Imposing Fairness Constraints in Synthetic Data Generation

Mahed Abroshan
The Alan Turing Institute
mahed.ab@gmail.com

Andrew Elliott
The University of Glasgow
The Alan Turing Institute
aelliott@turing.ac.uk

Mohammad Mahdi Khalili
The Ohio State University
Yahoo Research
khalili.17@osu.edu

Abstract

In several real-world applications (e.g., online advertising, item recommendations, etc.) it may not be possible to release and share the real dataset due to privacy concerns. As a result, synthetic data generation (SDG) has emerged as a promising solution for data sharing. While the main goal of private SDG is to create a dataset that preserves the privacy of individuals contributing to the dataset, the use of synthetic data also creates an opportunity to improve fairness. Since there often exist historical biases in the datasets, using the original real data for training can lead to an unfair model. Using synthetic data, we can attempt to remove such biases from the dataset before releasing the data. In this work, we formalize the definition of fairness in synthetic data generation and provide a general framework to achieve fairness. Then we consider two notions of counterfactual fairness and information filtering fairness and show how our framework can be used for these definitions.

1 INTRODUCTION

The availiability of high quality unbiased data has become a bottleneck for the development of machine learning methods in several applications. For example, in healthcare and finance, privacy concerns and regulations may not allow data holders to release the data reducing the availability of data. One form of Synthetic Data Generation (SDG), namely, differentially private synthetic data generation (DP-SDG) can be a promising solution for the data sharing problem (Xie et al., 2018; Jordon et al., 2018). These models attempt to create a dataset that resembles the real data while satisfying some level of privacy (Dwork, 2006). In general, when evaluating SDG models, the criteria for measuring the performance are usually fidelity, diversity, and privacy. The first two criteria account for the similarity between the generated data and the real data, and the third criterion measures the privacy leakage of the model.

An additional criterion that can be added to these three criteria is fairness. It has been shown that the ML models can reflect pre-existing biases in the training dataset and in some cases exacerbate such biases (Dressel and Farid, 2018; Harwell, 2018; Dastin, 2018; Ganev et al., 2021). Several fairness definitions and methods have been proposed in the literature to attempt to mitigate unfairness and biases (Mehrabi et al., 2021; Zuo et al., 2024; Khalili et al., 2023; Zhang et al., 2022; Khalili et al., 2021a; Wang et al., 2022a; Agarwal et al., 2018). These methods impose an additional constraint (depending on the fairness notion of choice) on the predictor to improve fairness. It is, however, a task for the user of the data to ensure the fairness constraint is satisfied. Using synthetic data creates an interesting opportunity to make sure that the released dataset is fair, and when it is used in a downstream task, the resulting model is also fair. The goal of fair synthetic data generation is to generate a dataset that is as close as possible to the real data while removing the discriminatory biases existing in the data. Inevitably, using this method will alter the distribution of the data and hence decrease the performance of the model trained on the generated data when measured on the original biased data. However, we note attempting to train a fair predictor on real data (rather than synthetic), will also cause some performance drop compared to the case where a fairness constraint is not imposed (Hardt et al., 2016). There are several different notions introduced for fairness, choosing the right notion depends on the policy maker's preference and the task at hand (Binns, 2018). In this work, we propose a general framework for

Proceedings of the 27$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

fair SDG that can be used for various fairness notions. To show the performance of our algorithm, we consider two notions of information filtering (van Breugel et al., 2021) and counterfactual fairness (CF) (Kusner et al., 2017) and provide theoretical and empirical analysis for these two notions.

Contribution: Our contributions can be summarized as follows: 1) In Section 2.2, We formalized the definition of fairness for synthetic data. We also provide a comparison with other existing definitions and point out a few inaccuracies in the literature. 2) More importantly, we provide three propositions to show why our definition is meaningful and why fair SDG leads to a fair predictor in downstream tasks. 3) Finally, we propose a general architecture based on generative adversarial network GANs, and implement our framework for two fairness definitions of information filtering and counterfactual fairness. The former does not need knowledge of the causal model while the latter does.

Related work: Several early works use synthetic data to reduce bias in a dataset, especially by creating additional samples for underrepresented groups (e.g. see Kamiran and Calders (2012)). Another similar approach is data augmentation as a pre-processing step, not for the goal of having additional samples for underrepresented groups, but for achieving a certain fairness notion e.g., Sharma et al. (2020); Feldman et al. (2015); Zhang et al. (2016). In this work, however, we are not attempting to augment or modify the existing dataset with synthesized samples, but our goal is to generate a completely new synthetic dataset using a generative model that is trained on real (and potentially biased) data. This is different from finding a fair representation of the data (Zemel et al., 2013), as here we also need fidelity, i.e., the synthetic data should resemble the real data. Another relevant line of work studies GANs' behavior and shows that they may exacerbate unfairness in some scenarios (Ganev et al., 2021), and some propose methods for improving this effect (Kenfack et al., 2022; Tan et al., 2020). We refer to following surveys for more extensive discussion Alves et al. (2023); Pessach and Shmueli (2022); Bourou et al. (2021). The most relevant works to our setting are Xu et al. (2018); van Breugel et al. (2021); Xu et al. (2019). In Xu et al. (2018), FairGAN, a GAN-based method is introduced to create a dataset that satisfies the demographic parity notion (see Section 2.1 for a formal definition). Their problem setting is slightly different from what we consider here. They want to create a dataset, such that *every* predictor trained on the dataset satisfies their fairness notion; whereas we want to align the incentives of a data user and have a synthetic dataset such that an *accurate* predictor satisfies the fairness notion (see Appendix B for a more detailed discussion). In Xu et al. (2019), CFGAN is proposed to synthesize a dataset satisfying counterfactual fairness (CF). In Section 3, we show that their method is not satisfying CF, but a relevant notion defined in Definition 6. Finally, DECAF (van Breugel et al., 2021), is a causally-aware method that is proposed to generate fair data for a notion called conditional fairness. We have a discussion on their method and the definition they propose in Appendix B.

2 FAIRNESS FOR SYNTHETIC DATA GENERATION (SDG)

In this section, we provide a general definition of fairness in SDG and using Propositions 1 to 3 show that this definition can reduce unfairness for an accurate predictor with respect to real data drawn from true distribution. Let us first review some of the commonly used fairness notions used for supervised models.

2.1 Algorithmic fairness

Algorithmic fairness is a well-established area, with several fairness definitions for supervised learning models (see e.g., Khalili et al. (2021b); Zhang et al. (2019); Dwork et al. (2012); Hardt et al. (2016)). The appropriate notion may be different for different datasets/tasks. Let (X, A, Y) be three random variables where $X \in \mathcal{X}$ denotes the feature vector, $A \in \mathcal{A}$ denotes the sensitive attribute (e.g., race, gender), and $Y \in \mathcal{Y}$ denotes the output/label.[1] Here we provide a few fairness definitions that are relevant to our work (several more can be found in Mehrabi et al. (2021)).

Definition 1. Demographic parity (DP) (Dwork et al., 2012), also known as statistical parity: A predictor $\hat{Y}$ satisfies demographic parity if $\Pr(\hat{Y} = y|A = a) = \Pr(\hat{Y} = y|A = a'), \forall a, a' \in \mathcal{A}, y \in \mathcal{Y}$. Further, δ-Approximate Demographic Parity (δ-ADP) can be defined as follows, $\frac{1}{2}\sum_{y\in\mathcal{Y}} |\Pr(\hat{Y} = y|A = a) - \Pr(\hat{Y} = y|A = a')| \leq \delta, \forall a, a' \in \mathcal{A}, y \in \mathcal{Y}$.

Definition 2. Counterfactual fairness (CF) (Kusner et al., 2017): A predictor $\hat{Y}$ satisfies counterfactual fairness if for any context $A = a$ and $X = x$, $\Pr(\hat{Y}_{A\leftarrow a}(U) = y|X = x, A = a) = \Pr(\hat{Y}_{A\leftarrow a'}(U) = y|X = x, A = a)$ holds for all value of $y \in \mathcal{Y}$ and $a' \in \mathcal{A}$. Here, U is the set of unobserved variables in the causal model, and $A \leftarrow a$ denotes an intervention on variable A.[2]

[1] In this work, we use lower case letters for the realizations of random variables.

[2] We refer readers to Kusner et al. (2017) for details of counterfactual fairness, counterfactual inference, and intervention using structural equations and a causal model. For simplicity, sometimes we use $\Pr(\hat{Y}_{A\leftarrow a} = y|X = x, A = a)$ instead of $\Pr(\hat{Y}_{A\leftarrow a}(U) = y|X = x, A = a)$.

Definition 3. Information filtering (IF) or condition fairness (van Breugel et al., 2021): A predictor $\hat{Y}$ satisfies IF with respect to $X_s \subseteq X$, if $\hat{Y} \perp\!\!\!\perp A | X_s$, i.e., $I(\hat{Y}; A|X_s) = 0$.

Note that the above notions are all defined to assess the fairness of a given predictor. However, in this work, we want to have a "fair dataset", and consequently there is no predictor immediately in the picture. Thus, there will be no variable $\hat{Y}$ in a fair dataset.

2.2 Fairness in SDG setting

Consider a dataset $\mathcal{D} = \{(X_i, A_i, Y_i)\}_{i=1}^n$ where (X_i, A_i, Y_i) follows distribution P. The goal of fair synthetic data generation is to create a dataset $\mathcal{D}' = \{(X_i', A_i', Y_i')\}_{i=1}^{n'}$ such that when $\mathcal{D}'$ is used in a downstream task as the training dataset, the resulting predictor satisfies some fairness notions of choice. Here (X_i', A_i', Y_i') is drawn from another distribution P' and is denoted by $(X_i', A_i', Y_i') \sim P'$. Note that no matter what the training data is, an end user can always create an unfair predictor (e.g., by only accepting men regardless of the rest of the features). Thus, the goal of fair SDG should be aligning incentives of the downstream user, such that maximizing the accuracy results in a fair predictor. Therefore, in fair synthetic data we argue that we should impose the fairness constraint on the labels of the generated data distribution P' instead of the predicted label of a predictor. For example, for demographic parity, we can impose that $P'(Y = y|A = a) = P'(Y = y|A = a'), \forall a, a' \in \mathcal{A}$. Our definition of fair SDG can be formulated as a constrained optimization, where the goal is to find a distribution $P'(X, A, Y)$ such that it satisfies a certain fairness notion (adopting the notation in van Breugel et al. (2021), we denote it by $\varphi((X, Y, A), P')$-fairness), while minimizing the distance P' from the real data distribution P. That is,

$$\min_{P'} d(P, P') \quad s.t. \;\; \varphi((X, Y, A), P') - \text{fairness}, \tag{1}$$

where d is any distributional distance of choice. To show that this definition is helpful, we prove that when an accurate predictor is trained on the fair dataset (following P'), then this predictor will satisfy some level of fairness (Proposition 1). Further, we will be using this predictor on the real data which will be drawn from P. Therefore, we show that the trained predictor will have a reasonable accuracy on P too (Proposition 2), and it will satisfy some level of fairness w.r.t P (Proposition 3). We prove the following propositions for demographic parity (DP) definition. We also show in Appendix A that the results can be extended to the Information filtering notion. We have also provided relevant propositions in Appendix E for extending the results to counterfactual fairness. In the results below, for simplicity, we assume that $\mathcal{X}$ is a countable set and $\mathcal{A} \in \{0, 1\}$. The proofs of these propositions are given in Appendix A. Let us first define the total variation distance.

Definition 4 (**Total Variation Distance**)**.** The total variation distance for two discrete probability distributions P and Q, defined on a countable space Ω, is defined as

$$\mathbb{TV}(P, Q) = \frac{1}{2} \sum_{x \in \Omega} |P(x) - Q(x)|. \tag{2}$$

Note that, when A is binary, δ-ADP (given in Definition 1) can be represented as $\mathbb{TV}(P(f(X)|A = 0), P(Y|A = 0)) \leq \delta$.

Proposition 1. *If a distribution P' satisfies δ-ADP, and a prediction algorithm $f : \mathcal{X} \to \{0, 1\}$ has an error probability of ϵ w.r.t P', i.e., $\Pr\{f(X) \neq Y\} \leq \epsilon$, then we have:*

$$\mathbb{TV}(P'(f(X)|A = 0), P'(f(X)|A = 1)) \leq \epsilon(1/P'(A = 0) + 1/P'(A = 1)) + \delta. \tag{3}$$

Proposition 2. *If a randomized classifier f has an error probability less than ϵ with respect to distribution P', i.e., $\mathbb{E}_{X,Y \sim P'}(1\{f(X) \neq Y\}) \leq \epsilon$,[3] and $\mathbb{TV}(P(X, Y), P'(X, Y)) \leq \delta$, then the error probability of f with respect to the distribution P can be bounded as follows: $\mathbb{E}_{X,Y \sim P}(1\{f(X) \neq Y\}) \leq \epsilon + 2\delta$.*

Proposition 3. *If a prediction algorithm $f : \mathcal{X} \to \{0, 1\}$ satisfies δ_1-ADP w.r.t P', that is $\mathbb{TV}(P'(f(X)|A = 0), P'(f(X)|A = 1)) \leq \delta_1$, and we have $\mathbb{TV}(P, P') \leq \delta_2$, then we have:*

$$\mathbb{TV}(P(f(X)|A = 0), P(f(X)|A = 1)) \leq \delta_2 h(p_0, p_1) + \delta_1, \tag{4}$$

where $p_0 = \min\{P(A = 0), P'(A = 0)\}$ and $p_1 = \min\{P(A = 1), P'(A = 1)\}$, and h is a bounded function that we introduce in the proof.

The above propositions show that if a generated distribution P' is close to P and satisfies δ-ADP, and we have a predictor with an arbitrarily small error, then the predictor will satisfy the fairness notion w.r.t real data with a small error.

3 METHOD

In this section, we provide a GAN implementation method for two fairness notions. First the Information Filtering notion (known as conditional fairness) which is

[3] $1(.)$ denotes an indicator function.

a generalization of fairness through unawareness (FTU) notion (Grgic-Hlaca et al., 2016) and demographic parity. The second is the counterfactual fairness notion (Kusner et al., 2017) which was defined in Definition 2 for the supervised learning setting. The GAN formulation with its flexible loss structure grants us the ability to match the formulation we proposed in equation (1) for different notions of fairness. To be explicit, we can take into account the fairness constraint by adding an additional loss to the generator loss which encourages the fairness notion of choice. We can then trade off the accuracy of reconstruction and fairness by varying relevant hyperparameters as we will see in the next few sections.

3.1 Information filtering fairness (conditional fairness)

Recall that the goal of fair SDG is to align the motivation of the data user for obtaining good accuracy, with satisfying the fairness constraint. For information filtering, we propose to impose the constraint $I(Y; A|X_s) = 0$ on the generated synthetic data distribution P', where X_s is a subset of features. This means that there is no information between Y and A given X_s, i.e., A may influence Y only via X_s. Assuming that X_s is the set of non-sensitive and useful attributes, this constraint ensures that the sensitive attribute does not influence the output from any other potential routes not included in X_s. For example, in the Berkeley alleged sex bias case, the reason that female applicants were rejected more often than male applicants was that they were applying to departments with lower admission rates more often compared to men. This should not be considered "unfair" (Chiappa, 2019). Hence, for example, enforcing $I(Y; A) = 0$ is not ideal. Instead, we can enforce $I(A; Y|X_s) = 0$ where here X_s is the department that the applicant has applied to. Note that if we wish to satisfy $I(Y; A) = 0$, we can accommodate this by setting $X_s = \emptyset$. IF is also related to the path-specific fairness notion introduced by Chiappa (2019) where some paths affecting output are deemed to be fair and some are unfair. In contrast to the path-specific fairness notion, the IF notion does not need any information on the underlying causal graph.

Connection to fairness through unawareness (FTU) and demographic parity (DP): FTU for a predictor holds when it does not explicitly use the sensitive attribute, i.e., when the output of a predictor f is only a function of X, not A. Note that in this case, $I(f(X); A|X) = 0$ holds. Therefore, FTU is connected to the IF notion when we choose $X_s = X$. Also, choosing $X_s = \emptyset$ means enforcing $I(Y; A) = 0$ which is equivalent to the demographic parity notion.

Implementation: To implement the information filtering notion, we use a GAN with an additional penalty term in the generator loss, as can be seen in equation (7). Note, this is in line with equation (1),where the first term ensures that P and P' are close and the second enforces the fairness constraint.

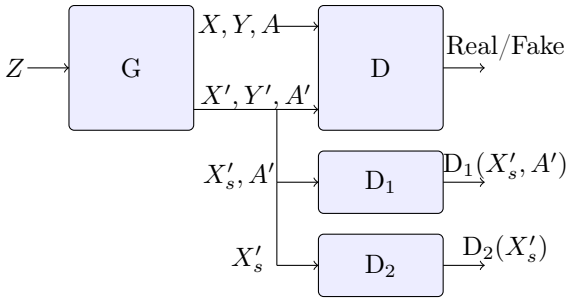

Figure 1: Information Filtering Fairness

To derive the functional form, we note that when $I(Y; A|X_s) = 0$ holds, then using A in addition to X_s will not help for the task of predicting Y. Hence, if we compare the two scenarios where in one, X_s is given as the input to an ideal decoder for predicting Y (referred to as D_2), and in the other scenario, both X_s and A are given to the decoder (which we will refer to as D_1), we expect both of the decoders to have equal performance in predicting Y. This observation encourages us to use the penalty term in equation (7). See Figure 1 for a graphical structure of the proposed method.

Putting this all together, mathematically, we have four networks/losses including the discriminator D, two decoders D_1 and D_2, and generator G. Two decoders D_1 and D_2 are trained to predict the synthetically generated Y', from (X', A'), and X', respectively. Here, we use the cross-entropy (CE) loss to measure the performance of both decoders D_1 and D_2. The input of generator G is Gaussian noise denoted by random variable Z.

$$\begin{aligned}\mathcal{L}_D =& \mathbb{E}_{x,y,a\sim P_{(X,A,Y)}} \log D(x, y, a) \\ &+ \mathbb{E}_{z\sim P(Z)}[\log(1 - D(G(z))]\end{aligned} \tag{5}$$

$$\mathcal{L}_{D_1} = \mathrm{CE}(Y', \mathrm{D}_1(A', X'_s)), \quad \mathcal{L}_{D_2} = \mathrm{CE}(Y', \mathrm{D}_2(X'_s)) \tag{6}$$

$$\begin{aligned}\mathcal{L}_G =& -\mathbb{E}_{z\sim P(Z)}[\log(1 - D(G(z))] \\ &+ \lambda \left(\mathbb{E}_{z\sim P(Z)}[|\mathcal{L}_{D_1} - \mathcal{L}_{D_2}|]\right)\end{aligned} \tag{7}$$

Further, for the special case where $X_s = \emptyset$, we note that $\mathcal{L}_{D_2} = C > \mathcal{L}_{D_1}$, and therefore $|\mathcal{L}_{D_1} - \mathcal{L}_{D_2}| \approx C - \mathcal{L}_{D_1}$, and therefore the generator loss simplifies to $\mathcal{L}_G = -\mathbb{E}_{z\sim P(Z)}[\log(1 - D(G(z))] - \lambda\mathbb{E}_{z\sim P(Z)}[\mathcal{L}_{D_1}]$.

Remark. An additional benefit of using synthetic data for fairness is that it allows us to impose multiple

fairness constraints on the released dataset simultaneously. For example, consider a scenario where there are two potential outputs of interest Y_1 and Y_2 in the dataset, and two sensitive attributes A_1 and A_2. Then, we can impose two constraints $I(Y_1; A_1|X) = 0$ and $I(Y_2; A_2|X) = 0$ by adding two penalty terms in equation (7).

3.2 Counterfactual fairness

In this section, we provide a method to create a dataset that is counterfactually fair. The definition of counterfactual fairness for a predictor was given in Section 2, which can be modified as follows for a SDG:

Definition 5. A generator producing samples (X, A, Y) with distribution P' is counterfactually fair if:

$$P'(Y_{A\leftarrow a} = y|X = x, A = a) = P'(Y_{A\leftarrow a'} = y|X = x, A = a),$$

for all $y \in \mathcal{Y}, x \in \mathcal{X}, a, a' \in \mathcal{A}$.

Here we note that a SDG method is proposed in Xu et al. (2019) to satisfy counterfactual fairness. Although they consider the counterfactual fairness condition in Kusner et al. (2017), and claim that they are attempting to satisfy this condition, they are actually considering interventions, not counterfactuals, and thus their method should be considered as an attempt to satisfy the following fairness constraint proposed in Kilbertus et al. (2017):

Definition 6. (Discrimination avoiding through causal reasoning): A generator said to be fair if the following holds:

$$P(Y = y|X = x, do(A = a)) = P(Y = y|X = x, do(A = a')).$$

This definition is based on the *do* operator (intervention) and thus is different from Definition 2, which considers counterfactuals. For more details about the differences between these two regimes see supplementary material of Kusner et al. (2017) and Glymour et al. (2016)[Chapter 4].

Consider the causal structure in Figure 3 (we refer to Pearl (2009) for an introduction to causal structures). Here we assume that we have the observed features X, sensitive attribute A, label Y, and an unobserved variable U. Following the example in Kusner et al. (2017), we can consider the law school admission problem, where X is a three-dimensional feature vector. The features include GPA, entrance exam score (LSAT), and gender. A is race and Y is the first-year average grade (the output). Here U, the unobserved variable, is considered as the true knowledge of law of the student. The proposed method is presented in Figure 2. We are using a conditional GAN structure, where A is fed to the generator as the condition and the generator

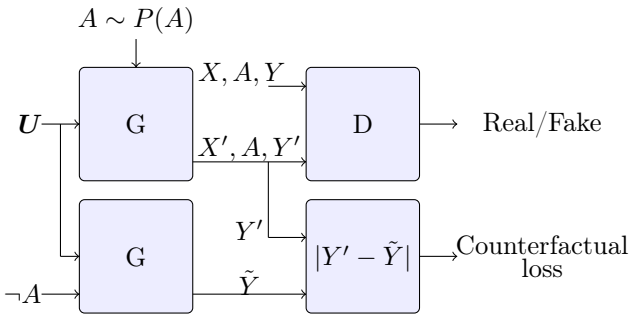

Figure 2: Counterfactual Fairness

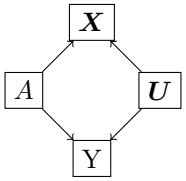

Figure 3: Underlying Causal Model

produces X' and Y', given the sensitive attribute, and random variable U. We are assuming here, that U is the unobserved variable. This architecture is consistent with the causal model. Our goal is to force G to produce the same output for the same individual (i.e., when U is fixed) in the counterfactual world. Note that this is a stronger condition comparing to the Definition 5 (in Proposition 4, we show that this is a sufficient condition). Hence, we add an additional penalty term to the generator loss to enforce CF constraint. Let us denote the output of the generator by $G(u, a)$ where u is an instance of random variable U and a is a sample of A. Also, we denote by $G_Y(u, a)$ the generated output Y'. Using a vanilla GAN our modified loss functions for the generator and discriminator are as follows:

$$\begin{aligned}\mathcal{L}_D =& \mathbb{E}_{x,a,y\sim P(X,A,Y)} \log D(x, a, y) \\ &+ \mathbb{E}_{u,a\sim P(U)P(A)}[\log(1 - D(G(u, a))] \end{aligned} \tag{8}$$

$$\begin{aligned}\mathcal{L}_G =& -\mathbb{E}_{u,a\sim P(U)P(A)}[\log(1 - D(G(u, a)) \\ &- \lambda(G_Y(u, a) - G_Y(u, \neg a))^2] \end{aligned} \tag{9}$$

Again, the parameter λ controls the trade-off between closeness to the true distribution and satisfying the fairness notion. Comparing equation (9) with equation (1), the first term of equation (9) attempts to minimize the distance between P' and P (when using vanilla GAN this will be Jensen-Shannon distance (Goodfellow et al., 2014), note that in experiments we use Wasserstein distance which is known to be more stable (Arjovsky et al., 2017)). The second term in equation (9) encourages the model to satisfy fairness constraint in equation (1). For achieving counterfactual fairness, one sufficient condition that is proposed in Kusner et al. (2017) is to only use variables that are non-descendent of A in

the causal graph, i.e. U in our setting, to produce Y. Thus, the fact that we are using A in our structure to create Y might seem counter-intuitive. However, note that not using descendent of sensitive attributes is only a sufficient condition, and it is possible to achieve counterfactual fairness, while using these variables if the model cancels out their effect, and this is exactly what the counterfactual loss does. If the counterfactual loss is zero, then we have the following proposition.

Proposition 4. *If the generator G has zero counterfactual loss, i.e., $G_Y(u, a) = G_Y(u, a')$ for all $u \in \mathcal{U}$ and $a, a' \in \mathcal{A}$, then the produced distribution $P'(X, A, Y)$ satisfies the counterfactual fairness constraint given in equation* (8).

Proof. See the appendix for the proof.

3.2.1 Generalizing the causal model

Up to this point, we introduced our method for a specific causal model. Here we want to show that using a technique similar to what is proposed (van Breugel et al., 2021)[Section 5.1], our method can be generalized to any given causal model represented by a directed acyclic graph (DAG). Following Pearl's notation (Pearl et al., 2000), consider a triple of (U, V, F), where V is all of the observed variables (in our setting X, A, and Y), U represents all unobserved variables, and F is the set of functions $\{f_1, \cdots, f_n\}$, that are corresponding to each $V_i \in V$, such that $V_i = f_i(pa_i, U_{pa_i})$, where $pa_i \subseteq V \setminus \{V_i\}$ and U_{pa_i} are observable and unobservable parents of V_i, these are called structural equations. Now each of these equations can be modeled by a separate generator $G_i : \mathbb{R}^{|pa_i|+|U_{pa_i}|} \to \mathbb{R}$. Features are generated sequentially following the order imposed by the underlying causal graph. Assuming that parents of V_i are already generated, we generate $\hat{V}_i = G_i(\hat{pa}_i, U_{pa_i})$, where $\hat{pa}_i$ are generated parents of V_i. The fact that the underlying graph is acyclic, enables this ordering. We refer to (van Breugel et al., 2021)[Section 5.1] for a detailed explanation of this method. For imposing CF, we can penalize the generator similar to equation (9). The value of A should first be flipped, while all unobserved values are fixed. Then we update the value of all descendants of A including potentially Y, and compute the counterfactual loss. Note that if Y is not a descendent of A then counterfactual fairness will hold by definition. A more detailed discussion and an example are given in supplementary material for the general causal model.

Remark. In our setting, the case where A is not a binary variable can also be handled. Instead of flipping A we can choose uniformly at random a value from $\mathcal{A} \setminus \{A\}$ and compute the loss. Alternatively, we can compute Y for all $a' \in \mathcal{A} \setminus \{A\}$ and define the penalty to be the average of counterfactual loss for all a'.

Remark. Note that it is possible to make our proposed models differentially private by using DP-SGD (Abadi et al., 2016). This is true for all discussed GAN-based methods e.g. FairGAN (Xu et al., 2018), DECAF (van Breugel et al., 2021), and CFGAN (Xu et al., 2019).

4 EXPERIMENTS

4.1 Information filtering notion

For the Information filtering notion, we replicate a benchmark from (van Breugel et al., 2021) which explores the fairness, and utility of classifiers trained on fair (and not fair) synthetic data using the Adult dataset (Dua and Graff, 2017). To do so we leverage the software implementation from Wang et al. (2022b), a replication study of the original paper.

The output in the dataset is a binary label that determines whether the individual's income is over 50k. It is known that bias between gender and income exists in this dataset (Feldman et al., 2015; Zhang et al., 2016). Thus, here we consider A to be the gender of the individual, Y is the binary label, and X is the rest of the features. We split the real dataset into training and test datasets with 2000 elements in the test set. We use the training dataset to train a generative model for synthetic data generation.

We briefly detail the benchmark (for full details see (van Breugel et al., 2021; Wang et al., 2022b)). We create the synthetic data using different baselines (see Table 1), and measure the quality and fairness of the generated samples (we also have the real data, named original as one of the baselines). For each row of the Table 1, similar to (Sajjadi et al., 2018; Kynkäänniemi et al., 2019), we evaluate the quality of the data via calculating the precision/recall/AUROC of the trained classifiers.

To evaluate fairness in the generated synthetic data, we consider FTU and DP fairness notions (as we discussed in Section 3.1, FTU and DP are two special cases of information filtering.) FTU is measured by computing the difference in the predictions when all features are the same and we only change A. In other words, if $g(x, a)$ is a classifier, the FTU is evaluated by $|\frac{1}{n}\sum_i g(x_i, 0) - \frac{1}{n}\sum_i g(x_i, 1)|$ (smaller value implies that g is better aligned with FTU notion). The DP is measured by computing the total variation distance. That is, $\frac{1}{n}|\sum_{\{i:a_i=0\}} g(x_i, a_i) - \sum_{\{i:a_i=1\}} g(x_i, a_i)|$.

We repeat our experiment 10 times with a different (random) training test split and calculate the mean and standard deviation. In our experiment, we use a Logistic Regression (LR) classifier trained by the

Table 1: SDG Performance in Terms of Quality and Fairness on Adult Dataset (Dua and Graff, 2017) we use the same procedure as van Breugel et al. (2021); Wang et al. (2022b) using a Logistic Regression

Method	Precision	Recall	AUROC	FTU	DP
Original	0.862 ± 0.005	0.933 ± 0.007	**0.885 ± 0.007**	0.065 ± 0.005	0.187 ± 0.015
GAN	0.781 ± 0.019	0.975 ± 0.034	0.796 ± 0.041	0.040 ± 0.048	0.052 ± 0.055
WGAN-GP	0.795 ± 0.046	0.315 ± 0.135	0.547 ± 0.042	0.051 ± 0.038	0.065 ± 0.032
FairGAN	0.771 ± 0.006	**0.993 ± 0.006**	0.774 ± 0.056	0.017 ± 0.011	0.029 ± 0.013
DECAF-ND	0.881 ± 0.022	0.783 ± 0.043	0.802 ± 0.008	0.149 ± 0.072	0.347 ± 0.065
DECAF-FTU	**0.884 ± 0.027**	0.778 ± 0.050	0.801 ± 0.006	0.019 ± 0.014	0.294 ± 0.074
DECAF-CF	0.779 ± 0.012	0.930 ± 0.024	0.745 ± 0.012	0.005 ± 0.003	0.039 ± 0.029
DECAF-DP	0.753 ± 0.003	0.957 ± 0.028	0.687 ± 0.018	**0.003 ± 0.003**	**0.011 ± 0.010**
IF $X_s = \emptyset$ ($\lambda = 0.001$)	0.829 ± 0.006	0.936 ± 0.008	0.846 ± 0.014	0.021 ± 0.016	0.098 ± 0.026
IF $X_s = \emptyset$ ($\lambda = 0.003$)	0.822 ± 0.008	0.937 ± 0.006	0.836 ± 0.011	0.092 ± 0.016	0.049 ± 0.028
IF $X_s = \emptyset$ ($\lambda = 0.005$)	0.823 ± 0.006	0.93 ± 0.0108	0.833 ± 0.0124	0.153 ± 0.046	0.022 ± 0.016
IF $X_s = \emptyset$ ($\lambda = 0.007$)	0.82 ± 0.0068	0.928 ± 0.012	0.827 ± 0.012	0.170 ± 0.034	0.027 ± 0.020
IF $X_s = \emptyset$ ($\lambda = 0.009$)	0.824 ± 0.0046	0.924 ± 0.010	0.827 ± 0.011	0.217 ± 0.043	0.025 ± 0.015
IF $X_s = X$ ($\lambda = 10^{-3}$)	0.849 ± 0.012	0.912 ± 0.010	0.833 ± 0.011	0.071 ± 0.02	0.192 ± 0.021
IF $X_s = X$ ($\lambda = 10^{-2}$)	0.881 ± 0.0125	0.854 ± 0.046	0.85 ± 0.0118	0.074 ± 0.028	0.277 ± 0.051
IF $X_s = X$ ($\lambda = 10^{-1}$)	0.872 ± 0.011	0.88 ± 0.022	0.856 ± 0.00812	0.041 ± 0.026	0.231 ± 0.044
IF $X_s = X$ ($\lambda = 10^{0.0}$)	0.873 ± 0.012	0.879 ± 0.014	0.860 ± 0.008	0.037 ± 0.023	0.235 ± 0.033

Sklearn library (Pedregosa et al., 2011). The results are shown in Table 1. Again, similar to (van Breugel et al., 2021; Wang et al., 2022b), we compare against a GAN, a WGAN (Gulrajani et al., 2017), FairGAN (Xu et al., 2018), and several variations of DECAF (van Breugel et al., 2021) including DECAF-ND (No Debiasing), DECAF-FTU (Fairness Through Unawareness), DECAF-CF (Conditional Fairness), DECAF-DP (Demographic Parity).

To satisfy DP, we set $X_s = \emptyset$. In Table 1, we observe a reasonable classifier performance and a reduction in DP as we increase λ as we would expect with a wider range of values of λ presented in Appendix D. Comparing with other methods, in Table 1, we can see that our method with $\lambda = 0.009$ has a similar mean DP level as FairGAN, DECAF-CF, and DECAF-DP. However, our algorithms can achieve better AUROC compared to the baselines. This implies that our algorithm can improve the fairness-accuracy trade-off.

For the case where $X_s = X$, the key fairness metric is FTU. Again, we observe reasonable classifier performance with high precision, recall, and AUROC, and a reduction in FTU as we increase λ as we would expect. We note that on average our algorithm with $X_s = X$ and $\lambda = 1$ has the highest AUROC while its FTU level is similar to FairGAN, DECAF-FTU, DECAF-CF, and DECAF-DP. This observation implies that our algorithm with $X_s = X$ improves the trade-off between FTU and accuracy.

In addition to the wider range of λs for the logistic regression case, we provide a variant of this experiment with a Multi-Layer Perceptron (MLP) model and related discussion in Appendix D due to space constraints) which highlight the importance of appropriate choice of λ. Full results can be found in Appendix Table 4. We want to point out that a reasonable variation in some of the statistics has noted due to variation in the GAN model used in the synthetic methods, and also in the classifiers, especially in the MLP case.

4.2 Comparison with in-processing fairness

The goal of this section is to compare the performance of a logistic regression model trained on synthetic data and a "fair" logistic regression model trained on real data by FairBatch (Roh et al., 2020). We investigate the performance of these two models in terms of accuracy and fairness level (Figure 4 left panel). Note that FairBatch is an in-processing algorithm for fair model training. This algorithm measures the rate of positive labels predicted by the model for each group in each epoch, and it changes the mini-batch sampling distribution in favor of the group that is disadvantaged. The figure shows that FairBatch has better accuracy. However, using synthetic data has the advantage that we can be sure that the fairness level of choice is observed by the user.

4.3 Law school experiment for counterfactual fairness

Similar to Kusner et al. (2017), here we use the law school admission example to evaluate our counterfactually fair generator. The dataset is constructed via a survey conducted by The Law School Admission Council across 163 law schools in the United States (Wightman, 1998). It contains information about 21,790 students. We have access to the following features: the entrance exam scores (LSAT), the grade-point average (GPA), the first-year average grade (FYA), along with race and gender of students. Now the goal is to predict FYA as a proxy for the student's success using other

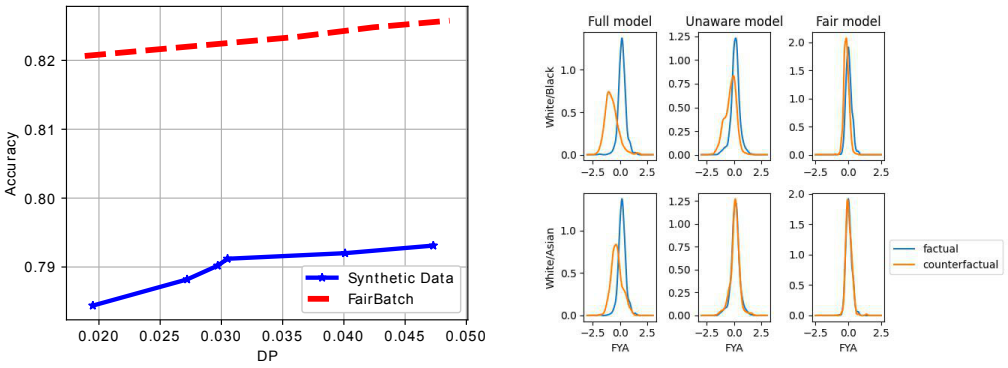

Figure 4: Additional results for our GAN based approach. **(Left)** Comparison of accuracy vs DP for a Logistic Regression (LR) model trained on real data (Adult dataset) using FairBatch method and a LR trained on a fair synthetic version of data. **(Right)** The distribution of estimated FYA for three models. For the fair model, we chose $\lambda = 0.004$. The factual curve is representing the White race, and the counterfactual curve is Black, and Asian in rows one and two respectively.

features in the dataset. Here we only consider race as the sensitive attribute, as Kusner et al. (2017) found that the data is counterfactually fair w.r.t gender. Our goal is to train a counterfactually fair regression model to predict FYA.

We denote the real dataset by $\mathcal{D}$ and split this dataset 80/20 into $\mathcal{D}_{\text{train}}/\mathcal{D}_{\text{test}}$ subsets. First, we use D_{train} to train the generative model proposed in Section 3.2 Counterfactual fairness Section and produce fair dataset $\mathcal{D}'$ of the same size as D_{train}. Now we train a linear regression model[4] on $\mathcal{D}'$, the fair dataset, to estimate FYA using the rest of the features (i.e., LSAT, GPA, race, and gender). Denote this regression model with f_{fair}. We also train two baseline models using the real data D. f_{full} is a linear regression model trained on D_{train} using all features (we expect this model to have the best accuracy and to be the most unfair model). Secondly, f_{unaware} is another linear regression model using only GPA and LSAT (thus is unaware of sensitive attributes). The accuracy of these models in terms of RMSE on the test data (D_{test}) is reported in Table 2. We can see that there is a slight reduction in the accuracy when we use synthetic data instead of real data. Also, as we increase λ the accuracy decreases.

For assessing whether the learned algorithm f_{fair} is counterfactually fair, the distribution of estimated FYA for both factual and counterfactual samples are required. To produce counterfactual samples, similar to Kusner et al. (2017), we fit the following linear model described in equation (10) as a counterfactual simulator on the data. Note that we are interested in satisfying the CF notion for the real data, thus we use $\mathcal{D}$ to train the simulator. This model follows the causal graph in Figure 3. In this example, R represents race, S is gender, and U is the unobserved variable representing the true knowledge of the law. Similar to Kusner et al. (2017) we use an MCMC model to estimate coefficients in equation (10) and also to estimate the value of U for each sample in the test data.

$$\begin{aligned} \text{GPA} &\sim \mathcal{N}(b_G + w_G^U U + w_G^R R + w_G^S S,\, \sigma_G), \\ \text{FYA} &\sim \mathcal{N}(w_F^U U + w_F^R R + w_F^S S,\, 1) \\ \text{LSAT} &\sim \text{Poisson}(\exp(b_L + w_L^U U + w_L^R R + w_L^S S)), \\ U &\sim \mathcal{N}(0, 1) \end{aligned} \tag{10}$$

For a given sample in the test data, we use the model simulator to get the counterfactual samples and then feed both of these samples to f_{fair} and also two baseline methods. The distribution of estimated FYA for the factual and counterfactual samples are shown in Figure 4 (right panel). We expect these two distributions to align if CF holds. In the right panel of Figure 4, we consider two scenarios. In the first row, we compare white vs black (white is factual, and black is counterfactual), and it can be observed that the full model is not fair. Unaware model improves the fairness, and the fair model is the fairest of the three. In the second row, we compare white and Asian and can see that both unaware and fair models seem to satisfy the CF constraint for this pair of sensitive attributes. We also want to report that we observe some instability in our GAN model (even without the fairness penalty) which could be because of small dataset, and therefore it is important to check the resultant distributions.

5 CONCLUSION AND FUTURE WORK

In this paper, we formalized a definition of fairness for a synthetic data generator and showed why this definition is useful. We also considered two fairness notions of information filtering and counterfactual fairness and

[4]Using the default model of sklearn (Pedregosa et al., 2011)

Table 2: RMSE of Fair and Baseline Models.

	RMSE
f_{full}	0.2486 ± 0.0011
f_{unaware}	0.2539 ± 0.0009
f_{fair} $(\lambda = 0.002)$	0.2608 ± 0.0015
f_{fair} $(\lambda = 0.004)$	0.2680 ± 0.0083
f_{fair} $(\lambda = 0.006)$	0.2708 ± 0.0113

proposed a new method for implementing these notions using GANs. However, our framework is more general and in fact any notion (group or individual) can be potentially enforced as long as the constraint can be represented as a regularization term. In Section 2, we considered an estimator with a good accuracy to state our three propositions. Due to insufficient information in inputs, finding an accurate estimator may not be possible. Thus, it is useful to find similar propositions for the Bayes estimator instead of the accurate estimator. Also, here we proved our results for notions that we considered, finding a more generic proposition is another direction for future work. In Section 3, we mentioned the possibility of using synthetic data for enforcing multiple fairness constraints together, exploration of this idea is another research direction.

Acknowledgment

This work is partially supported by the NSF under grants IIS-2301599 and ECCS-2301601.

References

Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018.

James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2018.

Cynthia Dwork. Differential privacy. In *Proceedings of the 33rd international conference on Automata, Languages and Programming-Volume Part II*, pages 1–12. Springer-Verlag, 2006.

Julia Dressel and Hany Farid. The accuracy, fairness, and limits of predicting recidivism. *Science advances*, 4(1):eaao5580, 2018.

Drew Harwell. The accent gap. `https://www.washingtonpost.com/graphics/2018/business/alexa-does-not-understand-your-accent/?utm_term=.ca17667575d1`, 2018.

Jeffrey Dastin. Amazon scraps secret ai recruiting tool that showed bias against women. `https://www.reuters.com/article/us-amazon-com-jobs-automation-insight`, 2018.

Georgi Ganev, Bristena Oprisanu, and Emiliano De Cristofaro. Robin hood and matthew effects–differential privacy has disparate impact on synthetic data. *arXiv preprint arXiv:2109.11429*, 2021.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

Zhiqun Zuo, Mahdi Khalili, and Xueru Zhang. Counterfactually fair representation. *Advances in Neural Information Processing Systems*, 36, 2024.

Mohammad Mahdi Khalili, Xueru Zhang, and Mahed Abroshan. Loss balancing for fair supervised learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 16271–16290. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/khalili23a.html`.

Xueru Zhang, Mohammad Mahdi Khalili, Kun Jin, Parinaz Naghizadeh, and Mingyan Liu. Fairness interventions as (dis) incentives for strategic manipulation. In *International Conference on Machine Learning*, pages 26239–26264. PMLR, 2022.

Mohammad Mahdi Khalili, Xueru Zhang, and Mahed Abroshan. Fair sequential selection using supervised learning models. *Advances in Neural Information Processing Systems*, 34:28144–28155, 2021a.

Yihe Wang, Mohammad Mahdi Khalili, and Xiang Zhang. Towards fair representation learning in knowledge graph with stable adversarial debiasing. In *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 901–909. IEEE, 2022a.

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International conference on machine learning*, pages 60–69. PMLR, 2018.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

Reuben Binns. Fairness in machine learning: Lessons from political philosophy. In *Conference on Fairness, Accountability and Transparency*, pages 149–159. PMLR, 2018.

Boris van Breugel, Trent Kyono, Jeroen Berrevoets, and Mihaela van der Schaar. Decaf: Generating fair

synthetic data using causally-aware generative networks. *Advances in Neural Information Processing Systems*, 34, 2021.

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *Advances in neural information processing systems*, 30, 2017.

Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and information systems*, 33(1): 1–33, 2012.

Shubham Sharma, Yunfeng Zhang, Jesús M Ríos Aliaga, Djallel Bouneffouf, Vinod Muthusamy, and Kush R Varshney. Data augmentation for discrimination prevention and bias disambiguation. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 358–364, 2020.

Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2015.

Lu Zhang, Yongkai Wu, and Xintao Wu. A causal framework for discovering and removing direct and indirect discrimination. *arXiv preprint arXiv:1611.07509*, 2016.

Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International conference on machine learning*, pages 325–333. PMLR, 2013.

Patrik Joslin Kenfack, Kamil Sabbagh, Adín Ramírez Rivera, and Adil Khan. Repfair-gan: Mitigating representation bias in gans using gradient clipping. *arXiv preprint arXiv:2207.10653*, 2022.

Shuhan Tan, Yujun Shen, and Bolei Zhou. Improving the fairness of deep generative models without retraining. *arXiv preprint arXiv:2012.04842*, 2020.

Guilherme Alves, Fabien Bernier, Miguel Couceiro, Karima Makhlouf, Catuscia Palamidessi, and Sami Zhioua. Survey on fairness notions and related tensions. *EURO Journal on Decision Processes*, page 100033, 2023.

Dana Pessach and Erez Shmueli. A review on fairness in machine learning. *ACM Computing Surveys (CSUR)*, 55(3):1–44, 2022.

Stavroula Bourou, Andreas El Saer, Terpsichori-Helen Velivassaki, Artemis Voulkidis, and Theodore Zahariadis. A review of tabular data synthesis using gans on an ids dataset. *Information*, 12(09):375, 2021.

Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. Fairgan: Fairness-aware generative adversarial networks. pages 570–575, 2018.

Depeng Xu, Yongkai Wu, Shuhan Yuan, Lu Zhang, and Xintao Wu. Achieving causal fairness through generative adversarial networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019.

Mohammad Mahdi Khalili, Xueru Zhang, Mahed Abroshan, and Somayeh Sojoudi. Improving fairness and privacy in selection problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8092–8100, 2021b.

Xueru Zhang, Mohammadmahdi Khaliligarekani, Cem Tekin, and mingyan liu. Group retention when using machine learning in sequential decision making: the interplay between user dynamics and fairness. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper_files/paper/2019/file/7690dd4db7a92524c684e3191919eb6b-Paper.pdf`.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

Nina Grgic-Hlaca, Muhammad Bilal Zafar, Krishna P Gummadi, and Adrian Weller. The case for process fairness in learning: Feature selection for fair decision making. In *NIPS symposium on machine learning and the law*, volume 1, page 2. Barcelona, Spain, 2016.

Silvia Chiappa. Path-specific counterfactual fairness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7801–7808, 2019.

Niki Kilbertus, Mateo Rojas Carulla, Giambattista Parascandolo, Moritz Hardt, Dominik Janzing, and Bernhard Schölkopf. Avoiding discrimination through causal reasoning. *Advances in neural information processing systems*, 30, 2017.

Madelyn Glymour, Judea Pearl, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *In-*

ternational conference on machine learning, pages 214–223. PMLR, 2017.

Judea Pearl et al. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, 19:2, 2000.

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Shuai Wang, Paul Verhagen, Jennifer Zhuge, and Velizar Shulev. Replication study of decaf: Generating fair synthetic data using causally-aware generative networks. In *ML Reproducibility Challenge 2021 (Fall Edition)*, 2022b.

Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. *Advances in neural information processing systems*, 31, 2018.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

Yuji Roh, Kangwook Lee, Steven Euijong Whang, and Changho Suh. Fairbatch: Batch selection for model fairness. In *International Conference on Learning Representations*, 2020.

Linda F Wightman. Lsac national longitudinal bar passage study. lsac research report series. 1998.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A Proofs

Proof of Proposition 1: Since $\mathbb{TV}$ satisfies triangle inequality, we have

$$\begin{aligned}\mathbb{TV}(P'(f(X)|A=0), P'(f(X)|A=1)) \leq\ & \mathbb{TV}(P'(f(X)|A=0), P'(Y|A=0)) \\ &+ \mathbb{TV}(P'(Y|A=0), P'(Y|A=1)) \\ &+ \mathbb{TV}(P'(Y|A=1), P'(f(X)|A=1)).\end{aligned} \tag{11}$$

The second term above is bounded by δ since P' satisfies δ-ADP. Note that, when A is binary, the definition of δ-ADP (given in Definition 1) can be represented as $\mathbb{TV}(P(f(X)|A=0), P(Y|A=0)) \leq \delta$. Now for the first term of RHS we have:

$$\begin{aligned}&\mathbb{TV}(P'(f(X)|A=0), P'(Y|A=0)) \\ &= \frac{1}{2}\sum_{y\in\{0,1\}} |P'(f(X)=y|A=0) - P'(Y=y|A=0)| \\ &= \frac{1}{2}\sum_{y\in\{0,1\}} |P'(f(X)=y, Y=y|A=0) + P'(f(X)=y, Y=1-y|A=0) - P'(Y=y|A=0)| \\ &= \frac{1}{2}\sum_{y\in\{0,1\}} |P'(f(X)=y, Y=y|A=0) + P'(f(X)=1-y, Y=y|A=0) \\ &- P'(f(X)=1-y, Y=y|A=0) + P'(f(X)=y, Y=1-y|A=0) - P'(Y=y|A=0)| \\ &= \frac{1}{2}\sum_{y\in\{0,1\}} |P'(Y=y|A=0) - P'(f(X)=1-y, Y=y|A=0) \\ &+ P'(f(X)=y, Y=1-y|A=0) - P'(Y=y|A=0)| \\ &= \frac{1}{2}\sum_{y\in\{0,1\}} |P'(f(X)=y, Y=1-y|A=0) - P'(f(X)=1-y, Y=y|A=0)| \\ &\leq \frac{1}{2}\sum_{y\in\{0,1\}} |P'(f(X)=y, Y=1-y|A=0)| \\ &\leq \frac{1}{2}\sum_{y\in\{0,1\}} \frac{P'(f(X)=y, Y=1-y)}{P'(A=0)} \\ &\leq \epsilon/2P'(A=0)\end{aligned}$$

Similarly for the third term of RHS of (1) we have:

$$\mathbb{TV}(P'(Y|A=1), P'(f(X)|A=1)) \leq \epsilon/2P'(A=1).$$

This completes the proof.

Proof of Proposition 2: The error probability of f on P' is less than ϵ , that is

$$Pr\{f(X) \neq Y\} = \mathbb{E}_{X,Y\sim P'(X,Y)}\, 1[f(X) \neq Y] \leq \epsilon. \tag{12}$$

Now, the error probability of f on distribution P can be upper bounded as follows:

$$\begin{aligned}\mathbb{E}_{X,Y\sim P(X,Y)}\, 1[f(X) \neq Y] &= \sum_{x,y} p(x,y) 1[f(x) \neq y] \\ &\leq \sum_{x,y} p'(x,y) 1[f(x) \neq y] + \sum_{x,y} |p(x,y) - p'(x,y)| 1[f(x) \neq y] \\ &\leq \epsilon + \sum_{x,y} |p(x,y) - p'(x,y)| \\ &\leq \epsilon + 2\delta\end{aligned}$$

Proof of Proposition 3: Similar to Proposition 1, using triangle inequality for $\mathbb{TV}$, we have

$$\begin{aligned}\mathbb{TV}(P(f(X)|A=0), P(f(X)|A=1)) \leq\ & \mathbb{TV}(P(f(X)|A=0), P'(f(X)|A=0)) \\ & + \mathbb{TV}(P'(f(X)|A=0), P'(f(X)|A=1)) \\ & + \mathbb{TV}(P'(f(X)|A=1), P(f(X)|A=1)).\end{aligned} \tag{13}$$

The second term above is bounded by δ_1 (since f satisfies ADP for P'), thus the RHS is upper bounded by

$$\mathbb{TV}\left(\frac{P(f(X), A=0)}{P(A=0)}, \frac{P'(f(X), A=0)}{P'(A=0)}\right) + \mathbb{TV}\left(\frac{P'(f(X), A=1)}{P(A=1)}, \frac{P(f(X), A=1)}{P(A=1)}\right) + \delta_1. \tag{14}$$

For the first term we have:

$$\begin{aligned}&\mathbb{TV}\left(\frac{P(f(X), A=0)}{P(A=0)}, \frac{P'(f(X), A=0)}{P'(A=0)}\right) \\ &= \sum_{y\in\{0,1\}} \left|\frac{1}{P(A=0)} P(f(X)=y, A=0) - \frac{1}{P'(A=0)} P'(f(X)=y, A=0)\right|\end{aligned} \tag{15}$$

Now for $y=0$ if the first term in (15) is larger than the second term then we have:

$$\begin{aligned}&\frac{1}{P(A=0)} P(f(X)=0, A=0) - \frac{1}{P'(A=0)} P'(f(X)=0, A=0) \\ &\leq \frac{1}{P(A=0)} P(f(X)=0, A=0) - \frac{1}{P(A=0)+\delta_2} P'(f(X)=0, A=0) && (16) \\ &\leq \frac{z+\delta_2}{P(A=0)} - \frac{z}{P(A=0)+\delta_2} && (17) \\ &\leq \frac{\delta_2(1+P(A=0))+\delta_2^2}{P(A=0)^2} && (18)\end{aligned}$$

In the above equations, we used the general definition of total variation ($\mathbb{TV}(P,Q) = \sup_A |P(A) - Q(A)|$). Now if we consider the other case that the second term in (15) is larger than the first term for $y=0$ we get the following upper bound:

$$\frac{\delta_2(1+P'(A=0))+\delta_2^2}{P'(A=0)^2} \tag{19}$$

Since $g(p) = \frac{\delta_2(1+p)+\delta_2^2}{p^2}$ is a decreasing function in p (for $p>0$) if we define $p_0 = \min\{P(A=0), P'(A=0)\}$ we can bound the (15) with

$$\frac{2\delta_2(1+p_0)+2\delta_2^2}{p_0^2}. \tag{20}$$

Similarly if we define $p_1 = \min\{P(A=1), P'(A=1)\}$, then the second term in (14) can be upper bounded with

$$\frac{2\delta_2(1+p_1)+2\delta_2^2}{p_1^2}. \tag{21}$$

Defining $h(p_0,p_1) = 2(\frac{2+p_0}{p_0^2} + \frac{2+p_1}{p_1^2})$ completes the proof.

Proof of Proposition 4: We want to prove the following equation, assuming that P' is the distribution induced from G which satisfies $G_Y(u,a) = G_Y(u,a')$:

$$P'(Y_{A\leftarrow a} = y | X=x, A=a) = P'(Y_{A\leftarrow a'} = y | X=x, A=a). \tag{22}$$

Let us denote with Q the posterior distribution of U condtioned on A and X. Then the LHS can be written as follows (we assume that $\mathcal{U}$ is a countable set):

$$\begin{aligned}P'(Y_{A\leftarrow a} = y | X=x, A=a) &= \sum_u P'(Y_{A\leftarrow a} = y | X=x, A=a, U=u) Q(U=u|X=x, A=a) && (23) \\ &= \sum_u P'(Y=y|A=a, U=u) Q(U=u|X=x, A=a) && (24)\end{aligned}$$

Similarly we have:

$$P'(Y_{A\leftarrow a'}=y|X=x,A=a)=\sum_u P'(Y_{A\leftarrow a'}=y|X=x,A=a,U=u)Q(U=u|X=x,A=a) \quad (25)$$

$$=\sum_u P'(Y=y|A=a',U=u)Q(U=u|X=x,A=a) \quad (26)$$

Now comparing (24) and (26) and noting that $G_Y(u,a)=G_Y(u,a')$ completes the proof.

B Comparison to previous SDG fairness definitions

Our proposed definition is similar to the definition in FairGAN Xu et al. (2018) and CFGAN Xu et al. (2019). These two works also impose a fairness constraint on the generated samples, i.e., generate P' such that it satisfies a fairness constraint (note that there is no formal definition in these two works). There is however a subtle difference between what FairGAN attempts to achieve and our definition. FairGAN's fairness of choice is demographic parity, which can be satisfied when Y and A are independent, that is $I(Y;A)=0$, where I is the mutual information between the two random variables. However, in the FairGAN implementation, another condition is enforced: $I(\hat{X},\hat{Y};\hat{A})=0$. The rationale for this could be to impose demographic parity for *any* predictor that is trained on the data. Note that, if $I(X;A)=0$ and f is a predictor, then $I(f(X);A)=0$ also holds. This is different from our definition, which imposes fairness only for accurate predictors. Making X and A independent seems very restrictive as there might be important information between X and A that we do not want to lose. This is also relevant to the Information filtering notion discussed in Section 2.1 and a comparison might be helpful. $I(X,Y;A)$ can be expressed as follows

$$I(X,Y;A)=I(Y;A)+I(X;A|Y). \quad (27)$$

When $I(X,Y;A)=0$ is enforced, then we have also $I(Y;A)=0$, which is what we are actually interested in. However, we are also enforcing $I(X;A|Y)=0$ which does not have a natural motivation. Now expanding the above mutual information starting with $I(X;Y)$, we get

$$I(X,Y;A)=I(X;A)+I(Y;A|X). \quad (28)$$

$I(X;Y)$ part of the above expansion contain useful information that we want to keep (and thus this suggests maybe enforcing $I(X,Y;A)=0$ is not ideal). In information filtering notion we attempt to enforce $I(Y;A|X)=0$ on the synthetic data.

B.1 Comparison with DECAF definition:

First, let us review the definition of fair synthetic data given in van Breugel et al. (2021): "A probability distribution $P'(X)$ is $(\mathscr{I}(A,Y),P)$-fair, if and only if the optimal predictor $\hat{Y}=f^*(X)$ of Y trained on $P'(X)$ satisfies $\mathscr{I}(A,Y)$ when evaluated on $P(X)$." This definition is ideal from the fairness point of view. The distribution P' here when used for training a model, will result in a predictor that is fair not on the synthetic training data but on the unseen real data (drawn from P). However, this definition does not guarantee any resemblance of the synthetic data (P') with the actual data (P). Note that the closeness of two distributions (fidelity) is usually a necessity in synthetic data. However, in DECAF work, there is no guarantee that P' and P will be close. In fact, DECAF may produce two distributions P' and P that have an arbitrarily large KL distance. For example, consider a dataset where we have $A\longrightarrow X\longrightarrow Y$. Assume that A is a Bernoulli binary random variable ($P(A=0)=P(A=1)$). When $A=0$, then $X=0$ and when $A=1$ then $X=m$. Also assume that $Y=\mathcal{N}(X,1)$. Now, considering DECAF method, (e.g., for satisfying SP) we need to remove both edges from A to X and then from X to Y, then Y will be either constant or a distribution independent from X and so m (depending on which strategy one chooses as explained in Section 5.2). Thus, it is clear that by increasing m we can have arbitrarily large KL-distance between P and P'.

Proof of Proposition 1 in DECAF van Breugel et al. (2021): The proof of Proposition 1 (main result in DECAF) seems to be incomplete. Firstly, there seems to be a typo in the proof. They consider $f^*(X)$ is the ideal classifier trained on P' (not P), thus we cannot assume that $f^*(X)=P(Y|X)$, we should assume that $f^*(X)=P'(Y|X)$ and then for P' we have the next equation $P'(Y|X)=P'(Y|\partial_{\mathcal{G}'}Y)$. The missing step is that

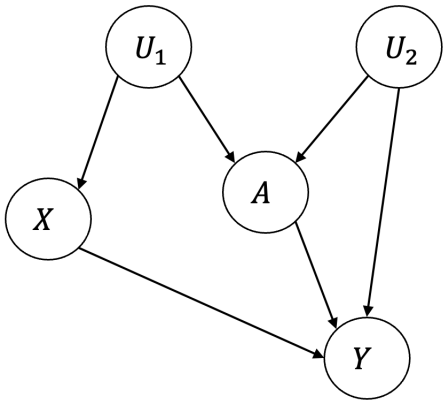

Figure 5: Example of a more complicated causal graph

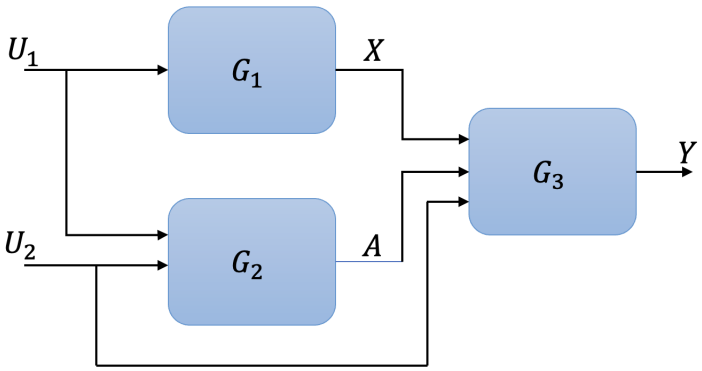

Figure 6: The GAN structure corresponding to Causal Graph in Figure 1

why $P'(Y|\partial_{\mathcal{G}'}Y) = P(Y|\partial_{\mathcal{G}'}Y)$ holds. For instance, using the method suggested in the paper, this equation does not hold in general. Assume that there is only one edge that is removed we have:

$$P'(Y|\partial_{\mathcal{G}'}Y) = P(Y|\partial_{\mathcal{G}'}Y, do(X_i = \tilde{x}_{ij})) \neq P(Y|\partial_{\mathcal{G}'}Y). \tag{29}$$

C Generalization of counterfactual fairness

In this section by an example we explain in more details how to generalize our proposed method for a given causal graph. Consider the causal graph in Figure 5. Here we have two unobserved variables U_1 and U_2, X represents features, A is the sensitive attribute, and Y is the output. Considering this graph, U_1 and U_2 will be the noise for the generators (note that their underlying distribution is known, and also we assumed that unobserved variables are independent). Then we will have two generators G_1 and G_2 to produce X and A given U_1 and U_1, U_2 respectively. Now all variables to generate Y are available. G_3 will have A, X, and U_2 as input and will produce Y as the output. The generator architecture is represented in Figure 5.

The generated samples X, A, Y will be fed to a discriminator. Also, for each sample we can alternate the value of A while the values of U_1 and U_2 and X is fixed to get the counterfactual output and then we can add counterfactual loss to the loss function of the generator.

D Additional results for Information filtering

In the main body we presented the results from a benchmark from van Breugel et al. (2021) which explores the fairness, and utility of classifiers trained on fair (and not fair) synthetic data. using the Adult dataset Dua and Graff (2017).

This section contains additional results on this experiment. Table 3, presents additional values of λ using the same set up as the main paper, demonstrating the performance of the model over these values. Additionally, Table 4 presents the results using an MLP rather than a logistic regression, matching the experiment in Wang et al. (2022b), abet with more variance.

In the MLP case, we note similar performance than LR, although with slightly lower classifier performance. For DP (i.e. when $X_s = \emptyset$) for we see a reduction in DP with increasing λ as we would expect, although with a reversion for high λ - see latter discussion. Although, with large standard deviations in some of the metrics across many of the methods, we have to be careful about interpretation.

For the FTU case ($X_S = X$) We do not observe an optimal scaling of FTU with λ possibly because of the higher variance of MLP. However, we do obtain similar FTU to the comparative methods, and outperform some. Note that the high variance for MLP case is observed even on the original data, but we have also observed reasonable variation in the output of IF, and sensitivity to the values of λ with counter-intuitive effects which we suspect is a consequence of the training process balancing each of the losses. Additional future work will look at improving this, potentially in this case, by exploring the complexity of the decoder architecture as this would allow us to model more complex relationships.

We additionally considering comparing to CFGAN both in the logistic regression and MLP cases however the results from the we obtained we not comparable to the results from the replication experiment from Wang et al. (2022b) (which we followed thoroughly) makes a different choices due to a different discretization in future work we will further explore this comparison

Table 3: Additional results on the performance of the Generative Models in Terms of Quality of Produced Samples and Fairness on Adult Dataset Dua and Graff (2017) we use the same procedure as van Breugel et al. (2021); Wang et al. (2022b) however with a Logistic Regression classifier.

	Data Quality			Fairness	
Method	Precision	Recall	AUROC	FTU	DP
Original	0.862 ± 0.005	0.933 ± 0.007	0.885 ± 0.007	0.065 ± 0.005	0.187 ± 0.015
GAN	0.781 ± 0.019	0.975 ± 0.034	0.796 ± 0.041	0.04 ± 0.048	0.052 ± 0.055
WGAN-GP	0.795 ± 0.046	0.315 ± 0.135	0.547 ± 0.042	0.051 ± 0.038	0.065 ± 0.032
FairGAN	0.771 ± 0.006	0.993 ± 0.006	0.774 ± 0.056	0.017 ± 0.011	0.029 ± 0.013
DECAF-ND	0.881 ± 0.022	0.783 ± 0.043	0.802 ± 0.008	0.149 ± 0.0722	0.347 ± 0.065
DECAF-FTU	0.884 ± 0.027	0.778 ± 0.05	0.801 ± 0.006	0.019 ± 0.014	0.294 ± 0.074
DECAF-CF	0.779 ± 0.012	0.93 ± 0.024	0.745 ± 0.012	0.005 ± 0.003	0.039 ± 0.029
DECAF-DP	0.753 ± 0.003	0.957 ± 0.028	0.687 ± 0.018	0.003 ± 0.003	0.011 ± 0.010
IF $X_s = \emptyset$ ($\lambda = 0.0$)	0.843 ± 0.015	0.918 ± 0.018	0.848 ± 0.007	0.076 ± 0.028	0.182 ± 0.035
IF $X_s = \emptyset$ ($\lambda = 0.001$)	0.829 ± 0.006	0.936 ± 0.008	0.846 ± 0.014	0.021 ± 0.016	0.098 ± 0.026
IF $X_s = \emptyset$ ($\lambda = 0.002$)	0.828 ± 0.007	0.932 ± 0.009	0.84 ± 0.01	0.075 ± 0.026	0.07 ± 0.018
IF $X_s = \emptyset$ ($\lambda = 0.003$)	0.822 ± 0.008	0.937 ± 0.006	0.836 ± 0.011	0.092 ± 0.016	0.049 ± 0.028
IF $X_s = \emptyset$ ($\lambda = 0.004$)	0.823 ± 0.006	0.934 ± 0.008	0.836 ± 0.016	0.097 ± 0.027	0.044 ± 0.019
IF $X_s = \emptyset$ ($\lambda = 0.005$)	0.823 ± 0.006	0.93 ± 0.011	0.833 ± 0.012	0.153 ± 0.046	0.022 ± 0.016
IF $X_s = \emptyset$ ($\lambda = 0.006$)	0.823 ± 0.008	0.932 ± 0.006	0.828 ± 0.013	0.148 ± 0.033	0.028 ± 0.008
IF $X_s = \emptyset$ ($\lambda = 0.007$)	0.82 ± 0.007	0.928 ± 0.012	0.827 ± 0.012	0.17 ± 0.034	0.027 ± 0.020
IF $X_s = \emptyset$ ($\lambda = 0.008$)	0.822 ± 0.007	0.929 ± 0.01	0.831 ± 0.009	0.154 ± 0.031	0.018 ± 0.014
IF $X_s = \emptyset$ ($\lambda = 0.009$)	0.824 ± 0.005	0.924 ± 0.010	0.827 ± 0.011	0.217 ± 0.043	0.025 ± 0.015
IF $X_s = \emptyset$ ($\lambda = 0.100$)	0.894 ± 0.019	0.564 ± 0.016	0.687 ± 0.018	0.056 ± 0.033	0.261 ± 0.028
IF $X_s = X$ ($\lambda = 0$)	0.839 ± 0.007	0.923 ± 0.013	0.846 ± 0.011	0.064 ± 0.014	0.17 ± 0.019
IF $X_s = X$ ($\lambda = 10^{-4}$)	0.843 ± 0.011	0.918 ± 0.013	0.845 ± 0.008	0.073 ± 0.012	0.179 ± 0.028
IF $X_s = X$ ($\lambda = 10^{-3}$)	0.849 ± 0.012	0.912 ± 0.010	0.833 ± 0.011	0.071 ± 0.02	0.192 ± 0.021
IF $X_s = X$ ($\lambda = 10^{-2}$)	0.881 ± 0.013	0.854 ± 0.046	0.85 ± 0.012	0.074 ± 0.028	0.277 ± 0.051
IF $X_s = X$ ($\lambda = 10^{-1}$)	0.872 ± 0.011	0.88 ± 0.022	0.856 ± 0.008	0.0408 ± 0.026	0.231 ± 0.044
IF $X_s = X$ ($\lambda = 10^{-0}$)	0.873 ± 0.012	0.879 ± 0.014	0.86 ± 0.008	0.037 ± 0.023	0.235 ± 0.033
IF $X_s = X$ ($\lambda = 10^{+1}$)	0.839 ± 0.026	0.889 ± 0.043	0.828 ± 0.016	0.161 ± 0.075	0.238 ± 0.105

Table 4: Performance of the Generative Models in Terms of Quality of Produced Samples and Fairness on Adult Dataset Dua and Graff (2017) we use the same procedure as van Breugel et al. (2021); Wang et al. (2022b) using a MLP classifier.

	Data Quality			Fairness	
Method	Precision	Recall	AUROC	FTU	DP
Original	0.878 ± 0.006	0.928 ± 0.007	0.908 ± 0.006	0.019 ± 0.01	0.185 ± 0.019
GAN	0.811 ± 0.06	0.921 ± 0.138	0.805 ± 0.071	0.099 ± 0.169	0.13 ± 0.206
WGAN-GP	0.729 ± 0.045	0.375 ± 0.125	0.472 ± 0.072	0.199 ± 0.108	0.194 ± 0.095
FairGAN	0.81 ± 0.046	0.892 ± 0.182	0.799 ± 0.053	0.139 ± 0.244	0.157 ± 0.244
DECAF-ND	0.877 ± 0.031	0.779 ± 0.054	0.797 ± 0.013	0.126 ± 0.060	0.356 ± 0.080
DECAF-FTU	0.882 ± 0.028	0.776 ± 0.056	0.797 ± 0.005	0.0211 ± 0.011	0.291 ± 0.069
DECAF-CF	0.78 ± 0.0158	0.923 ± 0.031	0.737 ± 0.02	0.0195 ± 0.019	0.034 ± 0.039
DECAF-DP	0.76 ± 0.001	0.954 ± 0.034	0.674 ± 0.029	0.020 ± 0.019	0.024 ± 0.029
IF $X_s = \emptyset$ ($\lambda = 0.0$)	0.799 ± 0.009	0.932 ± 0.023	0.755 ± 0.0198	0.012 ± 0.009	0.083 ± 0.028
IF $X_s = \emptyset$ ($\lambda = 0.001$)	0.797 ± 0.007	0.929 ± 0.013	0.717 ± 0.039	0.036 ± 0.029	0.058 ± 0.024
IF $X_s = \emptyset$ ($\lambda = 0.002$)	0.8 ± 0.011	0.918 ± 0.012	0.721 ± 0.034	0.089 ± 0.031	0.034 ± 0.023
IF $X_s = \emptyset$ ($\lambda = 0.003$)	0.801 ± 0.010	0.923 ± 0.011	0.722 ± 0.039	0.1 ± 0.080	0.042 ± 0.022
IF $X_s = \emptyset$ ($\lambda = 0.004$)	0.799 ± 0.008	0.917 ± 0.018	0.713 ± 0.030	0.104 ± 0.051	0.02 ± 0.018
IF $X_s = \emptyset$ ($\lambda = 0.005$)	0.801 ± 0.009	0.914 ± 0.02	0.721 ± 0.0381	0.149 ± 0.076	0.030 ± 0.022
IF $X_s = \emptyset$ ($\lambda = 0.006$)	0.8 ± 0.009	0.919 ± 0.015	0.713 ± 0.030	0.143 ± 0.046	0.016 ± 0.011
IF $X_s = \emptyset$ ($\lambda = 0.007$)	0.804 ± 0.009	0.912 ± 0.020	0.711 ± 0.042	0.174 ± 0.043	0.028 ± 0.015
IF $X_s = \emptyset$ ($\lambda = 0.008$)	0.801 ± 0.009	0.918 ± 0.017	0.704 ± 0.042	0.135 ± 0.044	0.019 ± 0.012
IF $X_s = \emptyset$ ($\lambda = 0.009$)	0.804 ± 0.009	0.905 ± 0.011	0.72 ± 0.034	0.243 ± 0.069	0.033 ± 0.025
IF $X_s = \emptyset$ ($\lambda = 0.100$)	0.895 ± 0.027	0.566 ± 0.029	0.712 ± 0.025	0.063 ± 0.051	0.242 ± 0.062
IF $X_s = X$ ($\lambda = 0.0$)	0.8 ± 0.008	0.926 ± 0.019	0.744 ± 0.030	0.017 ± 0.018	0.090 ± 0.028
IF $X_s = X$ ($\lambda = 10^{-4}$)	0.812 ± 0.008	0.918 ± 0.012	0.786 ± 0.015	0.015 ± 0.015	0.099 ± 0.025
IF $X_s = X$ ($\lambda = 10^{-3}$)	0.813 ± 0.009	0.919 ± 0.016	0.776 ± 0.019	0.013 ± 0.010	0.108 ± 0.029
IF $X_s = X$ ($\lambda = 10^{-2}$)	0.867 ± 0.024	0.855 ± 0.071	0.829 ± 0.024	0.049 ± 0.019	0.247 ± 0.075
IF $X_s = X$ ($\lambda = 10^{-1}$)	0.877 ± 0.014	0.871 ± 0.031	0.842 ± 0.012	0.015 ± 0.014	0.241 ± 0.054
IF $X_s = X$ ($\lambda = 10^{0.0}$)	0.899 ± 0.016	0.825 ± 0.042	0.847 ± 0.013	0.018 ± 0.012	0.318 ± 0.058
IF $X_s = X$ ($\lambda = 10^{+1}$)	0.879 ± 0.042	0.823 ± 0.074	0.825 ± 0.022	0.040 ± 0.035	0.254 ± 0.131

The following figures show how the difference between the losses of the two decoders (in IF model) changes with batch size for different values of λ. Here we show the figure for $\lambda = 10^{-1}, 10^{-2}, 10^{-3}$.

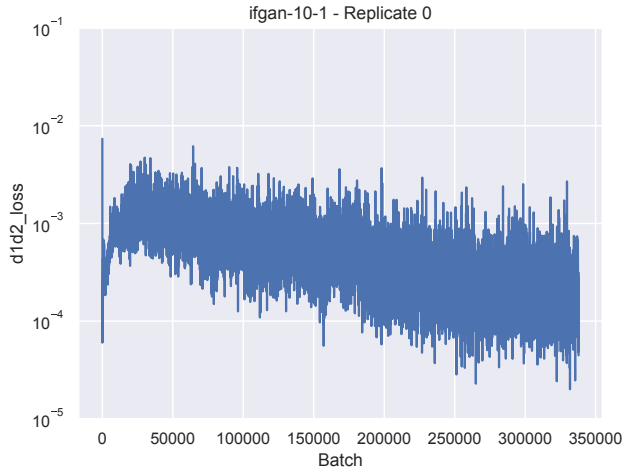

Figure 7: $\lambda = 10^{-1}$

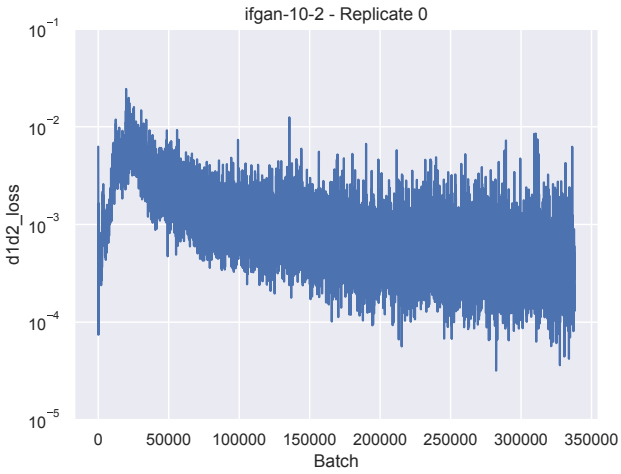

Figure 8: $\lambda = 10^{-2}$

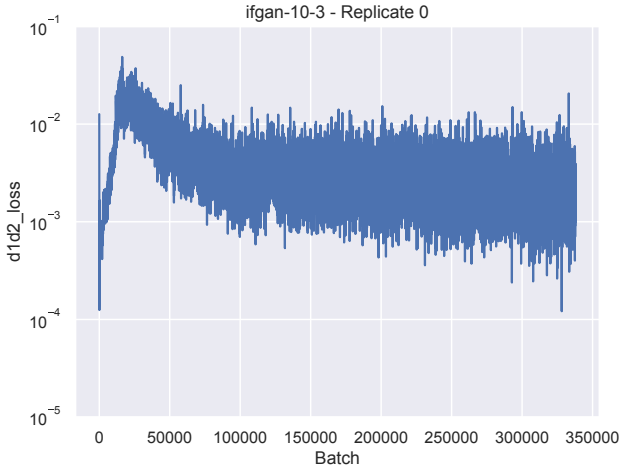

Figure 9: $\lambda = 10^{-3}$

E Extending Propositions 1 and 3 to Counterfactual Fairness

Proposition 5. Let $f : \mathcal{X} \to \{0, 1\}$ be a prediction algorithm and $\hat{Y} = f(X)$. If a distribution P' approximately satisfies counterfactual fairness, that is $\mathbb{TV}(P'(Y_{A\leftarrow 0}(U)|X = x, A = a), P'(Y_{A\leftarrow 1}(U)|X = x, A = a)) \leq \delta, \forall x, a$, and prediction algorithm $f : \mathcal{X} \to \{0, 1\}$ satisfies the following w.r.t P', $P'\{\hat{Y}_{A\leftarrow a}(U) \neq Y_{A\leftarrow a}(U)\} \leq \epsilon, \forall a$, then we have:

$$\mathbb{TV}(P'(\hat{Y}_{A\leftarrow 0}(U)|X = x, A = a)), P'(\hat{Y}_{A\leftarrow 1}(U)|X = x, A = a)) \leq 2\epsilon + \delta. \tag{30}$$

Proof. Note that $\mathbb{TV}$ satisfies triangle inequality. Therefore, we have,

$$\begin{aligned}
&\mathbb{TV}(P'(\hat{Y}_{A\leftarrow 0}(U)|X = x, A = a), P'(\hat{Y}_{A\leftarrow 1}(U)|X = x, A = a)) \\
&\leq \mathbb{TV}(P'(\hat{Y}_{A\leftarrow 0}(U)|X = x, A = a), P'(Y_{A\leftarrow 0}(U)|X = x, A = a)) \\
&+ \mathbb{TV}(P'(Y_{A\leftarrow 0}(U)|X = x, A = a), P'(Y_{A\leftarrow 1}(U)|X = x, A = a)) \\
&+ \mathbb{TV}(P'(Y_{A\leftarrow 1}(U)|X = x, A = a), P'(\hat{Y}_{A\leftarrow 1}(U)|X = x, A = a)).
\end{aligned} \tag{31}$$

Note that based on our assumption P' satisfies counterfactual fairness, and the second term above is bounded by δ . Now for the first term of RHS we have:

$$
\begin{aligned}
&\mathbb{TV}(P'(\hat{Y}_{A\leftarrow 0}(U)|X=x,A=a),P'(Y_{A\leftarrow 0}(U)|X=x,A=a))\\
&=\frac{1}{2}\sum_{y\in\{0,1\}}|P'(\hat{Y}_{A\leftarrow 0}(U)=y|X=x,A=a)-P'(Y_{A\leftarrow 0}(U)=y|X=x,A=a)|\\
&=\frac{1}{2}\sum_{y\in\{0,1\}}|P'(\hat{Y}_{A\leftarrow 0}(U)=y,Y_{A\leftarrow 0}(U)=y|X=x,A=a)\\
&+P'(\hat{Y}_{A\leftarrow 0}(U)=y,Y_{A\leftarrow 0}(U)=1-y|X=x,A=a)-P'(Y_{A\leftarrow 0}(U)=y|X=x,A=a)|\\
&=\frac{1}{2}\sum_{y\in\{0,1\}}|P'(\hat{Y}_{A\leftarrow 0}(U)=y,Y_{A\leftarrow 0}(U)=y|X=x,A=a)\\
&+P'(\hat{Y}_{A\leftarrow 0}(U)=1-y,Y_{A\leftarrow 0}(U)=y|X=x,A=a)-P'(\hat{Y}_{A\leftarrow 0}(U)=1-y,Y_{A\leftarrow 0}(U)=y|X=x,A=a)\\
&+P'(\hat{Y}_{A\leftarrow 0}(U)=y,Y_{A\leftarrow 0}(U)=1-y|X=x,A=a)-P'(Y_{A\leftarrow 0}(U)=y|X=x,A=a)|\\
&=\frac{1}{2}\sum_{y\in\{0,1\}}|P'(Y_{A\leftarrow 0}(U)=y|X=x,A=a)-P'(\hat{Y}_{A\leftarrow 0}(U)=1-y,Y_{A\leftarrow 0}(U)=y|X=x,A=a)\\
&+P'(\hat{Y}_{A\leftarrow 0}(U)=y,Y_{A\leftarrow 0}(U)=1-y|X=x,A=a)-P'(Y_{A\leftarrow 0}(U)=y|X=x,A=a)|\\
&=\frac{1}{2}\sum_{y\in\{0,1\}}|P'(\hat{Y}_{A\leftarrow 0}(U)=y,Y_{A\leftarrow 0}(U)=1-y|X=x,A=a)\\
&-P'(\hat{Y}_{A\leftarrow 0}(U)=1-y,Y_{A\leftarrow 0}(U)=y|X=x,A=a)|\\
&\leq\sum_{y\in\{0,1\}}|P'(\hat{Y}_{A\leftarrow 0}(U)=y,Y_{A\leftarrow 0}(U)=1-y|X=x,A=a)|\\
&\leq\epsilon
\end{aligned}
$$

Similarly for the third term of RHS of (31) we have:

$$\mathbb{TV}(P'(\hat{Y}_{A\leftarrow 0}(U)|X=x,A=a),P'(Y_{A\leftarrow 0}(U)|X=x,A=a))\leq\epsilon.$$

This completes the proof.

□

Proposition 6. Assume $\mathbb{TV}(P(X_{A\leftarrow a}(U)|X=x,A=a),P'(X_{A\leftarrow a}(U)|X=x,A=a)\leq\epsilon\forall x,a$.[5] If a prediction algorithm $f:\mathcal{X}\rightarrow\{0,1\}$ approximately satisfies CF w.r.t P', that is $\mathbb{TV}(P'(\hat{Y}_{A\leftarrow 0}(U)|X=x,A=a),P'(\hat{Y}_{A\leftarrow 1}(U)|X=x,A=a))\leq\delta,\forall x,a$, then we have,

$$\mathbb{TV}(P(\hat{Y}_{A\leftarrow 0}|X=x,A=a),(P(\hat{Y}_{A\leftarrow 1}|X=x,A=a))\leq 2\epsilon+\delta. \tag{32}$$

Proof. Similar to Proposition 1, using triangle inequality for $\mathbb{TV}$, we have

$$
\begin{aligned}
&\mathbb{TV}(P(\hat{Y}_{A\leftarrow 0}(U)|X=x,A=a),P(\hat{Y}_{A\leftarrow 1}(U)|X=a,A=a))\leq\\
&\mathbb{TV}(P(\hat{Y}_{A\leftarrow 0}(U)|X=x,A=a),P'(\hat{Y}_{A\leftarrow 0}(U)|X=x,A=a))+\\
&\mathbb{TV}(P'(\hat{Y}_{A\leftarrow 0}(U)|X=x,A=a),P'(\hat{Y}_{A\leftarrow 1}(U)|X=x,A=a))+\\
&\mathbb{TV}(P'(\hat{Y}_{A\leftarrow 1}(U)|X=x,A=a),P(\hat{Y}_{A\leftarrow 1}(U)|X=x,A=a)).
\end{aligned}
\tag{33}
$$

[5]Note that $P(X_{A\leftarrow a}(U)=x'|X=x,A=a)=\sum_u P(X_{A\leftarrow a}(u)=x')P(U=u|X=x,A=a)$. This condition implies that not only the factual data have similar distribution under P and P', but also the counterfactual data follow similar distribution under P and P'

The second term above is bounded by δ_1 (since $\hat{Y}$ approximately satisfies CF with respect to P').For the first term of (33), we have:

$$\begin{aligned}&\mathbb{TV}(P(\hat{Y}_{A\leftarrow 0}(U)|X=x,A=a),P'(\hat{Y}_{A\leftarrow 0}(U)|X=x,A=a))\\&\quad=\sum_{y\in\{0,1\}}\mathbb{TV}(P(f(X_{A\leftarrow 0}(U))|X=x,A=a),P'(f(X_{A\leftarrow 0}(U))|X=x,A=a))\leq\epsilon\end{aligned}\tag{34}$$

For the third term of (33), we have:

$$\begin{aligned}&\mathbb{TV}(P(\hat{Y}_{A\leftarrow 1}(U)|X=x,A=a),P'(\hat{Y}_{A\leftarrow 1}(U)|X=x,A=a))\\&\quad=\sum_{y\in\{0,1\}}\mathbb{TV}(P(f(X_{A\leftarrow 1}(U))|X=x,A=a),P'(f(X_{A\leftarrow 1}(U))|X=x,A=a))\leq\epsilon\end{aligned}\tag{35}$$

This completes the proof.

□