The state-of-the-art of Causal fairness-aware GAN-Based synthetic data generators

Report for "*Statistics for Machine Learning*" Ph.D A.y. Course 24/25

Andrea Iommi

September 2025

1 Introduction

This report aims to describe and discuss some state-of-the-art methodologies in synthetic data generation. The works presented are GAN–based causal fairness-aware approaches for generating tabular synthetic datasets. The works chosen [11, 10, 4, 1] are presented in chronological order. Starting from FairGAN [11] (2018), which was one of the first approaches to tackle discrimination in the generative methods, we explore more sophisticated and causal-grounded architectures such as CFGAN [10] (2019) and DECAF [4] (2021) to arrive at the more recent paper explored, CFSDG [1] (2022). Each paper presented outlines the limitations of the previous works.

Motivation. Synthetic data generation (SDG) denotes an essential tool for Machine Learning (ML), particularly in high-risk fields such as healthcare [7], business [2], or recruitment [3]. In many of these applications, privacy concerns and regulations make it impossible to release real datasets, creating a bottleneck for training ML architectures. Synthetic data (SD) offers a promising solution by creating a dataset that resembles the real data while protecting the privacy of individuals. Another essential challenge is related to the fairness of the data. Historical datasets often contain inherent biases, and ML models trained on such data can reflect and even exacerbate existing discrimination. A subfield of the SDG work on removing these discriminatory biases before the data is used. In other words, the goal is to produce datasets that are similar to the original data, but discrimination-free, so that any downstream model trained on them will also be fair.

The report is organised as follows: in Section 2, we provide a brief outline of the key concepts utilised by papers, in Section 3, we describe the contributions by providing an intuition behind the methods, in Section 4 we argue about the strength and weakness of the algorithms, and finally in Section 5 we wrap up the analysed papers providing some idea about future works.

2 Background

2.1 Structural Causal Models

A Structural Causal Model (SCM) [6, 9, 8] describes a data-generating process by relating random variables in cause-effect pairs. Let $\boldsymbol{X} = \{X_1, \dots, X_d\}$ be d observable random variables, defined by a set $\mathbf{F}$ of structural equations:

$$X_i := f_i(\mathbf{PA}_i, U_i) \qquad \text{for } i = 1, \dots, d \tag{1}$$

where $\mathbf{U} = \{U_1, \dots, U_d\}$ are d independent exogenous (unobserved) random variables, and $\mathbf{PA}_i \subseteq \boldsymbol{X} \setminus \{X_i\}$ are the causal parents of X_i. The equations describe the causal mechanism by which an X_i is generated from its causal parents and an exogenous variable U_i. Formally, a SCM $\mathcal{M}$ is a tuple $\mathcal{M} = \langle \mathbf{U}, P(\mathbf{U}), \boldsymbol{X}, \mathbf{F} \rangle$, where $P(\mathbf{U}) = \prod_i P(U_i)$ is the probability distribution of the exogenous variables. The parental relations in a SCM induce a *causal graph* $\mathcal{G}$, in which the nodes represent random variables and a directed edge $X_j \to X_i$ denotes a causal relation between $X_j \in \mathbf{PA}_i$ and X_i. We assume Directed Acyclic Graphs (DAGs), meaning there are no loops in $\mathcal{G}$. Then the data generation process can proceed by following a topological order of the variables given the graph. Under the Markov property assumption, the induced probability on $\boldsymbol{X}$ can then be factorized as $P(\boldsymbol{X}) = \prod_i P(X_i | \mathbf{PA}_i, U_i)$.

Atomic interventions. SCMs allow us to reason about the effect of external manipulations on the system. An *atomic intervention* consists of setting a variable X_j to a value x regardless of its natural causes, denoted by the *do-operator* $\mathrm{do}(X_j = x)$ [6]. Formally, this is modelled by replacing the structural equation for X_j with the constant assignment $X_j := x$, while leaving all other equations unchanged. The resulting modified model generates a new distribution that captures the causal effect of the intervention. In formula:

$$P_{\boldsymbol{X}}^{\hat{\mathcal{M}}} =: P_{\boldsymbol{X}}^{\mathcal{M};\mathrm{do}(X_j=x)} \tag{2}$$

Counterfactual. Beyond interventions, SCMs enable reasoning about *counterfactuals*, i.e., statements of the form: "What would Y have been if $\boldsymbol{X}$ had taken value x, given that we observed evidence $\boldsymbol{X} = \boldsymbol{x}$?". Counterfactual queries require three steps: (i) *abduction*, where we update the distribution of exogenous variables $\mathbf{U}$ given the observed evidence; (ii) *action*, where we modify the model by applying the intervention $\mathrm{do}(\boldsymbol{X} = \boldsymbol{x})$; and (iii) *prediction*, where we compute the value (or distribution) of Y in the modified model using the inferred $\mathbf{U}$. In formula [1]:

$$\mathcal{M}_{\boldsymbol{X}=\boldsymbol{x}} := \langle \mathbf{U}, P(\mathbf{U})^{\mathcal{M}|\boldsymbol{X}=\boldsymbol{x}}, \boldsymbol{X}, \mathbf{F} \rangle \tag{3}$$

where $P(\mathbf{U})^{\mathcal{M}|\boldsymbol{X}=\boldsymbol{x}} := P(U|\boldsymbol{X} = \boldsymbol{x})$. The new set of noise variables is no longer jointly independent.

2.2 Notations and Definitions

Let $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}, \mathcal{A}\} \sim P_{data}$ (simply P) be a dataset where $\mathcal{X}$ is the unprotected features, $\mathcal{Y}$ is the target, and $\mathcal{A}$ is the protected feature drawn from a data distribution P. Denote $\mathcal{Q}$ as a

[1] I have slightly change the notation presented in [6], to be consistent with the whole report.

concatenation of the unprotected features $\mathcal{X}$ and the protected feature $\mathcal{A}$, i.e., $\mathcal{Q} = \mathcal{X} \cup \mathcal{A}$. The "Fairness" generative models aim to approximate the distribution P with P' such that a synthetic dataset $\mathcal{D}' \sim P'$ is as close as possible to $\mathcal{D}$ and discrimination-free. This property ensures that ML models trained on that data are fair [4, 11]. For the remainder of the paper, we use the following notation: we refer to the training set and data distribution as $\mathcal{D}$ and P, respectively. Instead, we denote the distribution approximated by the model and the synthetic dataset as P' and $\mathcal{D}'$.

Finally, we refer to $\boldsymbol{X}$ as a random variable that encompasses the unprotected features, A the protected attribute, Y the target (often referred to as "decision") and $\hat{Y} = f : \mathcal{Q} \rightarrow \mathcal{Y}$ as a downstream task predictor.

Definition 1 (Fairness Through Unawareness (FTU))**.** *A predictor $\hat{Y}$ is fair iff protected attributes A are not explicitly used in the prediction.*

In other words, when $A \notin \boldsymbol{Q} \implies \boldsymbol{Q} = \boldsymbol{X}$ only.

Definition 2 (Statistical parity (SP) or Demographic parity(DP))**.** *A predictor $\hat{Y}$ satisfies the DP if $P(\hat{Y}|A = a) = P(\hat{Y}|A = a')$, i.e., $A \perp\!\!\!\perp \hat{Y} \quad \forall a, a' \in \mathcal{A}$.*

Definition 2 is a group-level fairness notion. It requires that the probability of receiving a positive (or negative) outcome is the same across all protected groups.

Definition 3 (Conditional Fairness (CF))**.** *A predictor $\hat{Y}$ is conditional fair iff $A \perp\!\!\!\perp \hat{Y}|\boldsymbol{R}$ where $\boldsymbol{R} \subset \boldsymbol{X}$, i.e. $\forall r, a, a' : P(\hat{Y}|\boldsymbol{R=r}, A = a) = P(\hat{Y}|\boldsymbol{R=r}, A = a')$*

The Definition 3 is a generalisation of FTU 1 and DP 2, by setting $\boldsymbol{R} = \boldsymbol{X}$ and $\boldsymbol{R} = \emptyset$, respectively.

Definition 4 (Counterfactual Fairness (COF))**.** *A predictor $\hat{Y}$ satisfies counterfactual fairness if for any context $A = a$ and $\boldsymbol{X} = \boldsymbol{x}$, $P(\hat{Y}_{A\leftarrow a} = y|\boldsymbol{X} = \boldsymbol{x}, A = a) = P(\hat{Y}_{A\leftarrow a'} = y|\boldsymbol{X} = \boldsymbol{x}, A = a)$ holds for all value of y and $a' \in \mathcal{A}$.*

Definition 4 is an individual-level fairness notion. It requires that for any given individual, the outcome would have been the same even if their protected attribute had been different, assuming all other background factors remained constant.

2.3 Generative Adversarial Networks

Generative Adversarial Networks (GANs) [5] are generative models designed to learn a data distribution and produce new samples that resemble those drawn from the true distribution. A GAN consists of two neural networks that are trained simultaneously: the *generator* and the *discriminator*. To simplify the notation, ignore the distinction between protected, unprotected attributes and target for the rest of this section, i.e., assume that $\mathcal{X} = \mathcal{X} \cup \mathcal{A} \cup \mathcal{Y}$. The generator $G : \mathcal{Z} \rightarrow \mathcal{X}$ maps a random noise vector $\boldsymbol{z} \sim P(\boldsymbol{Z})$ into the data space $\mathcal{X}$. Its goal is to produce synthetic samples $\boldsymbol{x'} \sim G(\boldsymbol{z})$ that are indistinguishable from real data. The discriminator $D : \mathcal{X} \rightarrow [0, 1]$ is a binary classifier that receives as input either a real data sample $\boldsymbol{x} \sim P(\boldsymbol{X})$ or a synthetic sample $\boldsymbol{x'} \sim G(\boldsymbol{z})$. It outputs a probability indicating how likely the input is to belong to the true data distribution rather than being generated. Training proceeds as a two-player minimax game. The discriminator aims to maximise the probability of correctly distinguishing real from fake samples, while the generator aims to minimise the discriminator's ability to detect fakes. Formally, the objective function is:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim P(\boldsymbol{X})} \big[\log D(\boldsymbol{x}) \big] + \mathbb{E}_{\boldsymbol{z} \sim P(\boldsymbol{Z})} \big[\log \big(1 - D(G(\boldsymbol{z})) \big) \big] \tag{4}$$

CausalGAN. [4, 10, 1] present a variation of vanilla GAN (Section 2.3) in which the generation process is distinct for each feature and follows a topological order provided by a causal graph $\mathcal{G}$. Let us rethink $\mathcal{D}$ (defined in Section 2.2) as a set of n observation $\mathcal{D} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$, where $\boldsymbol{x} \in \mathbb{R}^d$, d is the number of features and n the observations.

A CausalGAN is constituted by d structural equation f_i (Equation 1) modelled by a separate conditional GAN $G_i : \mathbb{R}^{|Pa(X_i)+1|} \rightarrow \mathbb{R}$. The generator takes as input the parents of X_i and a random noise $\boldsymbol{z}$. The features are generated sequentially following the topological ordering of the underlying causal DAG. Subsequently, the synthetic sample $\boldsymbol{x}'$ is passed to a discriminator D, which is trained to distinguish the generated samples from original samples as in Equation 4. In other words, in the vanilla GAN, the generator is a single neural network; on the other hand, in a CausalGAN, the generator is a set of CGANs, one for each feature, which are connected by the causal relationship imposed by the graph $\mathcal{G}$.

3 Contributions

3.1 FairGAN

Fairness-aware Generative Adversarial Network (FairGAN) [11] is one of the first papers that exploits a GAN–based approach to tackle the discrimination in tabular SDG. The intuition behind the model is to extend the vanilla GAN architecture (Section 2.3) by imposing an additional fairness constraint. The constraint motivates the model to make both the unprotected features and the target independent of the protected attribute.

Let us assume a dataset $\mathcal{D}$ (as defined in Section 2.2) where both target $\mathcal{Y}$ and the protected feature $\mathcal{A}$ are binary. The architecture consists of one generator, G, and two discriminators, D^1 and D^2. The generator generates fake samples $\mathcal{D}' = \{(\boldsymbol{x}', y')\}$ conditioned on the protected attribute $a \sim P(A)$, D^1 aims to ensure that generated data $\{(\boldsymbol{x}', a, y')\}$ is close to the real data $\{(\boldsymbol{x}, a, y)\}$ as possible, while the other discriminator D^2 aims to ensure there are no correlation between $\boldsymbol{X}'$ and S and no correlation between Y' and S. a is a sample from the marginal distribution of the dataset $\mathcal{D}$.

In other words, FairGAN generates the unprotected attributes $\boldsymbol{x}'$ and decision y' given the protected attribute a, and achieves $\boldsymbol{X}' \perp\!\!\!\perp S$ and $Y' \perp\!\!\!\perp S$. Therefore, the generated data can meet the requirements in terms of Demographic parity (Section 2) and FTU (Section 1). FairGAN tries to optimise:

$$\min_G \max_{D^1, D^2} J(G, D^1, D^2) = J_1(G, D^1) + \lambda J_2(G, D^2) \tag{5}$$

where:

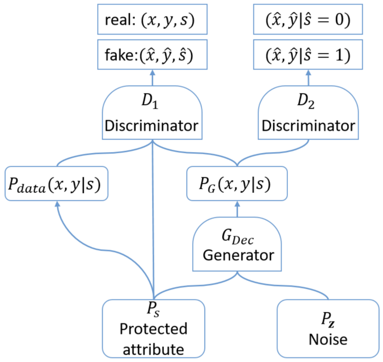

Figure 1: The Structure of FairGAN

$$
\begin{aligned}
J_1(G, D^1) &= \mathbb{E}_{a\sim P(A),(\boldsymbol{X},Y)\sim P(\boldsymbol{X},Y|A)}[\log D^1(\boldsymbol{x}, y, a)] \\
&\quad + \mathbb{E}_{a\sim P(A),(\boldsymbol{x}',y')\sim P_G(\boldsymbol{X}',Y'|A)}[1 - \log D^1(\boldsymbol{x}', y', a)] \\
J_2(G, D^2) &= \mathbb{E}_{(\boldsymbol{x}',y')\sim P_G(\boldsymbol{X}',Y'|A=1)}[\log D^2(\boldsymbol{x}', y')] \\
&\quad + \mathbb{E}_{(\boldsymbol{x}',y')\sim P_G(\boldsymbol{X}',Y'|A=0)}[1 - \log D^2((\boldsymbol{x}', y'))]
\end{aligned} \tag{6}
$$

The parameter λ balances the satisfaction of the fairness notion. Figure 1 shows the model's architecture. In particular, the discriminator D^2 tries to distinguish the data distribution derived from the different protected value conditions. The generator should achieve the situation in which $P_G(\boldsymbol{X}', Y'|A = 0) \approx P_G(\boldsymbol{X}', Y'|A = 1)$.

3.2 Causal fairness-aware GAN

Causal fairness-aware GAN (CFGAN) [10] represents GAN–based SDG, which ensures fairness in the synthetic dataset by using the fairness notion (see Section 2.2) integrated in the loss functions. The architecture is similar to FairGAN (Section 3.1), but also exploits causal reasoning (Section 2.1). In detail, CFGAN relies on the CausalGAN (Section 2.3) but extends the architecture by including an additional generator and discriminator. Hence, the model consists of two generators, G^1 and G^2, and two discriminators, D^1 and D^2. One discriminator is used to optimise the data utility, and another to enforce the fairness constraint. We specify that each generator G is composed of a set of sub-generators $G^i = \{G^i_0, \ldots, G^i_d\}$. In addition, the framework assumes a dataset $\mathcal{D}$ and a causal graph $\mathcal{G}$ that faithfully respects the dataset. The key concept behind the proposed model is to minimise the "change in distribution" caused by an intervention or counterfactual in the causal structure. Intuitively, if an intervention on the protected attribute does not change the data distribution, we achieve fairness in the data generation process. To introduce the algorithm, we need to describe some useful notions of fairness: total effect, path-specific effect, and counterfactual effect.

The *total effect* measures the causal effect of $\boldsymbol{X}$ on Y where the intervention is transferred along all causal paths (i.e., directed paths) from $\boldsymbol{X}$ to Y. For the rest of this paper, we refer $P(Y|\text{do}(\boldsymbol{X} = \boldsymbol{x}))$ as $P(Y_{\boldsymbol{x}})$.

Definition 5 (Total effect)**.** *The total effect of the value change of* $\boldsymbol{X}$ *from* $\boldsymbol{x}_1$ *to* $\boldsymbol{x}_2$ *on* Y *is given by* $TE(\boldsymbol{x}_2, \boldsymbol{x}_1) = P(Y_{\boldsymbol{x}_2}) - P(Y_{\boldsymbol{x}_1})$*. When* $TE(\boldsymbol{x}_2, \boldsymbol{x}_1) = 0$*, it means no changes.*

The path-specific effect measures the causal effect of $\boldsymbol{X}$ on Y where the intervention is transferred only along a subset of causal paths from $\boldsymbol{X}$ to Y, which is also referred to as the π-specific effect, denoting the subset of causal paths as π.

Definition 6 (Path-specific effect)**.** *Given a path set* π*, the* π*-specific effect of the value change of* $\boldsymbol{X}$ *from* $\boldsymbol{x}_1$ *to* $\boldsymbol{x}_2$ *on* Y *(with reference* $\boldsymbol{x}_1$*) is given by* $SE_\pi(\boldsymbol{x}_2, \boldsymbol{x}_1) = P(Y_{\boldsymbol{x}_2}|\pi) - P(Y_{\boldsymbol{x}_1}|\pi)$*, where* $P(Y_{\boldsymbol{x}}|\pi)$*. When* $SE_\pi(\boldsymbol{x}_2, \boldsymbol{x}_1) = 0$*, it means no specific effect.*

In both the total effect and path-specific effect, the intervention is applied to the entire population. The counterfactual effect measures the causal effect while the intervention is performed, conditioning on only certain individuals or groups specified by a subset of observed variables $\boldsymbol{O} = \boldsymbol{o}$.

Definition 7 (Counterfactual effect)**.** *Given a context* $\boldsymbol{O} = \boldsymbol{o}$*, the counterfactual effect of the value change of* $\boldsymbol{X}$ *from* $\boldsymbol{x}_1$ *to* $\boldsymbol{x}_2$ *on* Y *is given by* $CE(\boldsymbol{x}_2, \boldsymbol{x}_1|\boldsymbol{o}) = P(Y_{\boldsymbol{x}_2}|\boldsymbol{o}) - P(Y_{\boldsymbol{x}_1}|\boldsymbol{o})$*.*

Architectures

The paper proposes three kinds of architecture based on the definition of fairness to adopt. We recall that all architectures adopt two generators denoted as G^1 and G^2 that share the same weights for each subnetwork $G_i \quad \forall_i \in \boldsymbol{X}$.

CFGAN based on Total Effect *CFGAN based on Total Effect* is the simplest architecture and tries to minimise the change in the Total effect notion (Definition 5). Similarly to CausalGAN (Section 2.3), G^1 aims to approximate the true distribution $P(\boldsymbol{X})$. On the other hand, in G^2, an intervention of s is applied, in formula $P(Y|\mathrm{do}(S = s))$, and the aim is to minimise the Total effect. We identify $P(Y_{s+})$ when $s = 1$ and $P(Y_{s-})$ when $s = 0$. Concerning the discriminators, D^1 is designed to distinguish between the real data $(x, y, s) \sim P(\boldsymbol{X}, Y, S)$ and the generated data $(\boldsymbol{x}', y', s') \sim P_{G^1}(\boldsymbol{X}', Y', S')$. D^2 is employed to distinguish between the two interventional distributions: $y'_{s+} \sim P_{G^2}(Y'_{s+})$ and $y'_{s-} \sim P_{G^2}(Y'_{s-})$. Thus, the overall cost function is defined as in Equation 5 where:

$$
\begin{aligned}
J_1(G^1, D^1) &= \mathbb{E}_{(\boldsymbol{x},y,s)\sim P(\boldsymbol{X},Y,S)}[\log D^1(\boldsymbol{x}, y, s)] + \mathbb{E}_{(\boldsymbol{x}',y',s')\sim P_{G^1}(\boldsymbol{X}',Y',S')}[1 - \log D^1(\boldsymbol{x}', y', s')] \\
J_2(G^2, D^2) &= \mathbb{E}_{y'_{s+}\sim P_{G^2}(Y'_{s+})}[\log D^2(y'_{s+})] + \mathbb{E}_{y'_{s-}\sim P_{G^2}(Y'_{s-})}[1 - \log D^2(y'_{s-})]
\end{aligned}
\tag{7}
$$

Optimising Equation 7 means obtaining a situation in which $P(Y'_{s+}) \sim P(Y'_{s-})$ that corresponds to no total effect change (Definition 5). In other words, we trained a GAN to generate the same data distribution independently of the sensitive attribute.

CFGAN based on Indirect Discrimination To mitigate the indirect discrimination (Definition 6), the authors change G^2. They consider two types of value settings for the sub-network G^2: *(i)* the reference setting and *(ii)* the interventional setting. In the reference setting $G^2_s = 0$ always, for interventional setting $G^2_s = 1$ if $s = s^+$ and $G^2_s = 0$ if $s = s^-$. Thus, each sub-network

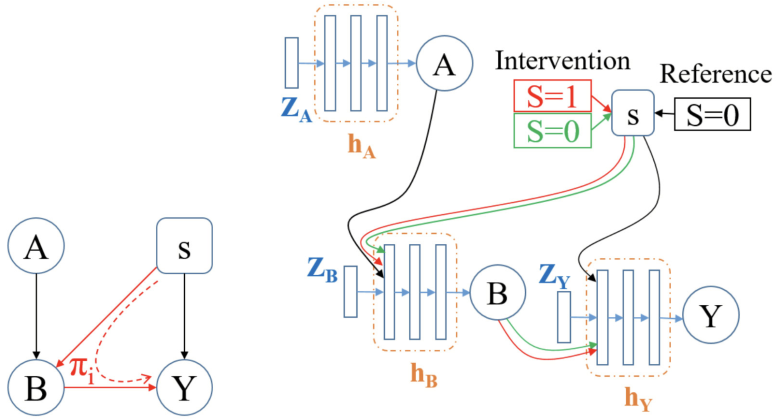

Figure 2: An example of the generator G^2 for CFGAN based on indirect discrimination. S is set to 1 or 0 and the transmission is set only along $\pi = \{S \to B \to Y\}$ to sample from the interventional distributions $P_{G^2}(A_{s^+}|\pi, B_{s^+}|\pi, Y_{s^+}|\pi)$ (red) and $P_{G^2}(A_{s^-}|\pi, B_{s^-}|\pi, Y_{s^-}|\pi)$ (green) respectively. S is set to be 0 for the reference setting.

may output two types of sample values according to the value setting of G^2. For a sub-network, if its corresponding node is not on any path in π, it always takes reference values as input and outputs reference values. For any other sub-network that is on at least one path in π, it may take both types of values as input and output both. Figure 2 shows an example. To achieve no indirect discrimination, the loss changes as follows:

$$J_2(G^2, D^2) = \mathbb{E}_{y'_{s+|\pi} \sim P_{G^2}(Y'_{s+|\pi})}[\log D^2(y'_{s+|\pi})] + \mathbb{E}_{y'_{s-|\pi} \sim P_{G^2}(Y'_{s-|\pi})}[1 - \log D^2(y'_{s-|\pi})] \tag{8}$$

CFGAN for Counterfactual Fairness In counterfactual fairness (Definition 7), the intervention is performed conditioning on a subset of variables $\boldsymbol{O} = \boldsymbol{o}$. What changes concerning the previous cases are the conditioning. Previously, we conditioned only on S (Total Effect and Indirect Discrimination); now we condition also on the subset features $\boldsymbol{O}$ (e.g., $\boldsymbol{O} = \{race, native\_country\}$). Specifically, first we generate samples with G^1 (the generator that tends to approximate the true distribution, as J_1 in Equation 7), then we take noise vectors $\boldsymbol{z}$ that generate samples in which $\boldsymbol{O} = \boldsymbol{o}$ (i.e., samples in which certain features have specific values). Finally, we utilise G^2, with the noise vector $\boldsymbol{z}$, to generate samples from the interventional distribution on S and $\boldsymbol{O}$ denoted by $P_{G^2}(\boldsymbol{X}'_s, Y'_s|\boldsymbol{o})$. The discriminator D^2 is revised to distinguish from $y'_{s+|o} \sim P(Y'_{s+|o})$ and $y'_{s-|o} \sim P(Y'_{s-|o})$. Hence, J_2 loss function becomes:

$$J_2(G^2, D^2) = \mathbb{E}_{y'_{s+|\boldsymbol{o}} \sim P_{G^2}(Y'_{s+|\boldsymbol{o}})}[\log D^2(y'_{s+|\boldsymbol{o}})] + \mathbb{E}_{y'_{s-|\boldsymbol{o}} \sim P_{G^2}(Y'_{s-|\boldsymbol{o}})}[1 - \log D^2(y'_{s-|\boldsymbol{o}})] \tag{9}$$

3.3 DECAF

DEbiasing CAusal Fairness (DECAF) [4] denotes a generic approach that assumes a given causal graph $\mathcal{G}$ and learns the structural equations through conditional GANs as done for CausalGAN

(Section 2.3). Before generating the data, we intervene in the derived SCM by removing dependencies that lead to unfairness in a downstream model (these can be specific to the fairness metric considered). Then we sample data by applying the (GAN—based) structural equations that remain after these dependencies are removed. To understand the logic behind DECAF, we have to introduce some definitions.

Definition 8 (Distributional fairness)**.** *A probability distribution $P'(\boldsymbol{X})$ is $(\mathcal{I}(A,Y),P)-fair$, iff the optimal predictor $\hat{Y}$ trained on $P'(\boldsymbol{X})$ satisfies $\mathcal{I}(A,Y)$, when evaluated on $P(\boldsymbol{X})$.*

In other words, when we train a predictor on $(\mathcal{I}(A,Y),P)-fair$ distribution $P'(\boldsymbol{X})$, we can only reach maximum performance if our model is fair. Note that the predictor is evaluated on the original distribution data. From a graphical point of view, let $\mathcal{G}$ be an assumed graph faithfully with respect to the distribution $P(\boldsymbol{X})$, and $\mathcal{G}' \subset \mathcal{G}$ (where some edges are removed) faithfully with respect to P'. The graphical condition is:

Definition 9 (Graphical condition)**.** *If for all $B \in \partial_{\mathcal{G}'}Y, A \perp\!\!\!\perp_{\mathcal{G}} B|R$ then the distribution $P'(\boldsymbol{X})$ is CF (Definition 3) fair w.r.t $P(\boldsymbol{X})$ given explanatory factor $\boldsymbol{R}$.*

Where $\perp\!\!\!\perp_{\mathcal{G}}$ denotes d-separation in $\mathcal{G}$ and $\partial_{\mathcal{G}'}Y$ denotes the Markov boundary of Y in graph $\mathcal{G}'$. This led to the following corollary:

Corollary 1 (CF debiasing)**.** *Any distribution $P'(\boldsymbol{X})$ with a graph $\mathcal{G}'$ can be made CF fair w.r.t $P(\boldsymbol{X})$ and explanatory features $\boldsymbol{R}$ by removing from $\mathcal{G}'$ edges $E = \{(B \rightarrow Y)$ and $(Y \rightarrow B) : \forall B \in \partial_{\mathcal{G}'}Y$ with $B \perp\!\!\!\perp_{\mathcal{G}} A|\boldsymbol{R}\}$.*

For FTU (Definition 1) (i.e. $\boldsymbol{R} = \boldsymbol{X} \setminus A$) and DP (Definition 2) (i.e. $\boldsymbol{R} = \varnothing$), Corollary 1 simplifies to:

Corollary 2 (FTU debiasing)**.** *Any distribution $P'(\boldsymbol{X})$ with graph $\mathcal{G}'$ can be made FTU (Definition 1) fair w.r.t. any distribution $P(\boldsymbol{X})$ by removing, if present, i) the edge between A and Y and ii) the edge $A \rightarrow C$ or $Y \rightarrow C$ for all shared children C.*

Corollary 3 (DP debiasing)**.** *Any distribution $P'(\boldsymbol{X})$ with graph $\mathcal{G}'$ can be made DP (Definition 2) fair w.r.t. $P(\boldsymbol{X})$ by removing, if present, the edge between B and Y for any $B \in \partial_{\mathcal{G}'}Y$ with $B \not\perp\!\!\!\perp_{\mathcal{G}} A$.*

DECAF has two distinct phases: the training and the inference. During the training stage, it learns the causal conditionals observed in the data through a causally informed GAN. At the generation (inference) stage, it intervenes on the learned conditionals, in such a way that the generator creates fair data.

Training. Each structural equation f_i (Equation 1) is modelled by a separate conditional GAN $G_i : \mathbb{R}^{|Pa(X_i)+1|} \rightarrow \mathbb{R}$ as done in CausalGAN (Section 2.3). The generator takes as input the parents of $\boldsymbol{X}_i$ and a random noise $\boldsymbol{z}$. Hence, the features are generated sequentially following the topological ordering of the underlying causal DAG. Subsequently, the synthetic sample $\boldsymbol{x'}$ is passed to a discriminator $D : \mathbb{R}^d \rightarrow [0,1]$, which is trained to distinguish the generated samples from original samples (see Equation 4).

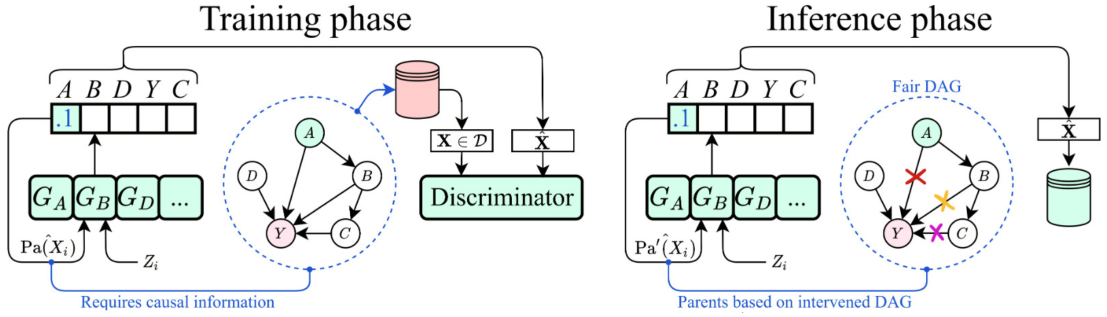

Figure 3: DECAf architecture

Inference. The training phase yields conditional generators $\{G_i\}_{i=1}^{d}$ which can be sequentially applied to generate data with the same output distribution as the original data. The key aspect of DECAF is the possibility to model the graph, by removing edges, to satisfy certain fairness notions (introduced in Corollary 2 and 3). This avoids the characteristics (such as causal relations) that we do not want to propagate in the SD. Removing an edge means applying a *do-operation* on the conditional distribution. For example, suppose we only want to remove $(i \to j)$: X_i is generated normally, but $\boldsymbol{X}_j$ is generated using the modified SEM, i.e., fixing $\boldsymbol{X}_j = \alpha$ or sampling it from the empirical marginal distribution.

3.4 CFSDG

Counterfactual Fairness in Synthetic Data Generation (CFSDG) [1] represents the most recent paper analysed. As in previous works, assume a causal graph $\mathcal{G}$ and a dataset $\mathcal{D}$ that comprise $\boldsymbol{X}$, A, Y the unobservable variable U, as in Figure 4. The intuition behind the architecture is to force the model to produce the same samples for all individuals in A through counterfactual reasoning (see Definition 4). To achieve this, the authors revisited the classical GAN architecture (Section 2.3) as follows.

The model consists of two generators, G^1 and G^2, and one discriminator D. Unlike previous works, generators are classical neural networks. The G^1 and G^2 share the same weight and take the same noise input vector $\boldsymbol{z}$ (the authors refer to the noise vector $\boldsymbol{z}$ as the unobservable variable U from the causal reasoning point of view). The generator G^1 takes as input the sensitive attribute a drawn from the marginal distribution, i.e., $a \sim P(A)$; on the other hand, the generator G^2 takes $\neg a$. The discriminator D aims to distinguish between the real and fake, as designated for D^1 in CFGAN (Section 3.2). Moreover, an additional loss is considered. The counterfactual loss ensures that the Y's generated by G^1 are as close as possible to those generated by G^2. In formula:

$$\begin{aligned} J_1 &= \mathbb{E}_{(x,a,y)\sim P(\boldsymbol{X},A,Y)} \log[D(x,a,y)] + \mathbb{E}_{u\sim P(U),a\sim P(A))}[\log(1 - D(G(u,a)))] \\ J_2 &= -\mathbb{E}_{u\sim P(U),a\sim P(A)}[\log(1 - D(G(u,a))) - \lambda(G_Y(u,a) - G_Y(u,\neg a))^2] \end{aligned} \tag{10}$$

The parameter λ balances the satisfaction of the fairness notion.

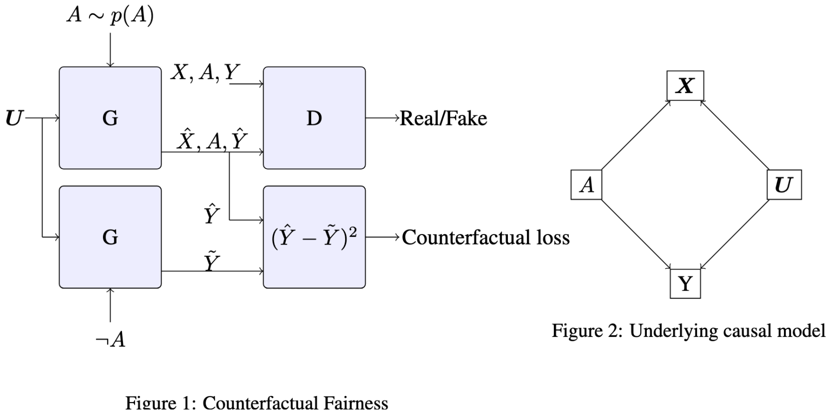

Figure 2: Underlying causal model

Figure 1: Counterfactual Fairness

Figure 4: Architecture of CFSDG

4 Discussion

Table 4 shows the main difference among the generative models. FairGAN (Section 3.1) aims to create generative models that produce datasets satisfying Demographic Parity (Definition 2). They discussed that a naive solution, in which the model is trained only with $\boldsymbol{X}$ and Y; and A is randomly sampled, is not a good solution, since $\boldsymbol{X}$ and Y may inherit discrimination and thus, A may be predictable given the other features. However, subsequent work (Sections 3.2, 3.3, 3.4) criticised FairGAN since the architecture is limited to satisfy only the DP notion in the ynthetic dataset, and they argue how the causal reasoning can be integrated in the training process to provide stronger fairness guarantees. Nevertheless, methods such as CFGAN (Section 3.2) and DECAF (Section 3.3) require having causal knowledge about data sources.

With CFGAN, the idea was born to introduce the *do-operator* to minimise the difference in distribution among the protected values using interventions and counterfactuals (see Section 2.1). Experimental results show that CFGAN w.r.t. FairGAN perform worse in Total effect (Definition 5) but better in Path-specific effect (Definition 6) [10][Table 1].

Although good results were achieved by CFGAN, DECAF pointed out a relevant limitation: the lack of treatment for multiple protected attributes. The DECAF's goal is not limited to achieving statistical constraints in the data, but also claims that the predictors trained on such data are fair as well. Even if its goal is slightly different from the previous ones, it proposes an architecture that not only allows multiple protected attributes but also manages different notions of fairness and "guarantees" fair predictors. However, in the article, there is no guarantee that P' and P are even remotely close.

For example, consider a dataset where we have $A \to X \to Y$. Assume that $A := Ber(0.5)$, $X := m \times \mathbb{1}\{a = 0\}$ and $Y := \mathcal{N}(X, 1)$. Now, for satisfying DP (Definition 3), 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. Increasing m, we can have a large KL-distance between P and P'.

The last paper, CFSDG (Section 3.4), proposes to achieve fairness to counterfactual loss. The main difference with respect to the rest of the papers lies in the intended effect of the generated data on machine learning models trained on it. The CFSDG aims to create a synthetic dataset where an accurate predictor is also a fair one. The goal is to align the incentives of the data user so that maximising accuracy naturally leads to a fair classifier. It does not force every possible classifier to be fair. DECAF and FairGAN's goals are significantly stronger; they aim to create a dataset such that every predictor trained on it will satisfy the DP. This is a more stringent condition on the data itself. Because, regardless of the data, an end user can always create an unfair predictor (e.g., by only accepting men, regardless of the other features). This is in contrast with CFSDG's definition, which only expects an accurate predictor to be fair.

Framework	**FairGAN** [11]	**CFGAN** [10]	**DECAF** [4]	**CFSDG** [1]
Only binary A	✔	✔	✖	✔
Causal Notion of f.	✖	✔	✔	✔
Require $\mathcal{G}$	✖	✔	✔	✖
Interventional f.	✖	✔	✖	✖
Fairness in $\mathcal{L}$	✔	✔	✖	✔
Counterfactual f.	✖	✖	✖	✔

Table 1: The table shows the main difference among the algorithms.

5 Conclusion

This report examined some of the most relevant works in fairness-aware SDG methods. Starting from FairGAN, which introduced fairness constraints to address DP, we examined further approaches, such as CFGAN, which incorporate causal models and counterfactual reasoning to provide stronger fairness guarantees. In contrast, DECAF formalised fairness through graphical conditions, enabling flexible debiasing strategies. Finally, CFSDG shifted the focus toward counterfactual reasoning to align fairness with predictive accuracy. Despite several contributions, some challenges remain. Most approaches assume access to causal graphs, which is often unrealistic to have or discover through Causal discovery.

All the proposed architectures yield excellent results in both data utility and fairness; however, they operate as black boxes. CausalGAN models provide insight into the generative process, since the underlying CF is understandable to a domain expert. Nevertheless, sub-generators, implemented as GANs, remain directly uninterpretable. Future research may explore the use of interpretable generators to replace GANs, thereby achieving full comprehensibility of the data generation process.

References

[1] Mahed Abroshan, Mohammad Mahdi Khalili, and Andrew Elliott. "Counterfactual fairness in synthetic data generation". In: *NeurIPS Workshop on Synthetic Data for Empowering ML Research*. 2022.

[2] Samuel A. Assefa et al. "Generating synthetic data in finance: opportunities, challenges and pitfalls". In: *ICAIF*. ACM, 2020, 44:1–44:8.

[3] Andrea Beretta et al. “Requirements of eXplainable AI in Algorithmic Hiring”. In: *AIMMES*. Vol. 3744. CEUR Workshop Proceedings. CEUR-WS.org, 2024.

[4] Boris van Breugel et al. “DECAF: Generating Fair Synthetic Data Using Causally-Aware Generative Networks”. In: *NeurIPS*. 2021, pp. 22221–22233.

[5] Ian J. Goodfellow et al. “Generative Adversarial Networks”. In: *CoRR* abs/1406.2661 (2014). arXiv: `1406.2661`. URL: `http://arxiv.org/abs/1406.2661`.

[6] Feng Li. “A Forecaster’s Review of Judea Pearl’s Causality: Models, Reasoning and Inference, Second Edition, 2009”. In: *CoRR* abs/2308.05451 (2023).

[7] Hajra Murtaza et al. “Synthetic data generation: State of the art in health care domain”. In: *Comput. Sci. Rev.* 48 (2023), p. 100546.

[8] Ana Rita Nogueira et al. “Methods and tools for causal discovery and causal inference”. In: *WIREs Data Mining Knowl. Discov.* 12.2 (2022).

[9] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

[10] Depeng Xu et al. “Achieving Causal Fairness through Generative Adversarial Networks”. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*. Ed. by Sarit Kraus. ijcai.org, 2019, pp. 1452–1458. DOI: `10.24963/IJCAI.2019/201`. URL: `https://doi.org/10.24963/ijcai.2019/201`.

[11] Depeng Xu et al. “FairGAN: Fairness-aware Generative Adversarial Networks”. In: *IEEE BigData*. IEEE, 2018, pp. 570–575.