Machine Learning (2023) 112:741–763
https://doi.org/10.1007/s10994-022-06206-8

Adversarial learning for counterfactual fairness

Vincent Grari[1,2] · **Sylvain Lamprier**[1] · **Marcin Detyniecki**[2,3]

Received: 20 November 2021 / Revised: 14 April 2022 / Accepted: 26 May 2022 /
Published online: 3 August 2022

Abstract
In recent years, fairness has become an important topic in the machine learning research community. In particular, counterfactual fairness aims at building prediction models which ensure fairness at the most individual level. Rather than globally considering equity over the entire population, the idea is to imagine what any individual would look like with a variation of a given attribute of interest, such as a different gender or race for instance. Existing approaches rely on Variational Auto-encoding of individuals, using Maximum Mean Discrepancy (MMD) penalization to limit the statistical dependence of inferred representations with their corresponding sensitive attributes. This enables the simulation of counterfactual samples used for training the target fair model, the goal being to produce similar outcomes for every alternate version of any individual. In this work, we propose to rely on an adversarial neural learning approach, that enables more powerful inference than with MMD penalties, and is particularly better fitted for the continuous setting, where values of sensitive attributes cannot be exhaustively enumerated. Experiments show significant improvements in term of counterfactual fairness for both the discrete and the continuous settings.

Keywords Counterfactual fairness · Adversarial neural network · Causal inference

Editors: Dana Drachsler Cohen, Javier Garcia, Mohammad Ghavamzadeh, Marek Petrik, Philip S. Thomas.

✉ Vincent Grari
vincent.grari@isir.upmc.fr

Sylvain Lamprier
sylvain.lamprier@isir.upmc.fr

Marcin Detyniecki
marcin.detyniecki@axa.com

[1] Sorbonne Université, CNRS, ISIR, F-75005, Paris, France

[2] AXA, Paris, France

[3] Polish Academy of Science, IBS PAN, Warsaw, Poland

1 Introduction

Fair machine learning aims at producing predictive models that do not induce any prejudice or favoritism toward an individual or a group based on a set of sensitive characteristics. As of now, a large majority of works in the field focused on group fairness, that ensures a form of conditional independence between outcomes of the models $\hat{Y}$ and any sensitive attribute A. However, group fairness may induce dramatic consequences for some individuals. For example, a person may be refused a position only because she belongs to a privileged group, regardless of her merit within the group.

Recently, Counterfactual fairness (Kusner et al., 2017) proposed to assess fairness at the individual level, by leveraging causal inference to ensure that some sensitive attributes are not the cause of a prediction change. It argues to lead to a more intuitive, powerful, and less error-prone way of reasoning about fairness (Chiappa, 2019). The idea is to imagine what any individual would look like with a variation of a given attribute of interest, such as a different gender or race for instances, in order to ensure similar outcomes for every alternate version of the same individual. While plenty of methods have been proposed recently to tackle this challenge for discrete variables, to the best of our knowledge no approach address the continuous case. The existing approches may not hold when, for instance, the sensitive attribute is the age or the weight of an individual. As discussed in Sect. 2.2, discretizing sensitive attributes is not an option in most of cases. Moreover, existing approaches present some limitations for counterfactual inference even in the discrete case (see end of Sect. 2.2).

The main contributions of this paper are:

- We propose a novel adversarial learning approach to overcome these limitations for counterfactual inference;
- Based on this, we define an approach for counterfactual fairness tolerant to continuous features, notably via a dynamic sampling method that focuses on individualized hard locations of the sensitive space;

Section 2 first gives details for counterfactual fairness, which we believe are essential for a good understanding of our contributions. Section 3 details our approach in two main steps. Section 4 evaluates performances for both the discrete and the continuous settings.

2 Background

Recently, there has been a dramatic rise of interest for fair machine learning by the academic community. Many questions have been raised, such as: How to define fairness (Hinnefeld et al., 2018; Hardt et al., 2016; Dwork et al., 2012; Kusner et al., 2017)? How to mitigate the sensitive bias (Zhang et al., 2018; Grari et al., 2019; Kamiran & Calders, 2012; Bellamy et al., 2018; Calmon et al., 2017; Zafar et al., 2015; Celis et al., 2019; Wadsworth et al., 2018; Louppe et al., 2017; Chen et al., 2019; Kearns et al., 2017)? How to keep a high prediction accuracy while remaining fair in a complex real-world scenario (Grari et al., 2019; Adel et al., 2019)? To answer these questions, three main families of fairness approaches exist in the literature. While pre-processing (Kamiran & Calders, 2012; Bellamy et al., 2018; Calmon et al., 2017) and post-processing (Hardt et al., 2016; Chen et al., 2019) approaches respectively act on the input or the output of a classically trained predictor, in-processing approaches mitigate the undesired bias directly during the training phase

(Zafar et al., 2015; Celis et al., 2019; Zhang et al., 2018; Wadsworth et al., 2018; Louppe et al., 2017). In this paper we focus on in-processing fairness, which reveals as the most powerful framework for settings where acting on the training process is an option.

Throughout this document, the aim is to learn a predictive function h_θ from training data that consists of m examples $(x_i, a_i, y_i)_{i=1}^m$, where $x_i \in \mathbb{R}^p$ is the p-sized feature vector X of the ith example, $a_i \in \Omega_A$ the value of its sensitive attribute and y_i its label to be predicted. According to the setting, the domain Ω_A of the sensitive attribute A can be either a discrete or a continuous set. The outcome Y is also either binary or continuous. The objective is to ensure some individual fairness guarantees on the outcomes of the predictor $\hat{Y} = h_\theta(X, A)$, by the way of Counterfactual Fairness.

2.1 Fairness definitions and metrics

The vast majority of fairness research works have focused on two metrics that have become very popular in the fairness field: *Demographic parity* (Dwork et al., 2012) and *Equalized odds* (Hardt et al., 2016). Both of them consider fairness globally, by focusing on equity between groups of people, classically defined according to one or several categorical sensitive attributes.

Some recent works recently proposed to extend this for the continuous setting by minimizing (non-linear) correlation between predictions $\hat{Y}$ and sensitive attributes A (Mary et al., 2019; Grari et al., 2019), that can be measured for instance via the Hirschfeld-Gebelein-Rényi maximal correlation (HGR).

However, even such approaches in the continuous setting only consider fairness globally and can lead to particularly unfair decisions at the individual level. For example, a fair algorithm can choose to accept a high MSE error for the outcome of a given person if this allows the distribution $P(\hat{Y}|A)$ to get closer to $P(\hat{Y})$. Penalization can be arbitrarily high on a given kind of individual profile compared to any other equivalent one, only depending on where the learning process converged. Global fairness is unfair.

To tackle this problem, Counterfactual fairness has been recently introduced for quantifying fairness at the most individual sense (Kusner et al., 2017). The idea is to consider that a decision is fair for an individual if it coincides with the one that would have been taken in a counterfactual world in which the values of its sensitive attributes were different. It leverages the previous work (Pearl, 2009), which introduced a causal framework to learn from biased data by exploring the relationship between sensitive features and data.

Definition 1 Counterfactual demographic parity (Kusner et al., 2017): A predictive function h_θ is considered counterfactually fair for a causal world G, if for any $x \in X$ and $\forall y \in Y$,$\forall (a, a\prime) \in \Omega_A^2$ with $a \neq a\prime$: $p(\hat{Y}_{A\leftarrow a} = y|X = x, A = a) = p(\hat{Y}_{A\leftarrow a\prime} = y|X = x, A = a)$, where $\hat{Y}_{A\leftarrow a'} = h_\theta(X_{A\leftarrow a'}, a')$ is the outcome of the predictive function h_θ for any transformation $X_{A\leftarrow a'}$ of input X, resulting from setting a' as its sensitive attribute value, according to the causal graph G.

Following Definition 1, an algorithm is considered counterfactually fair in term of demographic parity if the predictions are equal for each individual in the factual causal world where $A = a$ and in any counterfactual world where $A = a'$. It therefore compares the predictions of the same individual with an alternate version of him/herself. Similar extension can be done to adapt the *Equalized Odds* objective for the Counterfactual framework

(Pfohl et al., 2019). Learning transformations $\hat{X}_{A\leftarrow a'}$ for a given causal graph is at the heart of Counterfactual Fairness, as described in below.

2.2 Counterfactual fairness

In this paper, we focus on the classical causal graph depicted in Fig. 1, often used in the counterfactual fairness literature (Kusner et al., 2017; Pfohl et al., 2019; Chiappa, 2019), which can apply for most applications. For more specific tasks, note further that our approach could be easily adapted for different graphs, such as those explored in Kusner et al. (2017) for instances. In this causal graph, both input X and outcome Y only depend on the sensitive attribute A and a latent variable U, which represents all the relevant knowledge non dependent on the sensitive feature A. In that setting, the knowledge of U can be used during training to simulate various versions of the same individual, corresponding to different values of A, in order to obtain a predictive function h_θ which respects the fairness objective from Definition 1. For any training sample, U has to be inferred since only X, A and Y are observed. This inference must however ensure that no dependence is created between U and A (no arrow from U to A in the graph from Fig. 1), unless preventing the generation of proper alternative versions of X and Y for any values A.

Concerning causal effect identifiability (i.e., whether a joint distribution of latent and observed confounder variables can be uniquely inferred from observations), sufficient conditions as raised in (Louizos et al., 2017; Madras et al., 2019; Kilbertus et al., 2020) imply strong assumptions which require specific directed acyclic graphs different from ours. As in Pfohl et al. (2019), that considers the same causal graph, we make no formal guarantee on identification even in the case where these assumptions hold (more information in their article). However, we argue that, given any distribution $P(U, A, X, Y)$ exactly inferred from a sufficiently large amount of observations (X, Y, A), with a constant prior on U, the counterfactual quantities $P(X_{A\leftarrow a'}|X, Y, A)$ and $P(Y_{A\leftarrow a'}|X, Y, A)$ are identifiable, whenever U is independent from A. From this, if the prior $P(U)$ is the true one, and the decoding is sufficiently powerful, a classifier can be trained to minimize counterfactual unfairness according to the inferred model (step 2 in the following).

Several current approaches Louizos et al. (2017); Kim et al. (2021); Kocaoglu et al. (2017); Xu et al. (2019) enforce fairness on counterfactual data generated by their model. These works, which do not focus on the final predictor itself, assume that giving fair generated counterfactual observations as input to a traditional machine learning algorithm is sufficient to maintain the fairness objective. We argue that it is not always the case and the final predictions need to be evaluated to ensure a good fairness level. For this reason, we rather leverage a two-step method, as already considered in Russell et al. (2017); Pfohl et al. (2019), that focus separately on Causal Inference (step 1) and Model Learning

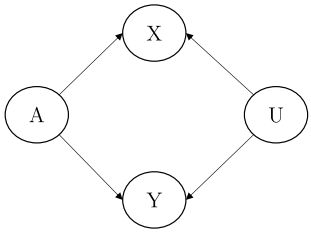


Fig. 1 Graphical causal model

(step 2). We develop and discuss the general principles of this family of methods in the following.

2.2.1 Step 1: Counterfactual Inference

The goal is to define a way to generate counterfactual versions of original individuals. As discussed above, this is usually done via approximate Bayesian inference, according to a pre-defined causal graph. The initial idea to perform inference was to suppose with strong hypothesis a non deterministic structural model with some specific distribution for all the causal links (Kusner et al., 2017). In this setting, the posterior distribution of U was estimated using the probabilistic programming language Stan (Team et al., 2016). Then, leveraging recent developments for approximate inference with deep learning, many works proposed to use Variational Autoencoding (Kingma & Welling, 2013) methods (VAE) to generalize this first model and capture more complex - non linear - dependencies in the causal graph. This leads to consider the following lower bound (ELBO) on the training set $\mathcal{D}$:

$$\mathcal{L}_{ELBO} = -\mathbb{E}_{\substack{(x,y,a) \sim \mathcal{D}, \\ u \sim q_\phi(u|x,y,a)}} \left[\log p_\theta(x,y|u,a)\right] + D_{KL}\big(q_\phi(u|x,y,a)||p(u)\big)]$$

where D_{KL} denotes the Kullback-Leibler divergence of the posterior $q_\phi(u|x,y,a)$ from a prior $p(u)$, typically a standard Gaussian distribution $\mathcal{N}(0,I)$. The posterior $q_\phi(u|x,y,a)$ is represented by a deep neural network with parameters ϕ, which typically outputs the mean μ_ϕ and the variance σ_ϕ of a diagonal Gaussian distribution $\mathcal{N}(\mu_\phi, \sigma_\phi I)$. The likelihood term factorizes as $p_\theta(x,y|u,a) = p_\theta(x|u,a)p_\theta(y|u,a)$, which are defined as neural networks with parameters θ. Since attracted by a standard prior, the posterior is supposed to remove probability mass for any features of U that are not involved in the reconstruction of X and Y. Since A is given together with U as input of the likelihoods, all the information from A should be removed from the posterior distribution of U.

However, some works (Chiappa, 2019; Louizos et al., 2017; Madras et al., 2019; Pfohl et al., 2019) show that the resulting latent space U and the sensitive variable A remain too highly correlated with this classical ELBO optimization. Some information from A leaks in the inferred U. To cope with it, a specific TARNet (Shalit et al., 2017) architecture can be employed (Madras et al., 2019) or a penalisation term can be added in the loss function. For example, (Chiappa, 2019; Pfohl et al., 2019) add a Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) constraint. The MMD term can be used to enforce all the different aggregated posterior to the prior distribution (Pfohl et al., 2019): $\mathcal{L}_{MMD}(q_\phi(u|A=a_k)||p(u))$ for all $a_k \in \Omega_A$ (referred to as MMD wrt $P(U)$ in the following). Alternatively, the constraint can directly enforce the matching between pairs of posteriors (Chiappa 2019): $\mathcal{L}_{MMD}(q_\phi(u|A=a_k)||q_\phi(u|A=a))$ for all $a_k \in \Omega_A$, with a standing for the original sensitive value of the considered individual (referred to as MMD wrt U_a in the following). Notice that while this additional term can improve independence, it can also encourage the model to ignore the latent confounders U, by being too restrictive. One possible approach to address this issue is to apply weights λ (hyperparameters) to control the relative importance of the different terms. In addition, we employ in this paper a variant of the ELBO optimization as done in Pfohl et al. (2019), where the $D_{KL}(q_\phi(u|x,y,a)||p(u))$ term is replaced by a MMD term $\mathcal{L}_{MMD}(q_\phi(u)||p(u))$ between the aggregated posterior $q_\phi(u)$ and the prior. This has been shown more powerful than the classical D_{KL} for ELBO

optimization in Zhao et al. (2017), as the latter can reveal as too restrictive (uninformative latent code problem) (Chen et al. 2016; Bowman et al. 2015; Sønderby et al. 2016) and can also tend to overfit the data (Variance Over-estimation in Feature Space). Finally, the inference for counterfactual fairness can be optimized by minimizing (Pfohl et al. 2019):

$$\mathcal{L}_{CE-VAE} = - \mathop{\mathbb{E}}_{\substack{(x,y,a) \sim \mathcal{D}, \\ u \sim q_\phi(u|x,y,a)}} \begin{bmatrix} \lambda_x \log(p_\theta(x|u,a)) + \\ \lambda_y \log(p_\theta(y|u,a)) \end{bmatrix} + \lambda_{MMD}\, \mathcal{L}_{MMD}(q_\phi(u)||p(u)) + \frac{\lambda_{ADV}}{|\Omega_A|} \sum_{a_k \in \Omega_A} \mathcal{L}_{MMD}(q_\phi(u|a=a_k)||p(u))$$

where λ_x, λ_y, λ_{MMD}, λ_{ADV} are scalar hyperparameters. The additional MMD objective can be interpreted as minimizing the distance between all moments of each aggregated latent code distribution and the prior distribution, in order to remove most sensitive dependency from the code generator. It requires however a careful design of the kernel used for MMD computations (typically a zero mean isotropic Gaussian). Note that we chose to present all models with a generic inference scheme $q(U|X, Y, A)$, while most approaches from the literature only consider $q(U|X, A)$. The use of Y as input is allowed since U is only used during training, for generating counterfactual samples used to learn the predictive model in step 2. Various inference schemes are considered in our experiments (Sect. 4).

2.2.2 Step 2: Counterfactual predictive model

Once the causal model is learned, the goal is to use it to learn a fair predictive function h_θ, by leveraging the ability of the model to generate alternative versions of each sample. The global loss function is usually composed of the traditional predictor loss $l(h_\theta(x_i, a_i), y_i)$ (e.g. cross-entropy for instance i) and the counterfactual unfairness estimation term $\mathcal{L}_{\mathcal{CF}}(\theta)$:

$$\mathcal{L} = \frac{1}{m} \sum_{i}^{m} l\big(h_\theta(x_i), y_i\big) + \mathcal{L}_{\mathcal{CF}}(\theta) \tag{1}$$

where λ is an hyperparameter which controls the impact of the counterfactual loss in the optimization. The counterfactual loss $\mathcal{L}_{\mathcal{CF}}(\theta)$ considers differences of predictions for alternative versions of any individual. For example, Russell et al. (2017) considers the following Monte-Carlo estimate from S samples for each individual i and each value $a \in \Omega_A$:

$$\mathcal{L}_{\mathcal{CF}}(\theta) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{m_a} \sum_{a_k \in \Omega_A} \frac{1}{S} \sum_{s=1}^{S} \Delta_{a_k}^{i,s} \tag{2}$$

where $\Delta_{a_k}^{i,s} = \Delta(h_\theta(x^s_{i,A\leftarrow a_i}, a_i), h_\theta(x^s_{i,A\leftarrow a_k}, a_k))$ is a loss function that compares two predictions, $x^s_{i,A\leftarrow a}$ denotes the s-th sample from the causal model for the i-th individual of the training set and the sensitive attribute value a. Following the causal model learned at step 1, $x^s_{i,A\leftarrow a}$ is obtained by first inferring a sample u from $q_\phi(u|x_i, a_i, y_i)$ and then sampling $x^s_{i,A\leftarrow a}$ using $p_\theta(x|u, a)$ with the counterfactual (or factual) attribute value a. According to the task, Δ can take various forms. For binary classification, it can correspond to a logit paring loss as done in Pfohl et al. (2019): $\Delta(z, z') = (\sigma^{-1}(z) - \sigma^{-1}(z'))^2$, where σ^{-1} is the logit function. For continuous outcomes, it can simply correspond to a mean squared difference.

2.2.3 Discussion

For now, state-of-the-art approaches have focused specifically on categorical variables A. Unfortunately, the classical methodology for CounterFactual Fairness as described above cannot be directly generalized for continuous sensitive attributes, because the two steps involve enumerations of the discrete counterfactual modalities a_k in the set Ω_A. Particularly in step 1, sampling A from a uniform distribution for approximating the expectation $E_{a\sim p(A)}\mathcal{L}_{MMD}\big(q_\phi(u|A=a)||p(u)\big)$ is not an option since this requires to own a good estimation of $q_\phi(u|A=a)$ for any $a \in \Omega_A$, which is difficult in the continuous case. While such a posterior can be obtained for discrete sensitive attributes (at least when $|\Omega_A| << m$) by aggregating the posteriors $q_\phi(u|x_i, a_i, y_i)$ over training samples i such that $a_i = a$, such a simple aggregation over filtered samples is not possible for continuous attributes. Note that splitting samples in bins regarding to their sensitive value A is not an option due to the difficulty for setting an effective discretization step size : while a large step size induces aggregating too different sensitive values (leading to dependencies on A inside bins), small steps imply unreliable aggregated posterior estimates due to small numbers of samples in each bin (especially for data unevenly spread over Ω_A).

Moreover, existing approaches based on MMD costs imply to infer codes U from a distribution that takes A as input, in order to be able to obtain the required aggregated distributions via: $q_\phi(u|a) = \mathbb{E}_{p_{data}(x,y|a)}[q_\phi(u|x, y, a)]$. Omitting A from the conditioning of the generator would correspond to assume the mutual independence of u and a given x and y, which is usually wrong. On the other hand, passing A to the generator of U can encourage their mutual dependency in some settings, as we observe in our experiments. This is not the case with our proposal below.

3 Adversarial learning for counterfactual fairness

In this section we revisit the 2 steps shown above by using adversarial learning rather than MMD costs for ensuring Counterfactual Fairness. Our contribution covers a broad range of scenarios, where the sensitive attribute A and the outcome value Y can be either discrete or continuous.

3.1 Step 1: Counterfactual Inference

To avoid the comparison of distributions for each possible sensitive value, which reveals particularly problematic in the continuous setting, we propose to employ an adversarial learning framework, which allows one to avoid the enumeration of possible values in Ω_A. We follow an approach similar to the adversarial auto-encoders proposed in Makhzani et al. (2015), but where the discriminator real/fake data is replaced by a sensitive value predictor. The idea is to avoid any adversarial function to be able to decode A from the code U inferred from the encoder q_ϕ, which allows one to ensure mutual independence of A and U. This defines a two-players adversarial game, such as in GANs (Goodfellow et al., 2014), where the goal is to find some parameters ϕ which minimize the loss to reconstruct X and Y, while maximizing the reconstruction loss of A according to the best decoder $p_\psi(A|U)$:

$$\arg \min_{\theta,\phi} \max_{\psi} \mathcal{L}_{ADV}(\theta, \phi, \psi) \tag{3}$$

$$\mathcal{L}_{ADV}(\theta,\phi,\psi) \geq - \mathop{\mathbb{E}}_{\substack{(x,y,a)\sim\mathcal{D},\\ u\sim q_\phi(u|x,y,a)}} \left[\lambda_x \log(p_\theta(x|u,a)) + \lambda_y \log(p_\theta(y|u,a))\right] + \lambda_{MMD}\,\mathcal{L}_{MMD}(q_\phi(u)||p(u)) + \lambda_{ADV} \mathop{\mathbb{E}}_{\substack{(x,y,a)\sim\mathcal{D},\\ u\sim q_\phi(u|x,y,a))}} [log(p_\psi(a|u))]$$

where λ_x, λ_y, λ_{MMD}, λ_{ADV} are scalar hyperparameters. Compared to existing approaches presented in the previous section, the difference is the last term which corresponds to the expectation of the log-likelihood of A given U according to the decoder with parameters ϕ. This decoder corresponds to a neural network which outputs the parameters of the distribution of A given U (i.e., the logits of a Categorical distribution for the discrete case, the mean and log-variance of a diagonal Gaussian in the continuous case).

All parameters are learned conjointly. Figure 2 gives the full architecture of our variational adversarial inference for the causal model from Fig. 1. It depicts the neural network encoder $q_\phi(U|X,Y,A)$ which generates a latent code U from the inputs X, Y and A. A neural network decoder $p_\theta(X,Y|U)$ reconstructs the original X and Y from both U and A. The adversarial network p_ψ tries to reconstruct the sensitive attribute A from the confounder U. As classically done in adversarial learning, we alternate steps for the adversarial maximization and steps of global loss minimization (one gradient descent iteration on the same batch of data at each step). Optimization is done via the re-parametrization trick (Kingma & Welling, 2013) to handle stochasticity.

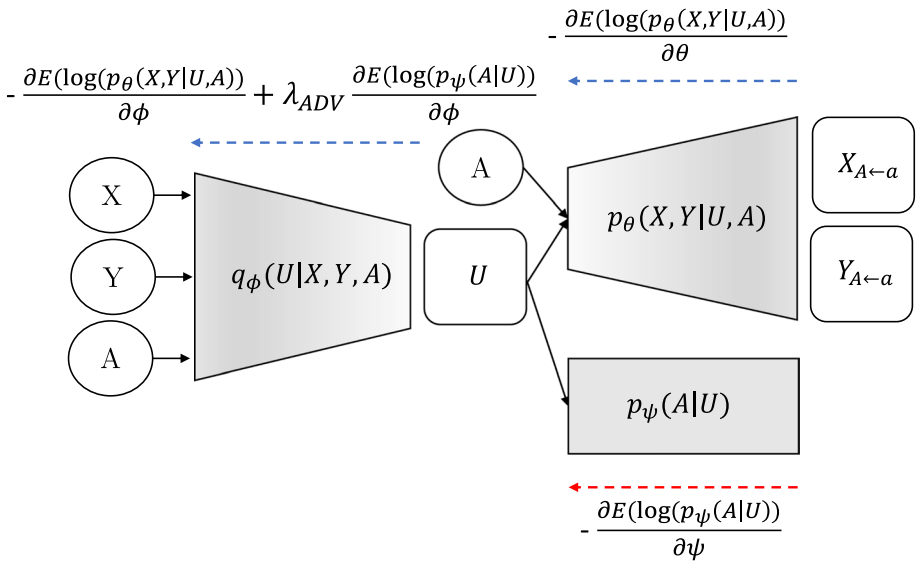


Fig. 2 Our Counterfactual inference architecture. Circles are observed variables, squares are samples from the neural distributions. Arrows represent retro-propagated gradients

3.2 Step 2: Counterfactual predictive model

As described in Sect. 2.3, the counterfactual fairness in the predictive model learned at step 2 is ensured by comparing, for each training individual, counterfactual predictions $Y_{A\leftarrow a'}$ for all $a' \in \Omega_A$. For the discrete case (i.e., A is a Categorical variable), we keep this process for our experiments. However, for the continuous setting (i.e., A is for instance generated from a Gaussian), such an approach must be somehow adapted, due to the infinite set Ω_A. In that case, we can consider a sampling distribution $P'(A)$ to formulate the following loss, which can be optimized via Monte-Carlo sampling and stochastic gradient descent (SGD):

$$\mathcal{L}_{CF}(\theta) = \frac{1}{m}\sum_{i}^{m} l(h_\theta(x_i), y_i) + \lambda \underset{\substack{u \sim P(u|x_i,a_i,y_i),\\ \tilde{x} \sim P(x|u_i,a_i),\\ a' \sim P'(A), x' \sim P(x|u,a')}}{\mathbb{E}} [(h_\theta(\tilde{x}) - h_\theta(x'))^2] \tag{4}$$

This formulation is equivalent to the one from Eq. (2), for continuous outcomes $\hat{Y}$ (thus considering a least squared cost as Δ) and for continuous attributes A (thus using the sampling distribution $P'(A)$ rather than considering every possible $a \in \Omega_A$). Note that using a non-uniform sampling distribution $P'(A)$ would enforce the attention of the penalisation near the mass of the distribution. This prevents using the prior of A estimated from the training set, since this would tend to reproduce inequity between individuals: counterfactual predictions for rare A values would be little taken into account during training. We therefore consider a uniform $P'(A)$ in our experiments for the continuous setting when using the $\mathcal{L}_{CF}(\theta)$ objective at step 2.

However, for the specific case of high-dimensional sensitive attributes A, using a uniform sampling distribution $P'(A)$ could be inefficient. The risk is that a high number of counterfactual samples fall in easy areas for the learning process, while some difficult areas - where an important work for fairness has to be performed - remain insufficiently visited. To tackle this problem, we propose to allow the learning process to dynamically focus on the most useful areas of Ω_A for each individual. During learning, we consider an adversarial process, which is in charge of moving the sampling distribution $P'(A)$, so that the counterfactual loss is the highest. This allows the learning process to select useful counterfactuals for ensuring fairness. Who can do more can do less: dynamically focusing on hardest areas allows one to expect fairness everywhere. Again, we face a two-players adversarial game, which formulates as follows:

$$\arg\min_{\theta} \arg\max_{\phi} \quad \mathcal{L}_{DynCF}(\theta, \phi)$$

$$\mathcal{L}_{DynCF}(\theta, \phi) = \frac{1}{m}\sum_{i}^{m} l(h_\theta(x_i), y_i) + \lambda \underset{\substack{u \sim P(u|x_i,a_i,y_i),\\ \tilde{x} \sim P(x|u,a_i),\\ a' \sim P_\phi(a|u), x' \sim P(x|u,a')}}{\mathbb{E}} [(h_\theta(\tilde{x}) - h_\theta(x'))^2] \tag{5}$$

Compared to Eq. (4), this formulation considers an adversarial sampling distribution $P_\phi(A|U)$ rather than a uniform static distribution $P'(A)$. It takes the form of a neural network that outputs the parameters of the sampling distribution for a given individual representation U. In our experiments we use a diagonal logit-Normal distribution $sigmoid(\mathcal{N}(\mu_\phi(u), \sigma^2_\phi(u)I))$, where $\mu_\phi(u)$ and $\sigma^2_\phi(u)$ stand for the mean and variance parameters provided by the network for the latent code u. Samples from this distribution are then

projected on the support Ω_A via a linear mapping depending on the shape of the set. Passing U as input for the network allows the process to define different distributions for different codes: according to the individual profiles, the unfair areas are not always the same. This also limits the risk that the adversarial process gets stuck in sub-optimums of the sensitive manifold. As done for adversarial learning in step 1, all parameters are learned conjointly, by alternating steps for the adversarial maximization and steps of global loss minimization. The re-parametrization trick (Kingma & Welling, 2013) is also used, for the adversarial optimization of $P_\phi(A|U)$.

4 Experiments

We empirically evaluate the performance of our contribution on 6 real world data sets. For the discrete scenario and specifically in the binary case ($Y \in \{0,1\}, A \in \{0,1\}$), we use 3 different popular data sets: the Adult UCI income data set (Dua & Graff, 2017) with a gender sensitive attribute (male or female), the COMPAS data set (Angwin et al., 2016) with the race sensitive attribute (Caucasian or not-Caucasian) and the Bank dataset (Moro et al., 2014) with the age as sensitive attribute (age is between 30 and 60 years, or not). For the continuous setting (Y and A are continuous), we use the 3 following data sets: the US Census dataset (US Census Bureau, 2019) with gender rate as sensitive attribute encoded as the percentage of women in the census tract, the Motor dataset The Institute of Actuaries of France (2015) with the driver's age as sensitive attribute and the Crime dataset (Dua & Graff, 2017) with the ratio of an ethnic group per population as sensitive attribute.

Additionally to the 6 real-world datasets, we consider a synthetic scenario, that allows us to perform a further analysis of the relative performances of the approaches. The synthetic scenario subject is a pricing algorithm for a fictional car insurance policy, which follows the causal graph from Fig. 1. We simulate both a binary and a continuous dataset from this scenario. The main advantage of these synthetic scenarios is that it is possible to get "ground truth" counterfactuals for each code U, obtained using the true relationships of the generation model while varying A uniformly in Ω_A. This will allow us to evaluate the counterfactual fairness of the models without depending on a given inference process for the evaluation metric, by relying on prediction differences between these true counterfactuals and the original individual. The objective of this scenario is to achieve a counterfactual fair predictor which estimates the average cost history of insurance customers. We suppose 5 unobserved variables (Aggressiveness, Inattention, Restlessness, Reckless and Overreaction) which corresponds to a 5 dimensional confounder U. The input X is composed of four explicit variables $X_1, ..., X_4$ which stand for vehicle age, speed average, horsepower and average kilometers per year respectively. We consider the policyholder's age as sensitive attribute A. The input X and the average cost variable Y are sampled from U and A as depicted in Fig. 1 from the main paper. We propose both a binary and a continuous version of this scenario. For both of them, 5000 individuals are sampled. Details of distributions used for the continuous setting of this synthetic scenario are given below:

$$
\begin{aligned}
U &\sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0.5 \\ 1 \\ 1.5 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}\right) \\
X1 &\sim \mathcal{N}(7 + 0.1 * A + U_1 + U_2 + U_3, 1); \\
X2 &\sim \mathcal{N}(80 + A + U_2^2, 10); \\
X3 &\sim \mathcal{N}(200 + 5 * A + 5 * U_3, 20); \\
X4 &\sim \mathcal{N}((10^4 + 5 * A + U_4 + U_5, 1000) \\
X &\sim [X1, X2, X3, X4]; \\
A &\sim \mathcal{N}[45, 5]; \\
Y &\sim \mathcal{N}(2 * (7 * A + 20 * \sum_j U_j), 0.1)
\end{aligned}
$$

4.1 Step 1: Counterfactual Inference

In this section, we report experiments performed for assessing our adversarial approach for Counterfactual Inference (step 1 of the previous section). We compare our adversarial approach with two version of the approach in Eq. , each using one of the two MMD constraints MMD wrt $P(A)$ or MMD wrt U_a as presented in Sect. 2.2 (step 1). Note that these approaches are not applicable for continuous datasets as discussed at the end of Sect. 2. For every approach, we compare three different inference schemes for U: $q_\phi(u|x, y, a)$, $q_\phi(u|x, y)$ and $q_\phi(u|x, a)$. As a baseline, we also use a classical Variational Autoencoder inference without counterfactual independence constraint (i.e., Eq. (4) without the last term).

All hyper-parameters for every approach have been tuned by 5-fold cross-validation. For the US Census data set for our approach for instance, the encoder q_ϕ architecture is an MLP of 3 hidden layers with 128, 64 and 32 units respectively, with ReLU activations. On this dataset, the decoder p_θ is an MLP of only one hidden layer with 64 units with a ReLu activation function and the output consists in one single output node with linear activation to reconstruct Y and 37 units to reconstruct X (number of features). The adversarial neural network p_ψ is an MLP of two hidden layers with 32 and 16 units respectively. For the binary datasets, a sigmoid is applied on the outputs of decoders for A and Y. For both MMD constraints we used a Gaussian radial basis function kernel. For all datasets, the prior distribution $p(U)$ considered for training the models is a five-dimensional standard Gaussian.

In order to evaluate the level of dependence between the latent space U and the sensitive variable A, we compare the different approaches by using the neural estimation of the *HGR* correlation coefficient given in Grari et al. (2019). This coefficient, assesses the level of non-linear dependency between two jointly distributed random variables. The estimator is trained for each dataset and each approach on the train set, comparing observed variables A with the corresponding inferred codes U.

For all data sets, we repeat five experiments by randomly sampling two subsets, 80% for the training set and 20% for the test set. Finally, we report the average reconstruction loss for X and Y on the test set, as long as the HGR between inferred test codes and the

corresponding sensitive attributes. Results of our experiments can be found in Table 1 for the discrete case and Table 2 for the continuous case. For all of them, we attempted via the different hyperparameters (λ_x, λ_y, λ_{MMD}, λ_{ADV}) to obtain the lower dependence measure while keeping the minimum loss as possible to reconstruct X and Y.

As expected, the baseline without the independence constraint achieves the best X and Y reconstruction loss, but this is also the most biased one with the worst dependence in term of HGR in most datasets. Comparing the different constraints in the discrete case, the adversarial achieves globally the best result with the lower HGR while maintaining a reasonable reconstruction for X and Y. It is unclear which MMD constraint performs better than the other. We observe that the best results in terms of independence are obtained without the sensitive variable given as input of the inference network (inference only with X and Y). Note however that for the MMD constraints, this setting implies to make the wrong assumption of independence of U w.r.t. A given X and Y for the estimation of the constraint (as discussed at the end of Sect. 2). This is not the case for our adversarial approach, which obtains particularly good results on this setting for discrete datasets. On continuous datasets, our approach succeeds in maintaining reasonable reconstruction losses for important gains in term of HGR compared to the classical VAE approach (without constraint). Interestingly, on these datasets, it appears that our approach obtains slightly better results when using the full information (X, Y and A) as input of the inference network. We explain this by the fact that removing the influence of a binary input is harder than the one of a smoother continuous one, while this can reveal as a useful information for generating relevant codes.

4.2 Step 2: Counterfactual predictive model

This section reports experiments involving the training procedure from step 2 as described in Sect. 3. The goal of these experiments is threefold: 1. assess the impact of the adversarial inference on the target task of counterfactual fairness, 2. compare our two proposals for counterfactual bias mitigation (i.e., using a uniform distribution or an adversarial dynamic one for the sampling of counterfactual sensitive values) and 3. assess the impact of the control parameter from Eq. (5).

The predictive model used in our experiments is a MLP with 3 hidden layers. The adversarial network P_ϕ from Eq. (5) is a MLP with 2 hidden layers and RELU activation. For all our experiments, a single counterfactual for each individual is sampled at each iteration during the training of the models. Optimization is performed using ADAM.

Tables 3 and 4 report results for the discrete and the continuous case respectively. The inference column refers to the inference process that was used for sampling counterfactuals for learning the predictive model. For each setting, we use the best configuration from Tables 1 and 2. The mitigation column refers to the type of counterfactual mitigation that is used for the results: No mitigation or L_{CF} (Eq. 2) for the discrete case; No mitigation, L_{CF} (Eq. 4) or L_{DynCF} (Eq. 5) for the continuous setting. Results are reported in terms of accuracy (for the discrete case) or MSE (for the continuous case) and of Counterfactual Fairness (CF). The CF measure is defined, for the m_{test} individuals from the test set, as:

$$CF = \frac{1}{m_{test}} \sum_{i}^{m_{test}} \mathbb{E}_{(x',a')\sim C(i)}[\Delta(h_\theta(x_i, a_i), h_\theta(x', a'))] \tag{6}$$

where $C(i)$ is the set of counterfactual samples for the i-th individual of the test set. This corresponds to counterfactuals sampled with the Adversarial inference process defined at

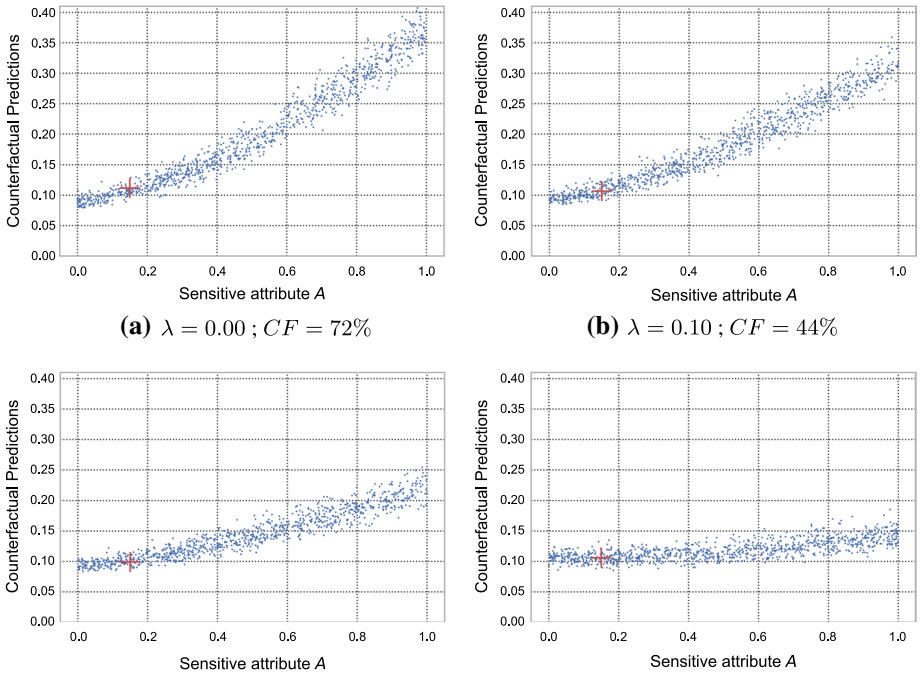
step 1 (with the best configuration reported in Tables 1 and 2). As discussed above, the synthetic datasets allow one to rely on "true" counterfactuals for the computation of counterfactual fairness, rather than relying on an inference process which may include some bias. For these datasets, we thus also report an additional RealCF metric, which is defined as in Eq. (6), but using these counterfactuals sampled from the true codes used to generate the test data. For both CF and RealCF, for every i from the test set, $|C(i)|$ equals 1 for binary settings and $|C(i)|$ equals 1000 for the continuous one. Δ is a cost function between two predictions, the logit paring cost for the binary case (more details given in Sect. 2.2 step 2) and a simple squared difference for the continuous setting.

Results from both tables first confirm the good behavior of our inference model from step 1, which allows one to obtain greatly better results than other inference processes for both the discrete and the continuous settings. Our adversarial counterfactual inference framework allows one to get codes that can be easily used to generate relevant counterfactual individuals. For this observation, the most important results are those given for the synthetic scenarios, for which the RealCF metric shows good results for our method, while strongly reliable since relying on counterfactuals sampled from true codes of individuals.

Secondly, results from Table 2 show that, even in the continuous setting where the enumeration of all values from Ω_A is not possible, it is possible to define counterfactual mitigation methods such as our approaches L_{CF} and L_{DynCF}. These two methods, used in conjunction with our Adversarial Inference, give significantly better results than no mitigation on every dataset. Interestingly, we also observe that L_{DynCF} allows one to improve results over L_{CF}, which shows the relevance of the proposed dynamic sampling process. Furthermore, note that we can reasonably expect even better results compared to L_{CF} on data with higher-dimensional sensitive attributes.

To illustrate the impact of the hyperparameter λ on the predictions accuracy (MSE Error) and the counterfactual fairness estimation (CF), we plot results for 10 different values of λ (5 runs each) on Fig. 4 for the Crime data set. It clearly confirms that higher values of λ produce fairer predictions, while a value near 0 allows one to only focus on optimizing the predictor loss. This is also observable from Fig. 3 which plots counterfactual predictions for a specific instance i from the test set. Higher values of λ produce clearly more stable counterfactual predictions.

In Fig. 5, we consider the distribution of considered counterfactual samples w.r.t. to the sensitive variable A for the uniform sampling strategy from $P'(A)$ and the dynamic strategy as defined in Eq. (5). This is done on the Motor dataset and for a specific randomly sampled instance i with sensitive attribute $a_i = 75$, at a given point of the optimization, far before convergence (the model is clearly unfair at this point). The blue points are the counterfactual fairness estimation $(h_\theta(X_{i,A\leftarrow a}, a) - h_\theta(X_{i,A\leftarrow a'}, a'))$ for each counterfactual sampled a' s (1.000 points) from the uniform distribution $P'(A)$. The red points are the counterfactual fairness estimations for counterfactuals corresponding to a' values (30 points) sampled from our dynamic distribution $P_\phi(a'|u) = \mathcal{N}(\mu_\phi(u), \sigma^2_\phi(u)I)$, where ϕ are the parameters of the adversarial network which optimizes the best mean and variance for each latent code u ($\mu_\phi(u)$ and $\sigma^2_\phi(u)$). Being optimized to maximize the error at each gradient step, the red points are sampled on lower values of A where the error is the most important. More importantly, very few points are sampled in the easy area, near the true sensitive value of i which is 75. This demonstrates the good behavior of our dynamic sampling process.



(a) $\lambda = 0.00\,;\, CF = 72\%$ **(b)** $\lambda = 0.10\,;\, CF = 44\%$

(c) $\lambda = 0.28\,;\, CF = 07\%$ **(d)** $\lambda = 0.30\,;\, CF = 04\%$

Fig. 3 Impact of λ (Crime data set) on a specific instance i. Blue points are counterfactual predictions $h_\theta(x_{i,A\leftarrow a'})$ from 1.000 points $A \leftarrow a'$ generated randomly. The red cross represents the prediction $h_\theta(x_{i,A\leftarrow a})$ for the real $A = a$ of instance i (Color figure online)

4.3 Total and counterfactual effect

In addition, we propose to compare performances of our approach with works based on fair data generation (Louizos et al., 2017; Kim et al., 2021; Kocaoglu et al., 2017; Xu et al., 2019), such as mentioned at the end of Sect. 2.2, to emphasize the benefits of our two-steps process for learning counter-factually fair prediction models.

Traditionally, these methods are evaluated in terms of total and counterfactual causal effect of the sensitive on the data generated by the models. Total causal effect (TCE) aims at assessing the statistical parity on the outcomes generated from causal intervention. TCE for binary sensitives is defined as:

$$TCE = P(Y_{A\leftarrow a_1}) - P(Y_{A\leftarrow a_0}) \tag{7}$$

where $Y_{A\leftarrow a}$ corresponds to generated causal transformation of input Y, resulting from setting a as the sensitive attribute to the corresponding individual, according to the causal graph G (i.e., obtained via distribution $P(Y_{A\leftarrow a}|X, Y, A)$).

A limit of such a metric is that it only considers fairness in the data given as training set for learning the predictor model. We claim that this is not enough since any residual bias in these data may strongly impact the final prediction (on both training and testing data). To overcome this limitation, and assess total effect of sensitives on predictions rather than on training data only, we introduce the Total Predictions Effect (TPE), which refers to the

Table 1 Inference results in the discrete case

		Adult UCI			Compas			Bank			Synthetic Scenario		
		Loss X	Loss Y	HGR	Loss X	Loss Y	HGR	Loss X	Loss Y	HGR	Loss X	Loss Y	HGR
($\{x, y, a\}$)	No Constraint, $q(u\|x, y, a)$	0.0781	0.0006	0.6984	0.0278	0.0041	0.6952	0.0963	0.0001	0.5988	0.2681	0.0085	0.9725
	Adv. Constraint, $q(u\|x, y, a)$	0.1091	0.0009	**0.5453**	0.0254	0.0020	**0.2693**	0.2038	0.0005	**0.3423**	0.2669	0.0721	**0.4167**
	MMD wrt $P(U)$, $q(u\|x, y, a)$	0.1286	0.0012	0.7017	0.0252	0.0029	0.6565	0.2002	0.0002	0.4521	0.2535	0.0839	0.6623
	MMD wrt U_a, $q(u\|x, y, a)$	0.0938	0.0009	0.7181	0.0259	0.0098	0.8892	0.1263	0.0003	0.5188	0.2762	0.0351	0.5697
($\{x, y\}$)	No Constraint, $q(u\|x, y)$	0.0786	0.0008	0.6077	0.0274	0.0133	0.3817	0.0957	0.0001	0.4989	0.2577	0.0022	0.6418
	Adv. Constraint, $q(u\|x, y)$	0.1272	0.0329	**0.1811**	0.0245	0.0013	**0.1728**	0.1858	0.0073	**0.2476**	0.2649	0.1015	0.4521
	MMD wrt $P(U)$, $q(u\|x, y)$	0.1287	0.0016	0.6092	0.0259	0.0055	0.4470	0.1898	0.0003	0.3716	0.2567	0.0885	0.6868
	MMD wrt U_a, $q(u\|x, y)$	0.0872	0.0013	0.6852	0.0266	0.0094	0.3109	0.1415	0.0003	0.3929	0.2674	0.0553	**0.4473**
($\{x, a\}$)	No Constraint, $q(u\|x, a)$	0.0982	0.3534	0.6689	0.0288	0.8246	0.3726	0.1391	0.2101	0.5572	0.2686	0.0128	0.7040
	Adv. Constraint, $q(u\|x, a)$	0.0995	0.3462	0.5259	0.0271	0.6889	0.4344	0.1880	0.2110	**0.3061**	0.2589	0.0980	**0.4264**
	MMD wrt $P(U)$, $q(u\|x, a)$	0.1308	0.3559	**0.3586**	0.0288	0.7611	0.4365	0.2141	0.2129	0.3386	0.2506	0.1176	0.6298
	MMD wrt U_a, $q(u\|x, a)$	0.0940	0.3603	0.5811	0.0278	0.7314	**0.3345**	0.1485	0.2135	0.5536	0.2584	0.1076	0.4692

Inference results in the discrete case

The results in bold represent the best performance achieved for the HGR dependence

Table 2 Inference results in the continuous case

	US census			Motor			Crime			Synthetic scenario		
	Loss X	Loss Y	HGR	Loss X	Loss Y	HGR	Loss X	Loss Y	HGR	Loss X	Loss Y	HGR
No Cons. $q(u\|x, y, a)$	0.1685	0.0019	0.5709	0.2526	0.0024	0.9023	0.4558	0.0016	0.9059	0.6788	0.0076	0.9523
No Cons. $q(u\|x, y)$	0.1690	0.0005	0.4163	0.3068	0.0034	0.9479	0.4523	0.0018	0.8998	0.6495	0.0003	0.6227
No Cons. $q(u\|x, a)$	0.1726	0.2886	0.8252	0.3377	0.9381	0.9728	0.4634	0.3999	0.9076	0.6751	0.4554	0.8650
Adv $q(u\|x, y, a)$	0.1617	0.0004	0.3079	0.4702	0.0035	0.2941	0.4865	0.0701	0.5268	0.6804	0.0088	0.2280
Adv $q(u\|x, y)$	0.1663	0.0009	0.2980	0.3694	0.0057	0.3314	0.4835	0.0571	0.6024	0.6633	0.1196	0.3175
Adv $q(u\|x, a)$	0.1828	0.2891	0.3285	0.4706	0.9878	0.2478	0.4904	0.3933	0.5810	0.6862	0.8819	0.5148

Inference results in the continuous case

Table 3 Counterfactual fairness results for the discrete case

Inference	Mitigation	Adult UCI		Compas		Bank		Synthetic scenario		
		Accuracy (%)	CF	Accuracy	CF	Accuracy (%)	CF	Accuracy (%)	CF	Real CF
Without Constraint	None	84.22	0.0096	67.12	0.0102	90.64	0.0369	99.49	0.1087	0.1810
	L_{CF}	83.28	0.0008	66.20	0.0051	90.46	0.0024	95.89	0.0757	0.1327
MMD	None	84.22	0.0116	67.12	0.0076	90.64	0.0469	99.49	0.1074	0.1775
	L_{CF}	83.84	0.0024	65.91	0.0041	90.64	0.0043	99.29	0.0893	0.1557
Adversarial	None	84.22	0.0114	67.12	0.0118	90.64	0.0376	99.49	0.1426	0.1838
	L_{CF}	83.74	0.0002	66.73	0.0001	90.60	0.000	93.19	0.0001	0.0014

Table 4 Counterfactual fairness results for the continuous case

Inference	Mitigation	US census		Motor		Crime		Synthetic scenario		
		Accuracy	CF	MSE	CF	MSE	CF	MSE	CF	Real CF
Adversarial	None	0.274	0.0615	0.938	0.0285	0.412	0.7412	0.454	0.2490	1.1248
	L_{CF}	0.289	0.0009	0.941	0.0009	0.452	0.0154	0.572	0.0014	0.2013
	L_{DynCF}	0.290	0.0008	0.940	0.0005	0.445	0.0076	0.568	0.0013	0.2000
Without Constraint	None	0.274	0.0433	0.938	0.0271	0.381	0.7219	0.454	0.2919	1.1338
	L_{CF}	0.307	0.0010	0.939	0.0021	0.407	0.2938	0.531	0.1968	0.3303
	L_{DynCF}	0.310	0.0008	0.942	0.0016	0.418	0.2881	0.546	0.1743	0.3188

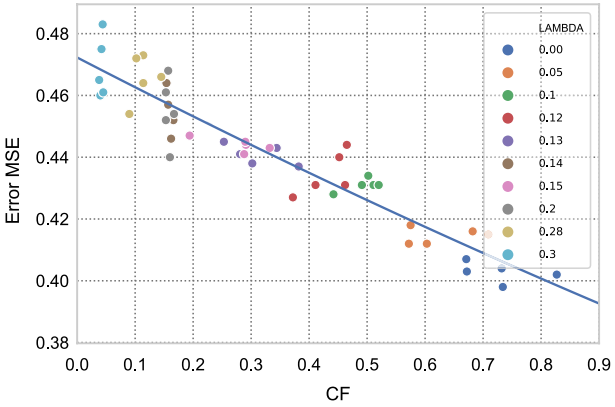


Fig. 4 Impact of hyperparameter λ (Crime data set): Higher values of λ produce fairer predictions, while λ near 0 allows to only focus on optimizing the regression loss

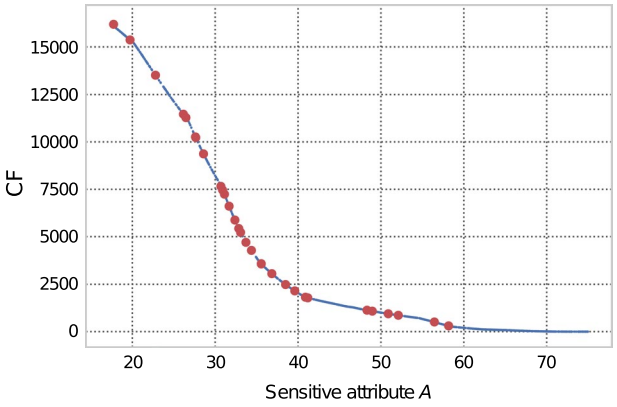


Fig. 5 Dynamic Sampling Visualization for a randomly sampled individual whose age A is 75. Red points are sampled counterfactuals from the dynamic distribution $P_\phi(a'|u)$ with u the inferred confounding for this individual (Color figure online)

statistical parity of the output prediction from intervention. The metric is defined in the binary case as:

$$TPE = P(h_\theta(X_{A\leftarrow a_1})) - P(h_\theta(X_{A\leftarrow a_0})) \tag{8}$$

which takes into account the fairness of the predictor from transformed data $X_{A\leftarrow a}$.

Following (Kim et al., 2021; Xu et al., 2019), we also consider counterfactual effects, which depend on the effect of the sensitive on the outcome for specific individuals (or groups of individuals). Similarly as for the total effect, for any observation o, we consider the Counterfactual Causal Effect defined as: $CCE = P(Y_{A\leftarrow a_1}|o) - P(Y_{A\leftarrow a_0}|o)$ and introduce the Counterfactual Prediction Effect as: $CPE = P(h_\theta(X_{A\leftarrow a_1})|o) - P(h_\theta(X_{A\leftarrow a_0})|o)$.

Causal Effect In Table 5, we represent the results from the different generated data observations on the Adult UCI dataset. We consider the condition observations o as the concatenation of the features *race* and *native_country* as in Xu et al. (2019); Kim et al. (2021) ($O = \{race, native\_country\}$). We report the chi-square distance χ^2 that indicates the similarity between the generated and the real dataset. We consider three baselines that are unaware of the fairness constraint: CausalGan (Kocaoglu et al., 2017) that preserves the

Table 5 Total causal effect and counterfactual causal effect on adult UCI

	Total causal effect (TCE)	Counterfactual causal effect (CCE)				X^2
		o_{00}	o_{01}	o_{10}	o_{11}	
Real Data	0.1936	0.1785	0.1266	0.1293	0.2023	0
Causal GAN	0.1729	0.0717	0.1201	0.1326	0.1856	20388
DCEVAE WR	0.1819	0.1694	0.1472	0.1522	0.1899	20822
OURS	0.1834	0.1783	0.1803	0.1778	0.1845	21641
CFGAN CE	0.0135	0.0586	0.0087	0.003	0.0148	20591
CFGAN TE	0.0171	0.007	0.0168	0.0201	0.0169	20541
DCEVAE	0.0050	0.0051	0.0040	0.0043	0.0051	21142

causal structures, the DCEVAE WR that represents the DCVAE architecture (Kim et al., 2021) without any fairness regulation term (i.e., $\beta_f = 0$ according to notations in (Kim et al., 2021)) and the original data. Our approach that contains no fairness penalty on the generated outcomes (in step 1) is also designed to reflect the causal structure. We also analyze the impact of CFGAN CE (Xu et al., 2019), which aims at decreasing the TCE in the generated data, CFGAN CE (Xu et al., 2019) which in turn aims at decreasing the 4 different (CCE), and finally DCEVAE, which corresponds to the DCVAE model with a fairness penalization set to $\beta_f = 0.3$.

As expected, only the three last methods, which act on the data rather than on the predictor itself, are able to mitigate TCE and CCE. Our method does not seek at mitigating biases in inferred outcomes, but seeks at leveraging inferred variables that allow it to learn a fair predictor. This is thus without any surprise that the reconstructed Y are not unbiased with regards to the sensitive; this is even a good indication of no information loss in the step 1 of our process, despite mitigating correlation between the latent confounder U and the sensitive A. In the following, we compare these observations to results in prediction effects.

Predictions Effect In this part, we focus on the level of fairness of the final predictor model. A Logistic Regression (LR), a Neural Network (NN) and a classification tree (CART) are considered in the following. These predictors are either trained on the datasets produced from generation-focused models (i.e., CausalGAN, CFGAN TE, CFGAN CE, DCVAE RW and DCEVAE), or trained in the second step as described in Sect. 3.2 for our two-steps model. Please note that our algorithm can only handle derivative gradient during the optimization, therefore we have discarded the tree CART.

We report the results for each prediction model in Table 6, in terms of TPE, CPE (measured on generated data for test samples) and prediction accuracy (measured on the original test dataset). From this table, we observe completely different results than those from previous table, with generation based models such as CFGAN greatly penalized compared to two-steps methods such as ours. This confirms our intuition that, even if produced data have biases well mitigated on the test set (as seen in Table 5), some small residuals of these biases can stay in the data. Then, the learning process is free to assign important emphasis on these problematic features, if this helps to achieve good prediction accuracy. In two-steps approaches such as ours, this is not the case, since biases of the outcomes are mitigated while learning prediction models, which enables more fairness robustness on test data.

Table 6 Total predictions effect and counterfactual predictions effect on adult UCI

	Total predictions effect (TPE)	Counterfactual predictions effect (CPE)				Accuracy
		o_{00}	o_{01}	o_{10}	o_{11}	
Causal GAN - NN	0.1834	0.1148	0.134	0.1353	0.1965	0.8138
Causal GAN - LR	0.1368	0.0634	0.0985	0.0576	0.1535	0.7997
Causal GAN - CART	0.2204	0.0163	0.112	0.1252	0.2482	0.8082
DCEVAE RW - NN	0.1782	0.1758	0.1768	0.1771	0.1786	0.8133
DCEVAE RW - LR	0.1867	0.1237	0.1866	0.16474	0.1912	0.8040
DCEVAE RW - CART	0.2161	0.0662	0.1726	0.22638	0.22742	0.8119
CFGAN CE - NN	0.1394	0.1312	0.1339	0.0968	0.1463	0.8085
CFGAN CE - LR	0.1486	0.0603	0.1161	0.0597	0.1662	0.8153
CFGAN CE - CART	0.1501	0.101	0.0993	0.1119	0.1612	0.8143
CFGAN TE - NN	0.1415	0.0637	0.1266	0.1059	0.1498	0.8129
CFGAN TE - LR	0.1793	0.0295	0.1603	0.0528	0.2029	0.8116
CFGAN TE - CART	0.1794	0.0244	0.1463	0.0802	0.2004	0.8096
DCEVAE - NN	0.0047	0.027	0.0205	0.0205	0.0021	0.7997
DCEVAE - LR	0.0172	0.0525	0.0169	0.0169	0.0157	0.8019
DCEVAE - CART	0.0297	0.0265	0.0243	0.0243	0.0255	0.7999
Ours - NN	**0.0007**	**0.0044**	**0.0009**	**0.0017**	**0.0014**	**0.8441**
Ours - LR	0.0139	0.0179	0.0175	0.0166	0.013	0.8279

The results in bold represent the best performance achieved for each column

5 Conclusion

We developed a new adversarial learning approach for counterfactual fairness. To the best of our knowledge, this is the first such method that can be applied for continuous sensitive attributes. The method proved to be very efficient for different dependence metrics on various artificial and real-world data sets, for both the discrete and the continuous settings. Finally, our proposal is applicable for any causal graph to achieve generic counterfactual fairness. As future works, it might be interesting to consider a generalization of our proposal for Path Specific (Chiappa, 2019) counterfactual fairness in the continuous case.

References

Adel, T., Valera, I., Ghahramani, Z., & Weller, A. (2019). One-network adversarial fairness. *AAAI'19, 33*, 2412–2420.

Angwin, J., Larson, J., Mattu, S., & Kirchner, L. (2016). Machine bias. *ProPublica*, May 23, 2016.

Bellamy, R.K., Dey, K., Hind, M., Hoffman, S.C., Houde, S., Kannan, K., Lohia, P., Martino, J., Mehta, S., Mojsilovic, A., et al. (2018). Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. arXiv preprint arXiv:1810.01943.

Bowman, S.R., Vilnis, L., Vinyals, O., Dai, A.M., Jozefowicz, R., & Bengio, S. (2015). Generating sentences from a continuous space. arXiv:1511.06349.

Calmon, F.P., Wei, D., Ramamurthy, K.N., & Varshney, K.R. (2017). Optimized data pre-processing for discrimination prevention. arXiv preprint arXiv:1704.03354.

Celis, L.E., Huang, L., Keswani, V., & Vishnoi, N.K. (2019). Classification with fairness constraints: A meta-algorithm with provable guarantees. In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 319–328.
Chen, J., Kallus, N., Mao, X., Svacha, G., & Udell, M. (2019). Fairness under unawareness: Assessing disparity when protected class is unobserved. In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 339–348.
Chen, X., Kingma, D.P., Salimans, T., Duan, Y., Dhariwal, P., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Variational lossy autoencoder. arXiv preprint arXiv:1611.02731.
Chiappa, S. (2019). Path-specific counterfactual fairness. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*, 7801–7808.
Dua, D., & Graff, C. (2017). UCI ML Repository. http://archive.ics.uci.edu/ml.
Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2012). Fairness through awareness. In: *ITCS'12*, pp. 214–226.
Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Networks, 1–9 arXiv:1406.2661.
Grari, V., Ruf, B., Lamprier, S., & Detyniecki, M. (2019). Fair adversarial gradient tree boosting. In: *ICDM'19*, pp. 1060–1065.
Grari, V., Ruf, B., Lamprier, S., & Detyniecki, M. (2019). Fairness-aware neural rényi minimization for continuous features. arXiv:1911.04929.
Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. (2012). A kernel two-sample test. *Journal of Machine Learning Research, 13*(1), 723–773.
Hardt, M., Price, E., & Srebro, N. (2016). Equality of opportunity in supervised learning. In: *Advances in Neural Information Processing Systems*, pp. 3315–3323.
Hinnefeld, J.H., Cooman, P., Mammo, N., & Deese, R. (2018). Evaluating fairness metrics in the presence of dataset bias. arXiv preprint arXiv:1809.09245
Kamiran, F., & Calders, T. (2012). Data preprocessing techniques for classification without discrimination. *Knowledge and Informatoin Systems, 33*(1), 1–33.
Kearns, M., Neel, S., Roth, A., & Wu, Z.S. (2017). Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. arXiv preprint arXiv:1711.05144.
Kilbertus, N., Ball, P.J., Kusner, M.J., Weller, A., & Silva, R. (2020). The sensitivity of counterfactual fairness to unmeasured confounding. In: *Uncertainty in Artificial Intelligence*, pp. 616–626 . PMLR.
Kim, H., Shin, S., Jang, J., Song, K., Joo, W., Kang, W., & Moon, I.-C. (2021). Counterfactual fairness with disentangled causal effect variational autoencoder. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 8128–8136.
Kingma, D.P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.
Kocaoglu, M., Snyder, C., Dimakis, A.G., & Vishwanath, S. (2017). Causalgan: Learning causal implicit generative models with adversarial training. arXiv preprint arXiv:1709.02023.
Kusner, M.J., Loftus, J., Russell, C., & Silva, R. (2017). Counterfactual fairness. In: Advances in Neural Information Processing Systems, pp. 4066–4076.
Louizos, C., Shalit, U., Mooij, J.M., Sontag, D., Zemel, R., & Welling, M. (2017). Causal effect inference with deep latent-variable models. In: *Advances in Neural Information Processing Systems*, pp. 6446–6456.
Louppe, G., Kagan, M., & Cranmer, K. (2017). Learning to pivot with adversarial networks. In: *Advances in Neural Information Processing Systems*, pp. 981–990.
Madras, D., Creager, E., Pitassi, T., & Zemel, R. (2019). Fairness through causal awareness: Learning causal latent-variable models for biased data. In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 349–358.
Makhzani, A., Shlens, J., Jaitly, N., & Goodfellow, I.J. (2015). Adversarial autoencoders. CoRR **abs/1511.05644**arXiv:1511.05644**.**
Mary, J., Calauzènes, C., & Karoui, N.E. (2019). Fairness-aware learning for continuous attributes and treatments. In: *ICML'19*, pp. 4382–4391.
Moro, S., Cortez, P., & Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems, 62*, 22–31.
Pearl, J. (2009). Causal inference in statistics: An overview. *Statistics surveys, 3,* 96–146.
Pfohl, S., Duan, T., Ding, D.Y., & Shah, N.H. (2019). Counterfactual reasoning for fair clinical risk prediction. arXiv preprint arXiv:1907.06260.
Russell, C., Kusner, M.J., Loftus, J., & Silva, R. (2017). When worlds collide: integrating different counterfactual assumptions in fairness. In: *Advances in Neural Information Processing Systems*, pp. 6414–6423.

Shalit, U., Johansson, F.D., & Sontag, D. (2017). Estimating individual treatment effect: generalization bounds and algorithms. In: *ICML'17*, pp. 3076–3085.
Sønderby, C.K., Raiko, T., Maaløe, L., Sønderby, S.K., & Winther, O. (2016). Ladder variational autoencoders. In: *NIPS'16*, pp. 3738–3746.
Team, S. D., Team. (2016). Rstan: The r interface to stan. *R package version, 2*(1), 522.
The Institute of Actuaries of France: Pricing Game 2015. https://freakonometrics.hypotheses.org/20191. Online; accessed 14 August 2019 (2015).
US Census Bureau (2019). US Census Demographic Data. https://data.census.gov/cedsci/. Online; accessed 03 April 2019.
Wadsworth, C., Vera, F., & Piech, C. (2018). Achieving fairness through adversarial learning: An application to recidivism prediction. arXiv:1807.00199.
Xu, D., Wu, Y., Yuan, S., Zhang, L., & Wu, X. (2019). Achieving causal fairness through generative adversarial networks. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*.
Zafar, M.B., Valera, I., Rodriguez, M.G., & Gummadi, K.P. (2015). Fairness constraints: Mechanisms for fair classification. arXiv preprint arXiv:1507.05259.
Zhang, B.H., Lemoine, B., & Mitchell, M. (2018). Mitigating unwanted biases with adversarial learning. In: *AAAI'18*, pp. 335–340.
Zhao, S., Song, J., & Ermon, S. (2017). Infovae: Information maximizing variational autoencoders. arXiv preprint arXiv:1706.02262.