union of them) then, as discussed above, the stratification leads to $p(g \in C) = p(\omega \in C)$: let us see if we can find a more general case in which $P(g \in C) = P(\omega \in C)$.

The Diversity Loss (DL) measure we are using (Section 2.4) for estimating the DL of a model g compared to a random variable w is based on aggregating measures of DL for several classes: the

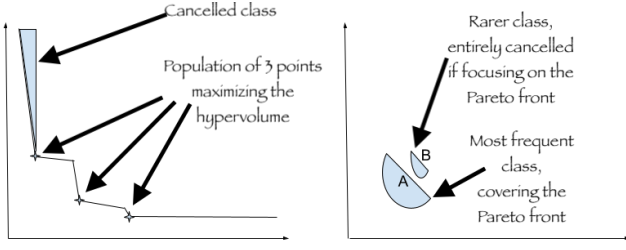


Figure 1: Bi-objective minimization, cases in which Pareto-dominance will be detrimental to diversity. Left: artificial counter-example showing that maximizing a numerical diversity criterion (the hypervolume) over the Pareto front might not provide diverse solutions. Here, we see a Pareto-front and the hypervolume-best approximation by 3 points. Dots: the 3 individuals maximizing the hypervolume. Gray areas: examples of classes that completely disappear if we consider those dots (as they maximize the hypervolume) rather than a random sampling of the Pareto front. Right: other counter-example. Class A is assumed to be much bigger than class B, and to have, therefore, better scores for both criteria: this is because, as discussed in the text, bigger classes typically have better scores (see Section 1.2). While local optimization from points in B will provide points in B, a global optimization based on Pareto fronts will provide only points in A: class B is not represented.

global diversity loss is $\Delta := 1 - \inf_{f_i > 0} f'_i / f_i$ where f_i is the target frequency for class i and f'_i is the observed frequency.

$$\begin{aligned} \Delta &= \max_{C; P(w \in C) > 0} \left(1 - \frac{P(g \in C)}{P(w \in C)} \right) \\ &= \max_{C; P(w \in C) > 0} \frac{1}{P(w \in C)} (P(w \in C) - P(g \in C)) \\ &= \max_{C; P(w \in C) > 0} \frac{1}{P(w \in C)} (pq - pq') \end{aligned}$$

where:

- q_j is the probability of class C in stratum D_j for the original random variable w i.e. $q_j = P(w \in C | w \in D_j)$;
- q'_j is the counterpart for the model g i.e. $q'_j = P(g \in C | g \in D_j)$.

The reweighting increases the DL for class C if $pq - pq' > pq - p'q'$ (where pq is short for $\sum_j p_j q_j$). This is equivalent to $q'(p' - p) > 0$ and $p(q - q') > 0$. This means that reweighting is detrimental for this measure if (i) $p(q - q') > 0$ and (ii) $q'(p' - p) > 0$ occur simultaneously: (i) means that $q - q'$ is overall positive on average for the frequencies p (i.e., g tends to underestimate class C), which is precisely the case of interest: this means that g is not doing well on C . And (ii) $q'(p' - p) > 0$: this implies that we tend to overestimate classes in which C has a low probability, which contradicts the general assumption “diversity loss usually occurs for rarer classes” in Section 2.3. Therefore, it seems unlikely that reweighting can worsen diversity loss, at least for this measure.

Baseline	Model	M	Diversity loss before reweight	Percentage of DL remaining with $d = 2$	Percentage of DL remaining with $d = 4$
PGAN	EG-CMA-10	3	0.442	53.266	42.257
PGAN	EG-CMA-20	3	0.513	49.901	32.176
PGAN	EG-CMA-40	3	0.663	83.654	40.683
PGAN	EG-D(1+1)-10	3	0.080	74.254	16.168
PGAN	EG-D(1+1)-20	3	0.070	72.913	30.008
PGAN	EG-D(1+1)-40	3	0.115	25.079	33.147
dataset	EG-RandomSearch-0	3	0.314	31.699	25.398
dataset	EG-RandomSearch-10	3	0.563	28.083	33.860
dataset	EG-RandomSearch-20	3	0.644	33.280	40.709
dataset	EG-RandomSearch-40	3	0.738	65.564	63.747
dataset	EG-CMA-0	3	0.343	40.314	16.914
dataset	EG-CMA-10	3	0.561	32.505	6.927
dataset	EG-CMA-20	3	0.617	27.205	29.584
dataset	EG-CMA-40	3	0.735	47.673	33.604
dataset	EG-D(1+1)-0	3	0.312	40.628	10.630
dataset	EG-D(1+1)-10	3	0.339	32.440	28.977
dataset	EG-D(1+1)-20	3	0.347	95.618	11.370
dataset	EG-D(1+1)-40	3	0.350	32.822	16.938

Table 3: Impact of reweighting with related variables on the diversity loss for classes A, B, C, D: we see that the original diversity loss is significant (4th column) and reduced a lot if we use 4 variables for reweighting (6th column). Even 2 variables contribute quite well to a significant reduction of diversity loss (5th column). Dataset: faces generated by StyleGAN2. Strata used for reweighting: logits of the output layer of R discretized with $M = 3$ and $d = 2$ (5th column) or $d = 4$ (6th column).

5 EXPERIMENTAL RESULTS

5.1 Framework

We compare our methods in different contexts. Each context (g, b) is defined by a generative model g to be compared to a baseline b (dataset or model). We check if g has a diversity loss, comparatively to b . We have 18 contexts, as described below. The baseline b is a dataset or a PGAN [12] trained on it (i.e., two possibilities here), and we try to fix the diversity loss when applying EvolGAN [23] with budget 10, 20, 40 (3 possibilities) and algorithm DOPO [22], CMA [5] or random search (3 possibilities): g can be any of these 9 combinations, and we consider the diversity loss compared to one of the two different possible b , hence 18 contexts (Table 9). Different contexts have different diversity losses: typically, CMA or RandomSearch lead to more diversity loss than DOPO.

We have checked that (naively) optimizing technical quality is detrimental to fairness (Appendix B). We show (Section 5.2) that applying reweighting according to target classes is unsurprisingly more effective than reweighting according to unrelated strata, but the latter methodology still does mitigate fairness issues. Then Section 5.3 compares various forms of user-assisted optimization for tackling fairness issues.

5.2 Reweighting mitigates fairness issues

5.2.1 Classes A, B, C, D. Table 3 presents the diversity loss and the fixed diversity loss when using reweighting. We use 2 or 4 variables correlated (though not equal) to the target attribute, namely the discretized predicted probabilities of the 4 modalities of the target class. As variables are correlated to the target problem, results are excellent.

Number of vars d	Discretization M	DL before reweighting	DL after reweighting
1	2		0.421
1	3		0.428
1	5	0.431	0.435
1	8		0.430
2	3		0.403
2	5	0.431	0.431
2	8		0.433
4	2		0.414
4	3		0.403
4	5	0.431	0.428
4	8		0.427
10	3		0.395
10	5	0.431	0.423
20	2		0.419
20	3		0.401
20	5	0.431	0.432
20	8		0.428
80	2		0.419
80	8	0.431	0.428

Table 4: Diversity loss for (A, B, C, D) after reweighting, in our hardest context (variables very uncorrelated to the target variable, namely E’). We observe that in most cases, the reweighting is still beneficial compared to 0.431 originally, though this difficult case does not lead to drastic improvements. Dataset: faces generated by StyleGan2. Strata: discretization of E’ with $d \in \{1, 2, 4, 10, 20, 80\}$ and $M \in \{2, 3, 5, 8\}$.

We now switch to a more challenging case. Table 4 compares various discretizations in the difficult context of reweighting variables unrelated to the target variables. E.g. (80,8) means that we use $d = 80$ variables and split each of them in $M = 8$ bins. We got the best results with 10 variables discretized in 3. There are four target classes for faces unrelated to emotions. The variables are the final layer of an emotion recognition network. Still, in that difficult case, Fig. 2 shows how diversity losses are moved in the right direction by the reweighting – not much, but beneficial, and most importantly, not detrimental.

5.2.2 Class E: confirming results for reweighting with unrelated variables. Table 5 presents the impact of reweighting using the probabilities of class A, B, C and D (discretized) on the diversity loss of class E. (ABCD) and E are unrelated, so this is unsupervised fairness improvement.

5.3 Multi-objective optimization: only some forms of MOO mitigate fairness issues

MOO typically has two phases:

- optimization run, building a possibly large Pareto front;
- selection of a reduced Pareto front for presentation to the user.

This does not cover all MOO methods. The second stage is not always present, as some tools are equipped with a mechanism for navigating the Pareto front. Also, sometimes the first stage includes inputs from the human. We will nonetheless consider the framework above in the present paper. As mentioned before, a simple solution for MOO is to do multiple simple runs (MSR): just run the algorithm several times, and consider the several outputs. We consider other methods, namely maximizing the hypervolume for phase 1 and using various techniques (IGD, EPS, RANDOM, see Appendix A) for constructing a subset.

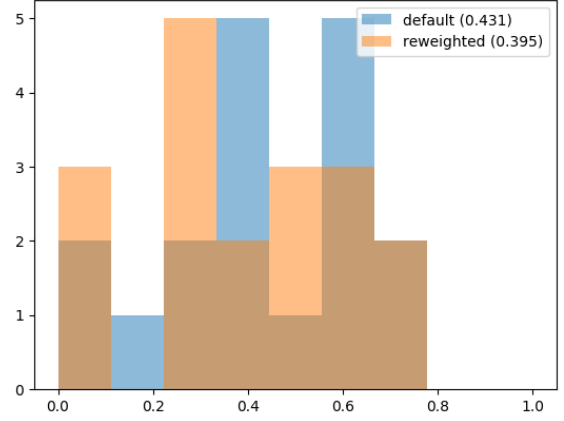


Figure 2: Hard case with unrelated reweighting variables: histogram of diversity losses for (A,B,C,D) using reweighting based on strata of R’, over each of 18 contexts (see text). The method is slightly beneficial; the average moves from 0.431 to 0.395. We use the best method in Table 4, rerun from scratch for mitigating the hyperparameter selection bias, getting the same 0.395 value. Dataset, strata, as in Table 4. X-axis: DL. Y-axis: number of contexts (out of 18) with DL falling in the given DL bin.

Tables 6 (target class is black) and 7 (target class is female Asian) show that the best results concerning maximum diversity are obtained by domain-covering or by MSR, and not by MOO approaches focusing on diversity over the Pareto front. The effective diversity measures are not based on Pareto-dominance. The best results are obtained either by pure MSR, using multiple runs and keeping all results, or by domain-covering, i.e., creating a subset using diversity in the image domain. This result is not so intuitive, so we ran additional experiments to check if Pareto-dominance can be detrimental to diversity. **We conclude that Pareto-based MOO can be detrimental to diversity even with a large budget and 16 generations instead of 1.** This is shown by Table 8: we do an additional experiment based on Pytorch-Gan-ZOO and variants. We use both single-objective optimization (EvolGan with budget 10000) and our MOO counterpart. We get a single image per run for single-objective optimization, and we can estimate DL as usual. We use MOO, with three objectives linearly combined in the single-objective case: minimizing the squared of the injected latent variables, maximizing the IQA score, and maximizing the discriminator score. We use a large budget and many generated individuals so that problems can not be attributed to the parametrization. We consider that the “frequency” of a class is the frequency at which at least one of the outputs contains that class (see Alg. 2). We see that MOO by classical Pareto-dominance is not always solving diversity issues. It works only when the method has over-optimized and completely destroyed diversity (Table 8: results are < 100% in the last column only if the diversity loss is > 95%). Whereas diversity in the domain (domain-covering) or simple multiplication of runs (as in MSR) works in many cases, optimization with Pareto-dominance can fail. We conclude that counter-examples as in Fig. 1 are not an exception but the standard behavior of Pareto-dominance: due to

Source	Target	d	M	DL	Remaining DL(%)
PGAN	EG-CMA 10	1	8	0.675	99.832
PGAN	EG-CMA 20	1	8	0.778	100.444
PGAN	EG-CMA 40	1	8	0.872	100.923
PGAN	EG-D(1+1) 10	1	8	0.108	103.808
PGAN	EG-D(1+1) 20	1	8	0.204	76.961
PGAN	EG-D(1+1) 40	1	8	0.333	92.000
PGAN	EG-RandomSearch 10	1	8	0.675	88.797
PGAN	EG-RandomSearch 20	1	8	0.785	100.171
PGAN	EG-RandomSearch 40	1	8	0.876	98.348
PGAN	EG-CMA-10	2	8	0.675	96.660
PGAN	EG-CMA-20	2	8	0.778	95.140
PGAN	EG-CMA-40	2	8	0.872	89.906
PGAN	EG-D(1+1)-10	2	8	0.108	103.808
PGAN	EG-D(1+1)-20	2	8	0.204	89.274
PGAN	EG-D(1+1)-40	2	8	0.333	98.751
PGAN	EG-RandomSearch-10	2	8	0.675	87.870
PGAN	EG-RandomSearch-20	2	8	0.785	91.072
PGAN	EG-RandomSearch-40	2	8	0.876	97.075
PGAN	EG-CMA-10	3	8	0.675	95.538
PGAN	EG-CMA-20	3	8	0.778	95.218
PGAN	EG-CMA-40	3	8	0.872	89.552
PGAN	EG-D(1+1)-10	3	8	0.108	90.259
PGAN	EG-D(1+1)-20	3	8	0.204	81.139
PGAN	EG-D(1+1)-40	3	8	0.333	84.623
PGAN	EG-RandomSearch-10	3	8	0.675	86.845
PGAN	EG-RandomSearch-20	3	8	0.785	87.502
PGAN	EG-RandomSearch-40	3	8	0.876	95.705
PGAN	EG-CMA-10	4	8	0.675	97.587
PGAN	EG-CMA-20	4	8	0.778	96.505
PGAN	EG-CMA-40	4	8	0.872	90.098
PGAN	EG-D(1+1)-10	4	8	0.108	70.592
PGAN	EG-D(1+1)-20	4	8	0.204	94.112
PGAN	EG-D(1+1)-40	4	8	0.333	88.999
PGAN	EG-RandomSearch-10	4	8	0.675	87.333
PGAN	EG-RandomSearch-20	4	8	0.785	88.270
PGAN	EG-RandomSearch-40	4	8	0.876	96.438

Table 5: Impact of reweighting on diversity loss for class E when using classes R as auxiliary variable. We see that adding variables almost always improves results, and cases in which reweighting is detrimental are rare. Dataset: faces generated by StyleGan2. Sensitive variables for which DL is computed: emotions. Strata: IQA values provided by R', i.e., logits of R, with discretization with $d \in \{1, 2, 3, 4\}$ variables and $M = 8$ equally likely bins per variable. Observation: increasing d reduces the DL after reweighting.

different scales of quality depending on the frequency of classes, we can not reliably use Pareto-dominance for selecting samples. MSR is the only method that did not have counter-examples. MOO methods based on Pareto fronts were ok only when the method for extracting representative images was based on domain-covering, i.e., unsupervised correction.

6 CONCLUSION

Quality improvement degrades diversity: We checked that improving the visual quality degrades diversity when biasing latent variables through IQA methods. The biasing effect is consistent with known facts.

To mitigate this issue, we propose two methods. The first (Alg. 1) is a drop-in improvement of a generative model: it can be applied as soon as we have some auxiliary features that we can use for defining strata. The second one is user-assisted (Alg. 2) and can use MOO (either with Pareto-dominance for selecting a subset or with diversity preservation for some features in the domain) or MSR.

Reweighting by related auxiliary variables: Unsurprisingly, reweighting by auxiliary variables close to the target classes is

Algorithm	Selector	Percentage
9 single-objective runs		
NGOpt 9	domain-covering	33
NGOpt 9	eps	33
NGOpt 9	loss-covering	33
NGOpt 9	msr	33
CMA		
CMA	domain-covering	33
CMA	eps	33
CMA	loss-covering	44
CMA	msr	66
Portfolio Discrete-(1 + 1)		
PortfolioDiscrete(1 + 1)	msr	16
PortfolioDiscrete(1 + 1)	eps	33
PortfolioDiscrete(1 + 1)	loss-covering	33
PortfolioDiscrete(1 + 1)	domain-covering	83
Differential Evolution		
DE	loss-covering	16
DE	eps	16
DE	domain-covering	33
DE	msr	55
Random Search		
RandomSearch	loss-covering	0
RandomSearch	msr	33
RandomSearch	eps	50
RandomSearch	domain-covering	66

Table 6: Multi-objective inspirational generation: the target is the face of a black person, originally very pixelized; the goal is to approximate it with PytorchGanZoo. We consider with which probability PytorchGanZoo generates at least one face of the correct ethnicity. Each algorithm generates nine faces. The best selector consists of picking up the nine outcomes of nine single runs (MSR: multiple single runs) or using domain covering, i.e., never using a Pareto-based measure. In conclusion, multi-objective optimization does work for generating diversity. However, we should not use Pareto-dominance and focus on multiple outcomes of random single-objective runs or diversity in the domain ("domain-covering" method), because fitness-based measures are too biased for being used for diversity.

Algorithm	Selector	Percentage
9 single-objective runs		
NGOpt 9	domain-covering	22
NGOpt 9	eps	0
NGOpt 9	loss-covering	5
NGOpt 9	msr	11
CMA		
CMA	domain-covering	27
CMA	eps	11
CMA	loss-covering	22
CMA	msr	0
Portfolio Discrete-(1 + 1)		
PortfolioDiscrete(1 + 1)	msr	0
PortfolioDiscrete(1 + 1)	eps	38
PortfolioDiscrete(1 + 1)	loss-covering	16
PortfolioDiscrete(1 + 1)	domain-covering	38
Differential Evolution		
DE	loss-covering	16
DE	eps	22
DE	domain-covering	5
DE	msr	11
Random Search		
RandomSearch	loss-covering	5
RandomSearch	msr	0
RandomSearch	eps	11
RandomSearch	domain-covering	33

Table 7: Counterpart of Table 6 for female Asian target. As in Table 6, domain-covering performs best.

Original model	EG40 variant	PF size	Subset	d, M	Diversity loss	Uncancelled loss (%)
PGAN	EG-RandomSearch	16	COV	5,2	0.726	111.333
PGAN	EG-CMA	16	COV	5,2	0.977	89.285
PGAN	EG-D(1+1)	16	COV	5,2	0.707	116.297
PGAN	EG-RandomSearch	16	IGD	5,2	0.72	112.165
PGAN	EG-CMA	16	IGD	5,2	0.973	90.008
PGAN	EG-D(1+1)	16	IGD	5,2	0.730	112.409
PGAN	EG-RandomSearch	16	Random	5,2	0.697	118.724
PGAN	EG-CMA	16	Random	5,2	0.969	89.622
PGAN	EG-D(1+1)	16	Random	5,2	0.726	115.778
PGAN	EG-RandomSearch	16	EPS	5,2	0.738	107.916
PGAN	EG-CMA	16	EPS	5,2	0.977	88.095
PGAN	EG-D(1+1)	16	EPS	5,2	0.709	116.377

Table 8: Column 6 shows the DL when moving from the original (column 1) to the improved version (column 2), and column 7 presents the part of this DL which is not solved by applying MOO for generating 16 points. There is a strong computational budget (10000) and a large generated set (16 points) in the present context. We consider that the result is ok if at least one of those 16 generations is of the expected class. Column 7 is frequently above 100%, i.e., results are worse than in the single-objective case generating only one image: this shows that even with favorable conditions, MOO based on Pareto-dominance can be detrimental. Only MSR (running several times and gathering the results) or domain-covering (i.e., good diversity for a side measure in the domain) provide stable improvements in the user-assisted context (Tables 6 & 7). Dataset: CelebA HQ (see https://github.com/tkarras/progressive_growing_of_gans). Model: PytorchGanZoo. Method: described in Section 3.3. Sensitive variables on which DL is measured: ethnicity.

very effective at reducing the diversity loss. We cancel the diversity loss when reweighting using the same target class. This incurs a computational cost and does not solve quality inside each class, but we recover target frequencies.

Reweighting by unrelated auxiliary variables: A good finding is that we never degrade performance by applying reweighting, even when using unrelated variables. There are good reasons for this (Section 4.2). We recommend reweighting by as many variables as possible (at least as long as there is data enough for computing statistics with enough precision). However, we acknowledge that this has a computational cost.

Using MOO, also without knowing categories: The idea of using MOO for generating diversity is intuitively appealing. However, only MSR (running several single objective problems) or domain-covering turned out to be effective. Methods based on Pareto-dominance can be detrimental. Phenomena, as described in Fig. 1, are not an exception, but the rule.

Side remarks & caveats

Combination with supervised fairness: we considered purely unsupervised fairness, but we could do the same in combination with given sensitive variables: after a first correction for given sensitive variables, we can add a correction with respect to some unrelated generic strata.

Impact of the optimization method: Tables 9, 3 and 8 show that CMA leads to more diversity loss compared to random search or

PortfolioDiscrete(1+1). This is reasonable as the prior distribution is ignored by CMA, whereas it impacts every other tested methods:

- Random search uses the prior distribution at each step for choosing a point;
- Discrete (1+1) algorithms use the marginal of the probability distribution for each modified variable.

We presented results for reweighting with statistics based on large datasets, so that there was no problem for precisely estimating p_i/p_j as needed: with small datasets, precision might be an issue.

APPENDIX

A SUBSAMPLING THE PARETO FRONT

To extract $1 \leq m \leq n$ points from an approximate Pareto set $\{x_1, \dots, x_n\}$, a range of approaches can be used:

- Random subset: just pick up m of the x_i , uniformly at random and without replacement.
- HV: pick up $\{x_{j_1}, \dots, x_{j_m}\}$ such that their Hypervolume C_h is maximal.
- Loss-covering, also known as IGD (inverted generational distance, [25]): pick up $\{x_{j_1}, \dots, x_{j_m}\}$ such that $C_l = \sum_{i=1}^n \inf_{j \leq m} \|F(x_i) - F(x_{j_j})\|^2$ is minimal, where $F(x) = (f_1(x), \dots, f_N(x))$.
- COV (covering the Pareto-front): pick up $\{x_{j_1}, \dots, x_{j_m}\}$ such that $C_d = \sum_{i=1}^n \inf_{j \leq m} \|x_i - x_{j_j}\|^2$ is minimal.
- Additive epsilon approximation (EPS, [16]): pick up $\{x_{j_1}, \dots, x_{j_m}\}$ such that $C_e = \max_{i=1}^n \inf_{j \leq m} \|F(x_i) - F(x_{j_j})\|_\infty$ is minimal, where $F(x) = (f_1(x), \dots, f_N(x))$.

In domain-covering, we do the same as COV, but over all generated points and not only the Pareto-front.

B (NAIVELY) OPTIMIZING \rightarrow LESS DIVERSITY

We train a PGAN [12] and then improve it using IQA as in [23]: $\text{PGAN} \rightarrow \text{EG10} \rightarrow \text{EG20} \rightarrow \text{EG40}$ (each “ \rightarrow ” being an improvement in terms of image quality by refining the latent variables using the image quality assessment tool as a criterion[23]). As noted in [23], the quality improvement in EvolGAN is related to some diversity losses: for horses, we get rid of bugs such as horses with 3 heads, which is in some sense a sort of diversity loss. Unfortunately, this also reduces diversity in the sense that relevant rare classes become rarer (Table 9): there is a diversity loss from the dataset to the PGAN, and this diversity loss is increased when we increase the budget of the GAN improvement by EvolGAN.

Further appendices can be found in the supplementary material.

EG variant	Diversity loss	Remaining diversity loss (%)
EG-CMA-10	0.675	97.587
EG-CMA-20	0.778	96.505
EG-CMA-40	0.872	90.098
EG-D(1+1)-10	0.108	70.592
EG-D(1+1)-20	0.204	94.112
EG-D(1+1)-40	0.333	88.999
EG-RandomSearch-10	0.675	87.333
EG-RandomSearch-20	0.785	88.270
EG-RandomSearch-40	0.876	96.438

Table 9: Diversity loss for class F (i.e., low aesthetics value according to AvA) for EG compared to PytorchGanZoo (EG is an improvement of PytorchGanZoo using K512 as an IQA for biasing the latent variables). The diversity loss depends on how strongly we improve the GAN using EvolGan (more budget = more improvement in terms of quality measured by K512). We also show (third column) how much the diversity loss is preserved in spite of reweighting w.r.t. E : numbers < 100% show that a part of the diversity loss is repaired. No number is greater than 100%: our method is never detrimental.

REFERENCES

- [1] H. Anadon. Face expression and ethnic recognition. <https://github.com/HectorAnadon/Face-expression-and-ethnic-recognition>, 2019.
- [2] N. Carraz Rakotonirina, A. Rasoanaivo, L. Najman, P. Kungurtsev, J. Rapin, F. Teytaud, B. Roziere, O. Teytaud, M. Wagner, P.-K. Wong, and V. Hosu. Many-Objective Optimization for Diverse Image Generation. working paper or preprint, Nov. 2021.
- [3] K. Choi, A. Grover, T. Singh, R. Shu, and S. Ermon. Fair generative modeling via weak supervision. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1887–1898. PMLR, 13–18 Jul 2020.
- [4] D. Dang and P. K. Lehre. Self-adaptation of mutation rates in non-elitist populations. In *Parallel Problem Solving from Nature - PPSN XIV - 14th International Conference, Edinburgh, UK, September 17-21, 2016, Proceedings*, pages 803–813, 2016.
- [5] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 11(1), 2003.
- [6] E. Hargreaves, C. Agosti, D. Menasché, G. Neglia, A. Reiffers-Masson, and E. Altman. Fairness in online social network timelines: Measurements, models and mechanism design. *Performance Evaluation*, 129:15–39, Feb 2019.
- [7] V. Hosu, B. Goldlucke, and D. Saupé. Effective aesthetics prediction with multi-level spatially pooled features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9375–9383, 2019.
- [8] V. Hosu, H. Lin, T. Sziranyi, and D. Saupé. KonIQ-10k: An ecologically valid database for deep learning of blind image quality assessment. *IEEE Transactions on Image Processing*, pages 1–1, 2020.
- [9] S. Hwang, S. Park, D. Kim, M. Do, and H. Byun. Fairfacegan: Fairness-aware facial image-to-image translation, 2020.
- [10] A. Jalal, S. Karmalkar, J. Hoffmann, A. Dimakis, and E. Price. Fairness for image generation with uncertain sensitive attributes. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4721–4732. PMLR, 18–24 Jul 2021.
- [11] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. *ICLR*, 2018.
- [12] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.
- [13] P. J. Kenfack, D. D. Arapov, R. Hussain, S. M. A. Kazmi, and A. M. Khan. On the fairness of generative adversarial networks (gans), 2021.
- [14] M. J. Kusner, J. Loftus, C. Russell, and R. Silva. Counterfactual fairness. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [15] S. Menon, A. Damian, S. Hu, N. Ravi, and C. Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models, 2020.
- [16] C. H. Papadimitriou and M. Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *41st Annual Symposium on Foundations of Computer Science*, pages 86–92, 2000.
- [17] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In X. Xie, M. W. Jones, and G. K. L. Tam, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 41.1–41.12. BMVA Press, September 2015.
- [18] D. Pessach and E. Shmueli. Algorithmic fairness, 2020.
- [19] M. Plenke. The reason this "racist soap dispenser" doesn't work on black skin. <https://www.mic.com/articles/124899/the-reason-this-racist-soap-dispenser-doesn-t-work-on-black-skin>, 2021.
- [20] E. Richardson and Y. Weiss. On GANs and GMMs, 2018.
- [21] M. Riviere. Pytorch GAN Zoo. https://GitHub.com/FacebookResearch/pytorch_GAN_zoo, 2019.
- [22] M. Riviere, O. Teytaud, J. Rapin, Y. LeCun, and C. Couprie. Inspirational adversarial image generation. *arXiv preprint 1906.11661*, 2019.
- [23] B. Rozière, F. Teytaud, V. Hosu, H. Lin, J. Rapin, M. Zameshina, and O. Teytaud. EvolGAN: Evolutionary Generative Adversarial Networks. In *Asia Conference on Computer Vision (ACCV)*, Virtual, Japan, Nov. 2020.
- [24] J. Salminen, S.-G. Jung, and B. J. Jansen. Detecting demographic bias in automatically generated personas. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI EA '19, page 1–6, New York, NY, USA, 2019. Association for Computing Machinery.
- [25] H. Sato, H. E. Aguirre, and K. Tanaka. Local dominance using polar coordinates to enhance multiobjective evolutionary algorithms. In *Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, volume 1, pages 188–195 Vol.1, 2004.
- [26] P. Sattigeri, S. C. Hoffman, V. Chenthamarakshan, and K. R. Varshney. Fairness gan, 2018.
- [27] P. Sattigeri, S. C. Hoffman, V. Chenthamarakshan, and K. R. Varshney. Fairness gan: Generating datasets with fairness properties using a generative adversarial network. *IBM Journal of Research and Development*, 63(4/5):3:1–3:9, 2019.
- [28] S. Tan, Y. Shen, and B. Zhou. Improving the fairness of deep generative models without retraining. *arXiv preprint arXiv:2012.04842*, 2020.
- [29] C. T. H. Teo and N.-M. Cheung. Measuring fairness in generative models, 2021.
- [30] K. Truong. This image of a white barack obama is ai's racial bias problem in a nutshell. *vice.com*, 2020.
- [31] 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.

SUPPLEMENTARY MATERIAL FOR: FAIRNESS IN GENERATIVE MODELING: DO IT UNSUPERVISED!

Disclaimer. We understand that neither the reviewers, the track chairs, nor the editors are required to look at the supplementary material. We also understand that they are requested to base their reviews and decisions solely on the main PDF manuscript.

(SUP1) Human raters

We use human raters for the two applications in Tables 6 and 7. Our raters are volunteers, without any time limit. A double-blind graphical user interface presents images. For labeling with ethnicity, we use a binary question.

(SUP2) Reweighting with respect to four VF binarized variables for a specific target

		StyleGan2	EG	Reweigh - EG
Selection rate in EG: 5.1%				
100/1979	corr.	.0480	.0297	.0258
100/1979	random	.0480	.0297	.0309
Selection rate in EG: 7.6%				
150/1979	corr.	.0480	.0271	.0219
150/1979	random	.0480	.0271	.0292
Selection rate in EG: 10.1%				
200/1979	corr.	.0480	.0247	.0207
200/1979	random	.0480	.0247	.0269
Selection rate in EG: 12.6%				
250/1979	corr.	.0480	.0273	.0279
250/1979	random	.0480	.0273	.0277
Selection rate in EG: 15.2%				
300/1979	corr.	.0480	.0239	.0245
300/1979	random	.0480	.0239	.0234
Selection rate in EG: 17.7%				
350/1979	corr.	.0480	.0285	.0333
350/1979	random	.0480	.0285	.0300
Selection rate in EG: 20.2%				
400/1979	corr.	.0480	.0353	.0350
400/1979	random	.0480	.0353	.0374
Selection rate in EG: 22.8%				
450/1979	corr.	.0480	.0358	.0370
450/1979	random	.0480	.0358	.0390
Selection rate in EG: 25.3%				
500/1979	corr.	.0480	.0344	.0354
500/1979	random	.0480	.0344	.0360
Selection rate in EG: 27.8%				
550/1979	corr.	.0480	.0344	.0364
550/1979	random	.0480	.0344	.0363
Selection rate in EG: 25.3%				
600/1979	corr.	.0480	.0331	.0340
600/1979	random	.0480	.0331	.0356

Table 10: Dataset: faces generated by StyleGan2. The frequency of black people in the different versions, depending on which strata are used for applying the reweighting method of Section 3.2. Random: four variables randomly picked up among the 128 binary variables built from VGG-Faces. Correlated: same VGG-Faces, but we use the most correlated ones.

Table 10 presents results of different methods in terms of the frequency of black people. In most cases, the frequency of black people decreased from the original 4.8% when applying EvolGan, but increased when applying reweighting. We note exceptions: whereas randomly chosen variables were always beneficial, very correlated variables failed in the most difficult cases.