A PROOFS

A.1 PROOFS FOR GSV

Proposition A.1 (**Proposition 2.1**). *The GSV have the following property*

$$\sum_{i=1}^{d} \Phi_i(f, \mathcal{F}, \mathcal{B}) = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{F}}[f(\boldsymbol{x})] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{B}}[f(\boldsymbol{x})]. \tag{11}$$

Proof. As a reminder, we have defined the vector

$$\boldsymbol{\Phi}(f, \mathcal{F}, \mathcal{B}) = \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}\sim\mathcal{F}\\ \boldsymbol{z}\sim\mathcal{B}}}\big[\boldsymbol{\phi}(f, \boldsymbol{x}, \boldsymbol{z})\big], \tag{12}$$

whose components sum up to

$$\begin{aligned}
\sum_{i=1}^{d} \Phi_i(f, \mathcal{F}, \mathcal{B}) &= \sum_{i=1}^{d} \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}\sim\mathcal{F}\\ \boldsymbol{z}\sim\mathcal{B}}}[\,\phi_i(f, \boldsymbol{x}, \boldsymbol{z})\,] && (13)\\
&= \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}\sim\mathcal{F}\\ \boldsymbol{z}\sim\mathcal{B}}}\left[\sum_{i=1}^{d} \phi_i(f, \boldsymbol{x}, \boldsymbol{z})\right] && (14)\\
&= \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}\sim\mathcal{F}\\ \boldsymbol{z}\sim\mathcal{B}}}[\,f(\boldsymbol{x}) - f(\boldsymbol{z})\,] && (15)\\
&= \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{F}}[f(\boldsymbol{x})] - \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathcal{B}}[f(\boldsymbol{z})] && (16)\\
&= \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{F}}[f(\boldsymbol{x})] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{B}}[f(\boldsymbol{x})], && (17)
\end{aligned}$$

where at the last step we have simply renamed a dummy variable. □

Proposition A.2 (**Proposition ??**). *Let S'_0 be **fixed**, and let $\xrightarrow{p}$ represent convergence in probability as the size M of the set $S'_1 \sim \mathcal{B}'^M$ increases, we have*

$$\widehat{\Phi}_s(f, S'_0, S'_1) \xrightarrow{p} \sum_{j=1}^{N_1} \omega_j\, \widehat{\Phi}_s(f, S'_0, \boldsymbol{z}^{(j)}). \tag{18}$$

Proof.

$$\begin{aligned}
\widehat{\boldsymbol{\Phi}}(f, S'_0, S'_1) &= \frac{1}{M^2} \sum_{\boldsymbol{x}^{(i)}\in S'_0} \sum_{\boldsymbol{z}^{(j)}\in S'_1} \boldsymbol{\phi}(f, \boldsymbol{x}^{(i)}, \boldsymbol{z}^{(j)})\\
&= \frac{1}{M} \sum_{\boldsymbol{z}^{(j)}\in S'_1} \left(\frac{1}{M} \sum_{\boldsymbol{x}^{(i)}\in S'_0} \boldsymbol{\phi}(f, \boldsymbol{x}^{(i)}, \boldsymbol{z}^{(j)})\right) \qquad (19)\\
&= \frac{1}{M} \sum_{\boldsymbol{z}^{(j)}\in S'_1} \widehat{\boldsymbol{\Phi}}(f, S'_0, \boldsymbol{z}^{(j)}).
\end{aligned}$$

Since S'_0 is assumed to be fixed, then the only random variable in $\widehat{\Phi}_s(f, S'_0, \boldsymbol{z}^{(j)})$ is $\boldsymbol{z}^{(j)}$ which represents an instance sampled from the $\mathcal{B}'$. Therefore, we can define $\psi(\boldsymbol{z}) := \widehat{\Phi}_s(f, S'_0, \boldsymbol{z})$ and we get

$$\begin{aligned}
\widehat{\Phi}_s(f, S'_0, S'_1) &= \frac{1}{M} \sum_{\boldsymbol{z}^{(j)}\in S'_1} \widehat{\Phi}_s(f, S'_0, \boldsymbol{z}^{(j)})\\
&= \frac{1}{M} \sum_{\boldsymbol{z}^{(j)}\in S'_1} \psi(\boldsymbol{z}^{(j)}) \qquad \text{with } S'_1 \sim \mathcal{B}'^M. \qquad (20)
\end{aligned}$$

By the weak law of large number, the following holds as M goes to infinity (Wasserman, 2004, Theorem 5.6)

$$\frac{1}{M}\sum_{\boldsymbol{z}^{(j)}\in S_1'}\psi(\boldsymbol{z}^{(j)}) \xrightarrow{p} \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathcal{B}'}[\psi(\boldsymbol{z})]. \tag{21}$$

Now, as a reminder, the manipulated background distribution is $\mathcal{B}' := \mathcal{C}(D_1, \boldsymbol{\omega})$ with $\boldsymbol{\omega} \neq \mathbf{1}/N_1$. Therefore

$$\begin{aligned}\widehat{\Phi}_s(f, S_0', S_1') &\xrightarrow{p} \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathcal{B}'}[\psi(\boldsymbol{z})] \\ &= \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathcal{C}(D_1,\boldsymbol{\omega})}[\psi(\boldsymbol{z})] \\ &= \sum_{j=1}^{N_1}\omega_j\psi(\boldsymbol{z}^{(j)}) \\ &= \sum_{j=1}^{N_1}\omega_j\widehat{\Phi}_s(f, S_0', \boldsymbol{z}^{(j)})\end{aligned} \tag{22}$$

concluding the proof. □

A.2 Proofs for Optimization Problem

A.2.1 Technical Lemmas

We provide some technical lemmas that will be essential when proving Theorem A.1. These lemmas and proofs are provided here for completeness and are not meant as contributions by the authors.

Let us first write the formal definition of the minimum of a function.

Definition A.1 (Minimum)**.** *Given some function $f : D \to \mathbb{R}$, the minimum of f over D (denoted $f^\star$) is defined as follows:*

$$f^\star = \min_{x\in D} f(x) \iff \exists x^\star \in D \text{ s.t. } f^\star = f(x^\star) \leqslant f(x) \quad \forall x \in D.$$

Basically, the notion of minimum coincides with the notion of infimum (highest lower bound) of $f(D)$ when this lower bound is attained for some $x^\star \in D$. For the rest of this appendix, we shall only study constrained optimization problems where points from the feasible set $D = \{(x, y) : x \in \mathcal{X}, y \in \mathcal{Y}_x \subset \mathcal{Y}\}$ can be *selected* by the following procedure

1. Choose some $x \in \mathcal{X}$
2. Given the selected x, choose some $y \in \mathcal{Y}_x \subset \mathcal{Y}$ where the set $\mathcal{Y}_x$ is non-empty and depends on the value of x.

When optimizing objective functions over these types of domains, one can optimize in two steps as highlighted in the following lemma.

Lemma A.1. *Given a feasible set D of the form described above and an objective function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, the following holds*

$$\min_{(x,y)\in D} f(x, y) = \min_{x\in\mathcal{X}} \min_{y\in\mathcal{Y}_x} f(x, y).$$

Proof. Let $\widetilde{f}(x) := \min_{y\in\mathcal{Y}_x} f(x, y)$, which is a well defined function on $\mathcal{X}$ since $\mathcal{Y}_x$ is non-empty for any $x \in \mathcal{X}$. By the definition of the minimum, we have

$$\forall x \in \mathcal{X},\ \exists y^\star(x) \in \mathcal{Y}_x \quad \text{s.t.} \quad \widetilde{f}(x) = f(x, y^\star(x)) \leqslant f(x, y) \quad \forall y \in \mathcal{Y}_x. \tag{23}$$

Now, we can optimize $\widetilde{f}$ with respect to x i.e. $f^\star = \min_{x\in\mathcal{X}} \widetilde{f}(x)$. By applying once again the definition of the minimum, we get

$$\exists x^\star \in \mathcal{X} \quad \text{s.t.} \quad f^\star = \widetilde{f}(x^\star) \leqslant \widetilde{f}(x) \quad \forall x \in \mathcal{X}. \tag{24}$$

By virtue of Equation 23, we have that $\widetilde{f}(x^\star) = f(x^\star, y^\star(x^\star)) = f(x^\star, y^\star)$, where we labeled $y^\star := y^\star(x^\star)$ for convenience. We get

$$\begin{aligned} \exists (x^\star, y^\star) \in D \quad \text{s.t.} \quad f(x^\star, y^\star) &\leqslant f(x, y^\star(x)) \quad \forall\, x \in \mathcal{X} && \text{(cf. Equation 24)} \\ &\leqslant f(x, y) \qquad\quad \forall\, y \in \mathcal{Y}_x. && \text{(cf. Equation 23)} \end{aligned}$$

Hence we have proven that $\exists (x^\star, y^\star) \in D \quad \text{s.t.} \quad f(x^\star, y^\star) \leqslant f(x, y) \;\; \forall (x, y) \in D$, which concludes the proof. □

Lemma A.2. *Given a feasible set D of the form described above and two functions $h : \mathcal{X} \to \mathbb{R}$ and $g : \mathcal{Y} \to \mathbb{R}$, then*

$$\min_{(x,y)\in D} \Big(h(x) + g(y) \Big) = \min_{x \in \mathcal{X}} \Big(h(x) + \min_{y \in \mathcal{Y}_x} g(y) \Big)$$

Proof. Applying Lemma A.1 with the function $f(x, y) := h(x) + g(y)$ leads to the desired result. □

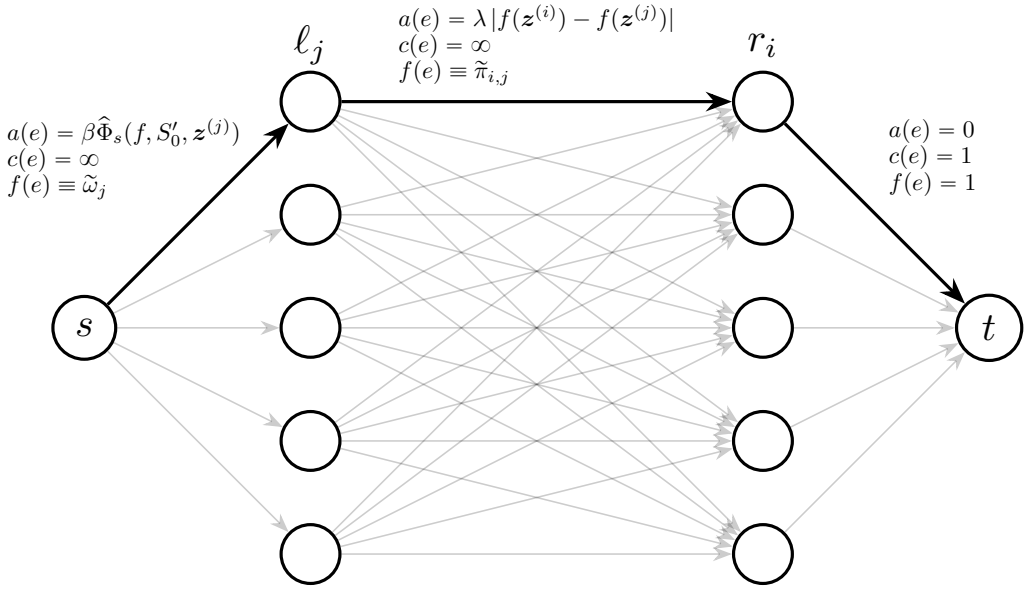

Figure 6: Graph $\mathbb{G}$ on which we solve the MCF. Note that the total amount of flow is $d = N_1$ and there are N_1 left and right nodes ℓ_j, r_i.

A.2.2 Minimum Cost Flows

Let $\mathbb{G} = (\mathcal{V}, \mathcal{E})$ be a graph with vertices $v \in \mathcal{V}$ with directed edges $e \in \mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, $c : \mathcal{E} \to \mathbb{R}^+$ be a capacity and $a : \mathcal{E} \to \mathbb{R}$ be a cost. Moreover, let $s, t \in \mathcal{E}$ be two special vertices called the source and the sink respectivelly, and $d \in \mathbb{R}^+$ be a total flow. The Minimum-Cost Flow (MCF) problem of $\mathbb{G}$ consists of finding the flow function $f : \mathcal{E} \to \mathbb{R}^+$ that minimizes the total cost

$$\begin{aligned} \min_f \quad & \sum_{e \in \mathcal{E}} a(e) f(e) \\ \text{s.t.} \quad & 0 \leqslant f(e) \leqslant c(e) \;\; \forall e \in \mathcal{E} \\ & \sum_{e \in u^+} f(e) - \sum_{e \in u^-} f(e) = \begin{cases} 0 & u \in \mathcal{V} \setminus \{s, t\} \\ d & u = s \\ -d & u = t \end{cases} \end{aligned} \tag{25}$$

where $u^+ := \{(u, v) \in \mathcal{E}\}$ and $u^- := \{(v, u) \in \mathcal{E}\}$ are the outgoing and incoming edges from u. The terminology of *flow* arises from the constraint that, for vertices that are not the source nor the sink, the outgoing flow must equal the incoming one, which is reminiscent of conservation laws in fluidic. We shall refer to $f((u, v))$ as the flow from u to v.

Now that we have introduced minimum cost flows, let us specify the graph that will be employed to manipulate GSV, see Figure 6. We label the flow going from the sink s to one of the left vertices as $\widetilde{\omega}_i \equiv \omega_i \times N_1$, and the flow going from ℓ_j to r_i as $\widetilde{\pi}_{i,j} \equiv \pi_{i,j} \times N_1$. The required flow is fixed at $d = N_1$.

Theorem A.1. *Solving the MCF of Figure 6 leads to a solution of the linear program in Algorithm 1.*

Proof. We begin by showing that the flow conservation constraints in the MCF imply that π is a coupling measure (i.e. $\pi \in \Delta(\mathcal{B}, \mathcal{B}')$), and $\boldsymbol{\omega}$ is constrained to the probability simplex $\Delta(N_1)$. Applying the conservation law on the left-side of the graph leads to the conclusion that the flows entering vertices ℓ_j must sum up to N_1

$$\sum_{j=1}^{N_1} \widetilde{\omega}_j = N_1.$$

This implies that $\boldsymbol{\omega}$ is must be part of the probability simplex. By conservation, the amount of flow that leaves a specific vertex ℓ_j must also be $\widetilde{\omega}_j$, hence

$$\sum_i \widetilde{\pi}_{ij} = \widetilde{\omega}_j.$$

For any edge outgoing from r_i to the sink t, the flow must be exactly 1. This is because we have N_1 edges with capacity $c(e) = 1$ going into the sink and the sink must receive an incoming flow of N_1. As a consequence of the conservation law on a specific vertex r_i, the amount of flow that goes into each r_i is also 1

$$\sum_j \widetilde{\pi}_{ij} = 1.$$

Putting everything together, from the conservation laws on $\mathbb{G}$, we have that $\boldsymbol{\omega} \in \Delta(N_1)$, and $\pi \in \Delta(\mathcal{B}, \mathcal{B}')$.

Now, to make the parallel between the MCF and Algorithm 1, we must use Lemma A.2. As a reminder, the Lemma states that for specific types of domains, one can solve the constrained optimization problem in two optimization steps. Note that $\boldsymbol{\omega}$ is restricted to the probability simplex, while π is restricted to be a coupling measure. Importantly, the set of all possible coupling measures $\Delta(\mathcal{B}, \mathcal{B}')$ is different for each $\boldsymbol{\omega}$ (and non-empty) because $\mathcal{B}'$ depends on $\boldsymbol{\omega}$. Hence, we study a feasible set with the same structure as the ones tackled in the Lemma A.2 (where $x \in \mathcal{X}$ becomes $\boldsymbol{\omega} \in \Delta(N_1)$) and $y \in \mathcal{Y}_x$ becomes $\pi \in \Delta(\mathcal{B}, \mathcal{B}')$) and we can apply the Lemma A.2 to the objective function of the MCF.

$$\begin{aligned}
\min_f \sum_{e \in \mathcal{E}} f(e)a(e) &= \min_{\widetilde{\omega}, \widetilde{\pi}} \sum_{j=1}^{N_1} \beta \widetilde{\omega}_j \widehat{\Phi}_s(f, S'_0, \boldsymbol{z}^{(j)}) + \lambda \sum_{i,j} \widetilde{\pi}_{ij} |f(\boldsymbol{z}^{(i)}) - f(\boldsymbol{z}^{(j)})| \\
&= \min_{\widetilde{\omega}, \widetilde{\pi}} \frac{N_1}{N_1} \left(\beta \sum_{j=1}^{N_1} \widetilde{\omega}_j \widehat{\Phi}_s(f, S'_0, \boldsymbol{z}^{(j)}) + \lambda \sum_{i,j} \widetilde{\pi}_{ij} |f(\boldsymbol{z}^{(i)}) - f(\boldsymbol{z}^{(j)})| \right) \\
&= N_1 \min_{\widetilde{\omega}, \widetilde{\pi}} \left(\beta \sum_{j=1}^{N_1} \frac{\widetilde{\omega}_j}{N_1} \widehat{\Phi}_s(f, S'_0, \boldsymbol{z}^{(j)}) + \lambda \sum_{i,j} \frac{\widetilde{\pi}_{ij}}{N_1} |f(\boldsymbol{z}^{(i)}) - f(\boldsymbol{z}^{(j)})| \right) \\
&= N_1 \min_{\boldsymbol{\omega} \in \Delta(N_1), \pi \in \Delta(\mathcal{B}, \mathcal{B}')} \left(\beta \sum_{j=1}^{N_1} \omega_j \widehat{\Phi}_s(f, S'_0, \boldsymbol{z}^{(j)}) + \lambda \sum_{i,j} \pi_{i,j} |f(\boldsymbol{z}^{(i)}) - f(\boldsymbol{z}^{(j)})| \right) \\
&= N_1 \min_{\boldsymbol{\omega} \in \Delta(N_1), \pi \in \Delta(\mathcal{B}, \mathcal{B}')} \Big(h(\omega) + g(\pi) \Big) \\
&= N_1 \min_{\boldsymbol{\omega} \in \Delta(N_1)} \Big(h(\omega) + \min_{\pi \in \Delta(\mathcal{B}, \mathcal{B}')} g(\pi) \Big) \qquad \text{(cf Lemma A.2)} \\
&= N_1 \min_{\boldsymbol{\omega} \in \Delta(N_1)} \left(\beta \sum_{j=1}^{N_1} \omega_j \widehat{\Phi}_s(f, S'_0, \boldsymbol{z}^{(j)}) + \lambda \min_{\pi \in \Delta(\mathcal{B}, \mathcal{B}')} \sum_{i,j} \pi_{i,j} |f(\boldsymbol{z}^{(i)}) - f(\boldsymbol{z}^{(j)})| \right) \\
&= N_1 \min_{\boldsymbol{\omega} \in \Delta(N_1)} \left(\beta \sum_{j=1}^{N_1} \omega_j \widehat{\Phi}_s(f, S'_0, \boldsymbol{z}^{(j)}) + \lambda \, \mathcal{W}(\mathcal{B}, \mathcal{B}'_{\boldsymbol{\omega}}) \right)
\end{aligned}$$

which (up to a multiplicative constant N_1) is a solution of the linear program of Algorithm 1. □

B SHAPLEY VALUES

B.1 LOCAL SHAPLEY VALUES

We introduce Local Shapley Values (LSV) more formally. First of, as explained earlier, Shapley values are based on coalitional game theory where the different features work together toward a common outcome $f(\boldsymbol{x})$. In a game, the features can either be present or absent, which is simulated by replacing some features by a baseline value $\boldsymbol{z}$.

Definition B.1 (Replace Function)**.** *Given an input of interest $\boldsymbol{x}$, a subset of features $S \subseteq \{1, 2, \dots, d\}$ that are considered active, and a baseline input $\boldsymbol{z}$, the replace-function $\boldsymbol{r}_S : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is defined as*

$$r_S(\boldsymbol{z}, \boldsymbol{x})_i = \begin{cases} x_i & \text{if } i \in S \\ z_i & \text{otherwise.} \end{cases} \tag{26}$$

We note that this function is meant to "activate" the features in S.

Now, if we let π be a random permutation of d features, and π_i denote all features that appear before i in π, the LSV are computed via

$$\phi_i(f, \boldsymbol{x}, \boldsymbol{z}) := \mathop{\mathbb{E}}_{\pi \sim \Omega} \big[f(\, \boldsymbol{r}_{\pi_i \cup \{i\}}(\boldsymbol{z}, \boldsymbol{x})\,) - f(\, \boldsymbol{r}_{\pi_i}(\boldsymbol{z}, \boldsymbol{x})\,) \big], \quad i = 1, 2, \dots, d, \tag{27}$$

where Ω is the uniform distribution over 2^d permutations. Observe that the computation of LSV is exponential w.r.t the number of features d hence model-agnostic computations are only possible with datasets with few features such as COMPAS and Adult-Income. For datasets with larger amounts of features the `TreeExplainer` algorithm (Lundberg et al., 2020) can be used to compute the LSV (cf. Equation 27) in polynomial time given that one is explaining a tree-based model.

B.2 CONVERGENCE

As a reminder, we are interested in estimating the GSV $\boldsymbol{\Phi} \equiv \boldsymbol{\Phi}(f, \mathcal{F}, \mathcal{B})$ which requires estimating expectations w.r.t the foreground and background distributions. Said estimations can be conducted with Monte-Carlo where we sample M instances

$$S_0 \sim \mathcal{F}^M \qquad S_1 \sim \mathcal{B}^M, \tag{28}$$

and compute the plug-in estimates

$$\begin{aligned} \widehat{\boldsymbol{\Phi}}(f, S_0, S_1) &:= \boldsymbol{\Phi}(f,\, \mathcal{C}(S_0, \mathbf{1}/M),\, \mathcal{C}(S_1, \mathbf{1}/M)) \\ &= \frac{1}{M^2} \sum_{\boldsymbol{x}^{(i)} \in S_0} \sum_{\boldsymbol{z}^{(j)} \in S_1} \boldsymbol{\phi}(f, \boldsymbol{x}^{(i)}, \boldsymbol{z}^{(j)}). \end{aligned} \tag{29}$$

We now show that, $\widehat{\boldsymbol{\Phi}}(f, S_0, S_1)$ is a consistent and asymptotically normal estimate of $\boldsymbol{\Phi}(f, \mathcal{F}, \mathcal{B})$

Proposition B.1. *Let $f : \mathcal{X} \to [0, 1]$ be a black box, $\mathcal{F}$ and $\mathcal{B}$ be distributions on $\mathcal{X}$, and $\widehat{\boldsymbol{\Phi}} \equiv \widehat{\boldsymbol{\Phi}}(f, S_0, S_1)$ be the plug-in estimate of $\boldsymbol{\Phi} \equiv \boldsymbol{\Phi}(f, \mathcal{F}, \mathcal{B})$, the following holds for any $\delta \in \,]0, 1[$ and $k = 1, 2 \dots, d$*

$$\lim_{M \to \infty} \mathbb{P}\left(|\widehat{\Phi}_k - \Phi_k| \geqslant \frac{F^{-1}_{\mathcal{N}(0,1)}(1 - \delta/2)}{2\sqrt{M}} \sqrt{\sigma^2_{10} + \sigma^2_{01}} \right) = \delta,$$

where $F^{-1}_{\mathcal{N}(0,1)}$ is the inverse Cumulative Distribution Function (CDF) of the standard normal distribution, $\sigma^2_{10} = \mathbb{V}_{\boldsymbol{x} \sim \mathcal{F}}[\, \mathbb{E}_{\boldsymbol{z} \sim \mathcal{B}}[\phi_i(f, \boldsymbol{x}, \boldsymbol{z}))]\,]$ and $\sigma^2_{01} = \mathbb{V}_{\boldsymbol{z} \sim \mathcal{B}}[\, \mathbb{E}_{\boldsymbol{x} \sim \mathcal{F}}[\phi_i(f, \boldsymbol{x}, \boldsymbol{z}))]\,]$.

Proof. The proof consists simply in noting that LSV $\phi_k(f, \boldsymbol{x}^{(i)}, \boldsymbol{z}^{(j)})$ are a function of two independent samples $\boldsymbol{x}^{(i)} \sim \mathcal{F}$ and $\boldsymbol{z}^{(j)} \sim \mathcal{B}$. The model f is assumed fixed and hence for any feature k we can define $h(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(j)}) := \phi_k(f, \boldsymbol{x}^{(i)}, \boldsymbol{z}^{(j)})$. Now, the estimates of GSV can be rewritten

$$\widehat{\Phi}_k(f, S_0, S_1) = \frac{1}{|S_0|\,|S_1|} \sum_{\boldsymbol{x}^{(i)} \in S_0} \sum_{\boldsymbol{z}^{(j)} \in S_1} h(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(j)}), \tag{30}$$

which we recognize as a well-known class of statistics called two-samples U-statistics. Such statistics are unbiased and asymptotically normal estimates of

$$\Phi_k(f, \mathcal{F}, \mathcal{B}) = \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}\sim\mathcal{F}\\ \boldsymbol{z}\sim\mathcal{B}}}[h(\boldsymbol{x}, \boldsymbol{z})]. \tag{31}$$

The asymptotic normality of two-samples U-statistics is characterized by the following Theorem (Lee, 2019, Section 3.7.1).

Theorem B.1. *Let $\widehat{\Phi}_k \equiv \widehat{\Phi}_k(f, S_0, S_1)$ be a two-samples U-statistic with $|S_0| = N, |S_1| = M$, moreover let $h(\boldsymbol{x}, \boldsymbol{z})$ have finite first and second moments, then the following holds for any $\delta \in]0, 1[$*

$$\lim_{\substack{N+M\to\infty\\ \text{s.t. } N/(N+M)\to p\in(0,1)}} \mathbb{P}\left(|\widehat{\Phi}_k - \Phi_k| \geqslant \frac{F^{-1}_{\mathcal{N}(0,1)}(1-\delta/2)}{\sqrt{M+N}} \sqrt{\frac{\sigma^2_{10}}{p} + \frac{\sigma^2_{01}}{1-p}} \right) = \delta,$$

where $\sigma^2_{10} = \mathbb{V}_{\boldsymbol{x}\sim\mathcal{F}}[\,\mathbb{E}_{\boldsymbol{z}\sim\mathcal{B}}[h(\boldsymbol{x}, \boldsymbol{z})]\,]$ and $\sigma^2_{01} = \mathbb{V}_{\boldsymbol{z}\sim\mathcal{B}}[\,\mathbb{E}_{\boldsymbol{x}\sim\mathcal{F}}[h(\boldsymbol{x}, \boldsymbol{z})]\,]$.

Proposition B.1 follows from this Theorem by choosing $N = M, p = 0.5$ and noticing that having a model with bounded outputs ($f : \mathcal{X} \to [0, 1]$) implies that $|h(\boldsymbol{x}, \boldsymbol{z})| \leqslant 1\ \forall \boldsymbol{x}, \boldsymbol{z} \in \mathcal{X}$ which means that $h(\boldsymbol{x}, \boldsymbol{z})$ has bounded first and second moments. □

B.3 Compute the LSV

Running Algorithm 1 requires computing the coefficients $\widehat{\Phi}_s(f, S'_0, \boldsymbol{z}^{(j)})$ for $j = 1, 2, \ldots, N_1$. To compute them, first note that they can be written in terms of LSV for all instances in S'_0

$$\widehat{\Phi}_s(f, S'_0, \boldsymbol{z}^{(j)}) = \frac{1}{M} \sum_{\boldsymbol{x}^{(i)}\in S'_0} \phi_s(f, \boldsymbol{x}^{(i)}, \boldsymbol{z}^{(j)}). \tag{32}$$

The LSV $\phi_s(f, \boldsymbol{x}^{(i)}, \boldsymbol{z}^{(j)})$ are computed deeply in the `SHAP` code and are not directly accessible using the current API. Hence, we had to access them using Monkey-Patching *i.e.* we modified the `ExactExplainer` class so that it stores the LSV as one of its attributes. The attribute can then be accessed as seen in Figure 7. The code is provided as a fork the `SHAP` repository. For the `TreeExplainer`, because its source code is in C++ and wrapped in Python, we found it simpler to simply rewrite our own version of the algorithm in C++ so that it directly returns the LSV, instead of Monkey-Patching the `TreeExplainer`.

```
# Mask features using the whole background distribution
mask = Independent(D_1, max_samples=len(D_1))
explainer = shap.explainers.Exact(model.predict_proba, mask)
# Explain all instances sampled from the foreground
explainer(S_0)
# The LSV are extracted with Monkey-Patching
LSV = explainer.LSV         # LSV.shape = (n_features, |S_0|, |D_1|)
Phi_S_0_zj = LSV.mean(1).T  # Phi_S_0_zj.shape = (|D_1|, n_features)
```

Figure 7: How we extract the LSV from the `ExactExplainer` via Monkey-Patching.

C STATISTICAL TESTS

C.1 KS TEST

A first test that can be conducted is a two-samples Kolmogorov-Smirnov (KS) test (Massey Jr, 1951). If we let

$$\widehat{F}_S(x) = \frac{1}{|S|} \sum_{z \in S} \mathbb{1}(z \leqslant x) \tag{33}$$

be the empirical CDF of observations in the set S. Given two sets S and S', the KS statistic is

$$\text{KS}(S, S') = \sup_{x \in \mathbb{R}} |\widehat{F}_S(x) - \widehat{F}_{S'}(x)|. \tag{34}$$

Under the null-hypothesis $H_0 : S \sim \mathcal{D}^{|S|}, S' \sim \mathcal{D}^{|S'|}$ for some univariate distribution $\mathcal{D}$, this statistic is expected to not be too large with high probability. Hence, when the company provides the subsets S'_0, S'_1, the audit can sample their own two subsets $f(S_0), f(S_1)$ uniformly at random from $f(D_0), f(D_1)$ and compute the statistics $\text{KS}(f(S_1), f(S'_1))$ and $\text{KS}(f(S_0), f(S'_0))$ to detect a fraud.

C.2 WALD TEST

An alternative is the Wald test, which is based on the central limit theorem. If $S_1 \sim \mathcal{B}^M$, then the empirical average of the model output over S_1 is asymptotically normally distributed as M increases i.e.

$$\text{Wald}(f(S_1), f(\mathcal{B})) := \frac{\frac{1}{M} \sum_{z \in f(S_1)} z - \mu}{\sigma / \sqrt{M}} \rightsquigarrow \mathcal{N}(0, 1), \tag{35}$$

where $\mu := \mathbb{E}_{z \sim f(\mathcal{B})}[z]$ and $\sigma^2 := \mathbb{V}_{z \sim f(\mathcal{B})}[z]$ are the expected value and variance of the model output across the whole background. The same reasoning holds for S_0 and the foreground $\mathcal{F}$. Applying the Wald test with significance α would detect a fraud when

$$|\,\text{Wald}(f(S'_1), f(\mathcal{B}))\,| > F^{-1}_{\mathcal{N}(0,1)}(1 - \alpha/2), \tag{36}$$

where $F^{-1}_{\mathcal{N}(0,1)}$ is the inverse of the CDF of a standard normal variable.

D METHODOLOGICAL DETAILS

D.1 TOY EXAMPLE

The toy dataset was constructed to closely match the results of the following empirical study comparing skeletal mass distributions between men and women (Janssen et al., 2000). First of, the sex feature was sampled from a Bernoulli

$$S \sim \text{Bernoulli}(0.5). \tag{37}$$

According to the Table 1 of Janssen et al. (2000), the average height of women participants was 163 cm while it was 177cm for men. Both height distributions had the same standard deviation of 7cm. Hence we sampled height via

$$\begin{aligned} H|S = \texttt{man} &\sim \mathcal{N}(177, 49) \\ H|S = \texttt{woman} &\sim \mathcal{N}(163, 49) \end{aligned} \tag{38}$$

It was noted in Janssen et al. (2000) that there was approximately a linear relationship between height and skeletal muscle mass for both sexes. Therefore, we computed the muscle mass M as

$$\begin{aligned} &M|\{H = h, S = \text{man}\} = 0.186h + 5\epsilon \\ &M|\{H = h, S = \text{woman}\} = 0.128h + 4\epsilon \\ &\text{with } \epsilon \sim \mathcal{N}(0, 1) \end{aligned} \tag{39}$$

The values of coefficients 0.186, 0.128 and noise levels 5 and 4 were chosen so the distributions of $M|S$ would approximately match that of Table 1 in Janssen et al. (2000). Finally the target was chosen following

$$\begin{aligned} &Y|\{H = h, M = m\} \sim \text{Bernoulli}(\, P(H, M)\,) \\ &\text{with } P(H, M) = \big[1 + \exp\{100 \times \mathbb{1}(H < 160) - 0.3(M - 28)\}\big]^{-1}. \end{aligned} \tag{40}$$

Simply put, the chances of being hired in the past (Y) were impossible for individuals with a smaller height than 160cm. Moreover, individuals with a higher mass skeletal mass were given more chances to be admitted. Yet, individuals with less muscle mass could still be given the job if they displayed sufficient determination. In the end we generated 6000 samples leading to the following disparity in Y

$$\mathbb{P}(Y = 1|S = \text{man}) = 0.733 \qquad \mathbb{P}(Y = 1|S = \text{woman}) = 0.110. \tag{41}$$

Table 2: Models Test Accuracy % (mean $\pm$ stddev).

	mlp	rf	xgb
COMPAS	68.2 ± 0.9	67.7 ± 0.8	68.6 ± 0.8
Adult	85.6 ± 0.3	86.3 ± 0.2	87.1 ± 0.1
Marketing		91.1 ± 0.1	91.4 ± 0.3
Communities		83 ± 2	82 ± 2

Table 3: Models Demographic Parity (mean $\pm$ stddev).

	mlp	rf	xgb
COMPAS	-0.12 ± 0.01	-0.11 ± 0.01	-0.11 ± 0.02
Adult	-0.20 ± 0.01	-0.19 ± 0.01	-0.192 ± 0.004
Marketing		-0.104 ± 0.005	-0.11 ± 0.01
Communities		-0.50 ± 0.01	-0.54 ± 0.02

D.2 Real Data

The datasets were first divided into train/test subsets with ratio $\frac{4}{5} : \frac{1}{5}$. The models were trained on the training set and evaluated on the test set. All categorical features for COMPAS, Adult, and Marketing were one-hot-encoded which resulted in a total of 11, 40, and 61 columns for each dataset respectively. A simple 50 steps random search was conducted to fine-tune the hyper-parameters with cross-validation on the training set. The resulting test set performance and demographic parities for all models and datasets, aggregated over 5 random data splits, are reported in Tables 2 and 3 respectively.

E ADDITIONAL RESULTS

E.1 MULTIPLE SENSITIVE ATTRIBUTES

We present preliminary results for settings where one wishes to manipulate the Shapley values of multiple sensitive features s each part of a set $s \in \mathcal{S}$. For example, in our experiments we considered `gender` as a sensitive attribute for the Adult-Income dataset and we showed that one can diminish the attribution of this feature. Nonetheless, there are two other features in Adult-Income that share information with this `gender`: `relationship` and `marital-status`. Indeed, `relationship` can take the value `widowed` and `marital-status` can take the value `wife`, which are both proxies of `gender=female`. For this reason, these two other features may be considered sensitive and decision-making that relies strongly on them may not be acceptable. Henceforth, we must derive a method that reduces the total attributions of the features in $\mathcal{S} = \{\texttt{gender}, \texttt{relationship}, \texttt{marital-status}\}$.

We first let $\beta_s := \text{sign}[\,\widehat{\Phi}_s(f, S_0', D_1)\,]$ for any $s \in \mathcal{S}$. In our experiments, all these signs will typically be negative. The proposed approach is to minimize the ℓ_1 norm

$$\|(\widehat{\Phi}_s(f, S_0', S_1'))_{s\in\mathcal{S}}\|_1 := \sum_{s\in\mathcal{S}} |\,\widehat{\Phi}_s(f, S_0', S_1')\,|, \tag{42}$$

which we interpret as the total amount of disparity we can attribute to the sensitive attributes. Remember that $\widehat{\Phi}_s(f, S_0', S_1')$ converges in probability to $\sum_{\boldsymbol{z}^{(j)}\in D_1} \omega_j\, \widehat{\Phi}_s(f, S_0', \boldsymbol{z}^{(j)})$ (cf. Proposition **??**). Therefore minimizing the ℓ_1 norm will require minimizing

$$\sum_{s\in\mathcal{S}} \beta_s \sum_{\boldsymbol{z}^{(j)}\in D_1} \omega_j\, \widehat{\Phi}_s(f, S_0', \boldsymbol{z}^{(j)}) = \sum_{\boldsymbol{z}^{(j)}\in D_1} \omega_j \sum_{s\in\mathcal{S}} \beta_s\, \widehat{\Phi}_s(f, S_0', \boldsymbol{z}^{(j)}), \tag{43}$$

which is again a linear function of the weights. We present Algorithm 4 as an overload of Algorithm 1 that now supports taking multiple sensitive attributes as inputs.

Algorithm 4 Compute non-uniform weights for multiple sensitive attributes $s \in \mathcal{S}$

procedure COMPUTE_WEIGHTS($D_1, \{\widehat{\Phi}_s(f, S_0', \boldsymbol{z}^{(j)})\}_{s,j}, \lambda$)
  $\beta_s := \text{sign}[\sum_{\boldsymbol{z}^{(j)}\in D_1} \widehat{\Phi}_s(f, S_0', \boldsymbol{z}^{(j)})\,] \quad \forall s \in \mathcal{S}$;
  $\mathcal{B} := \mathcal{C}(D_1, \mathbf{1}/N_1)$ ▷ Unmanipulated background
  $\mathcal{B}'_{\boldsymbol{\omega}} := \mathcal{C}(D_1, \boldsymbol{\omega})$ ▷ Manipulated background as a function of $\boldsymbol{\omega}$
  $\boldsymbol{\omega} = \arg\min_{\boldsymbol{\omega}} \;\sum_{\boldsymbol{z}^{(j)}\in D_1} \omega_j \sum_{s\in\mathcal{S}} \beta_s\, \widehat{\Phi}_s(f, S_0', \boldsymbol{z}^{(j)}) + \lambda \mathcal{W}(\mathcal{B}, \mathcal{B}'_{\boldsymbol{\omega}})$
  return $\boldsymbol{\omega}$;

The only difference in the resulting MCF is that we must use the cost $a(e) = \sum_{s\in\mathcal{S}} \beta_s\, \widehat{\Phi}_s(f, S_0', \boldsymbol{z}^{(j)})$ for edges (s, ℓ_j) in the graph $\mathbb{G}$ of Figure 6. This new algorithm is guaranteed to diminish the ℓ_1 norm of the attributions of all sensitive features. However, that this does not imply that all sensitive attributes will diminish in amplitude. Indeed, minimizing the sum of multiple quantities does not guarantee that each quantity will diminish. For example, $4 + 7$ is smaller than $6 + 6$ although 4 is smaller than 6 and 7 is higher than 6. Still, we see reducing the ℓ_1 norm as a natural way to hide the total amount of disparity that is attributable to the sensitive features. Another important methodological change is the way we select the optimal hyper-parameter λ in Algorithm 3. Now at line 13, we use the ℓ_1 norm $\sum_{s\in\mathcal{S}} |\sum_{\boldsymbol{z}^{(j)}\in D_1} \omega_j\, \widehat{\Phi}_s(f, S_0', \boldsymbol{z}^{(j)})|$ as a selection criterion.

Figures 8 and 9 present preliminary results of attacks on three RFs/XGBs fitted on Adults with different train/test splits. We note that in all cases, before the attack, the three sensitive features had large negative attributions. By applying our method, we can considerably reduce the amplitude of the two sensitive attributes. The attribution of the remaining sensitive feature remains approximately constant or slightly becomes more negative. We leave it as future work to run large scale experiments with multiple sensitive features for various datasets.

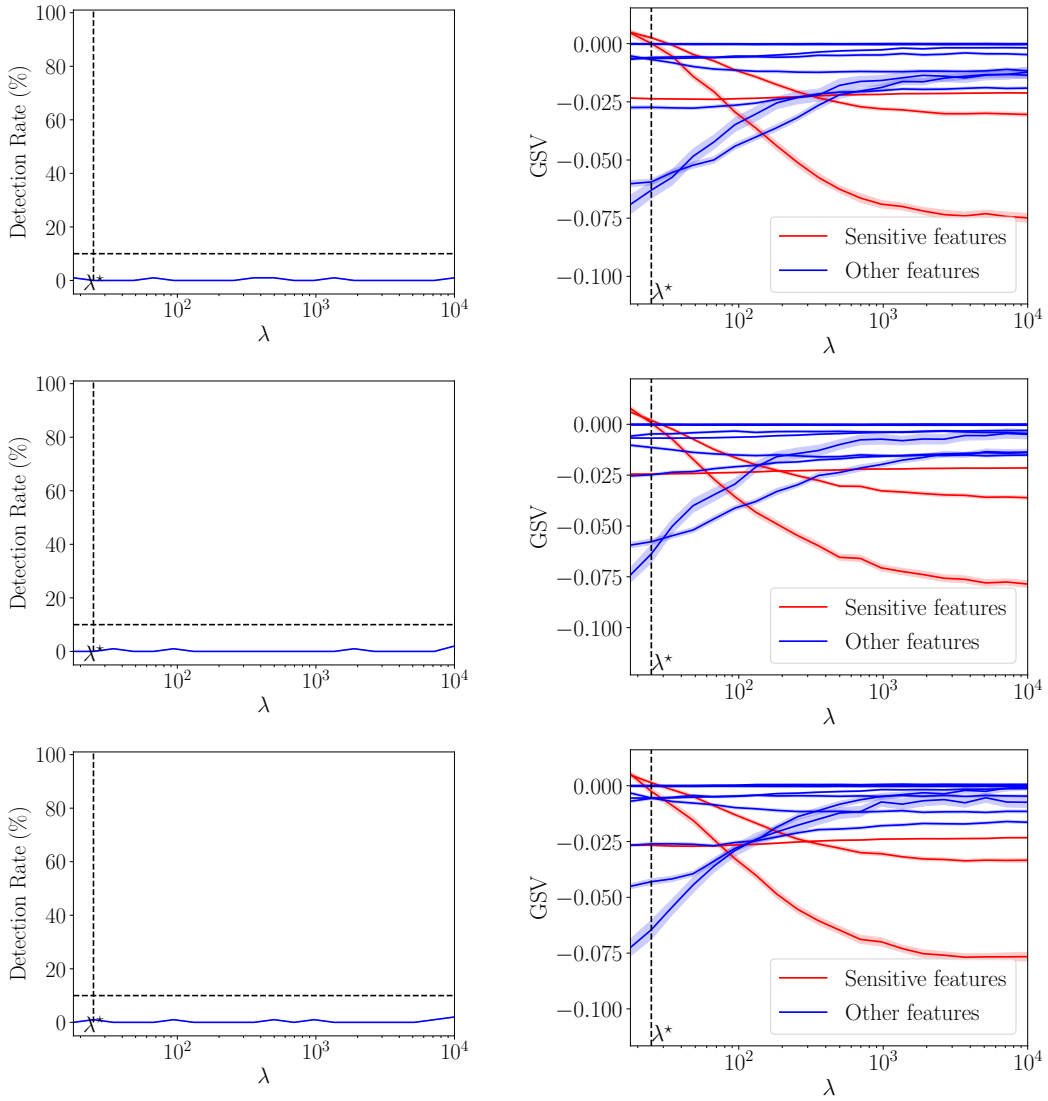

Figure 8: Example of log-space search over values of λ using RFs classifier fitted on Adults and three sensitive attributes. Each row is a different train/test split seed. (Left) The detection rate as a function of the parameter λ of the attack. (Right) For each value of λ, the vertical slice of the 11 curves is the GSV obtained with the resulting $\mathcal{B}'_\omega$. The goal here is to reduce the amplitude all sensitive features (red curves) in order to hide their contribution to the disparity in model outcomes.

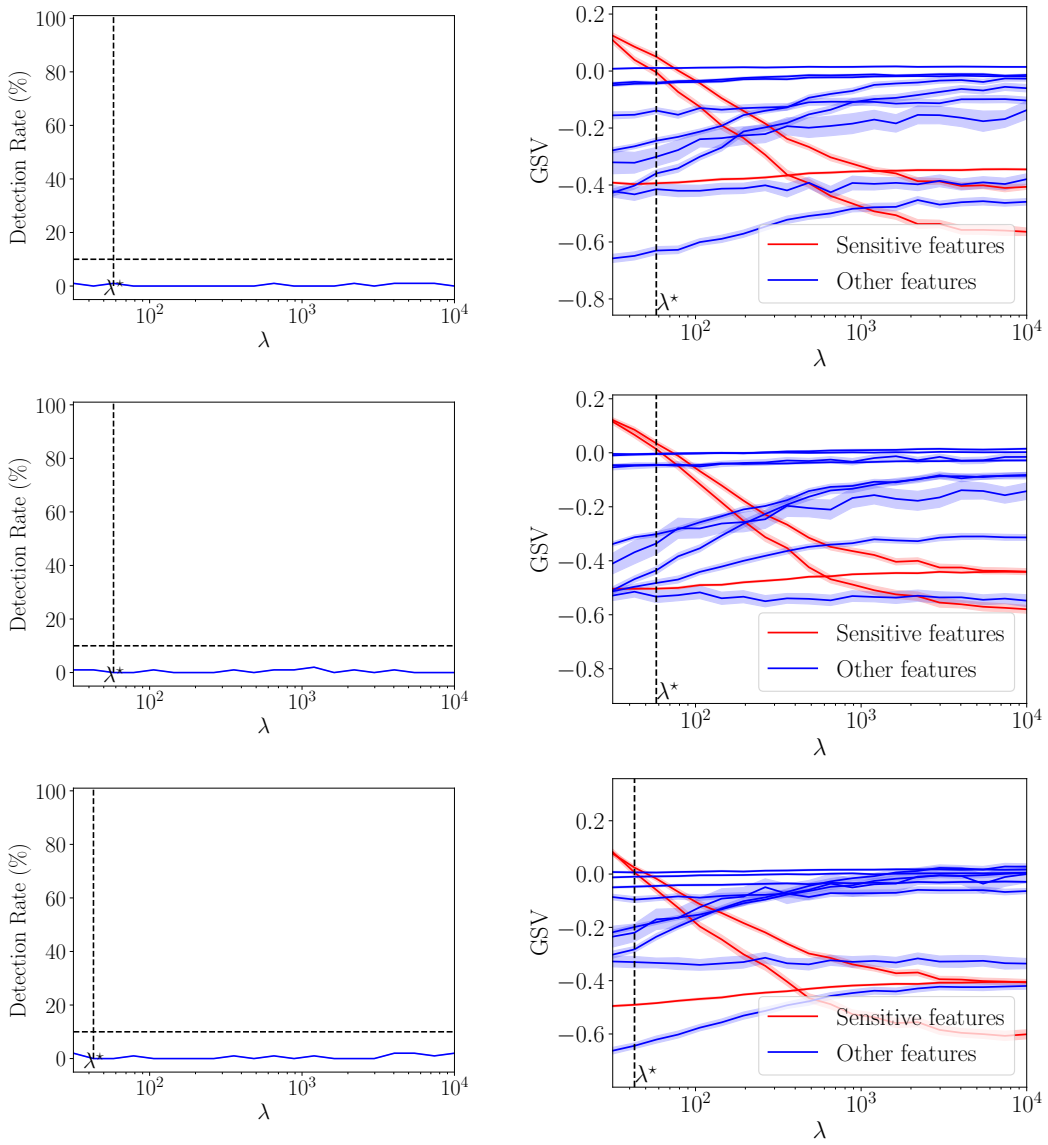

Figure 9: Example of log-space search over values of λ using XGBs classifier fitted on Adults and three sensitive attributes. Each row is a different train/test split seed. (Left) The detection rate as a function of the parameter λ of the attack. (Right) For each value of λ, the vertical slice of the 11 curves is the GSV obtained with the resulting $\mathcal{B}'_\omega$. The goal here is to reduce the amplitude all sensitive features (red curves) in order to hide their contribution to the disparity in model outcomes.

E.2 EXAMPLES OF ATTACKS

In this section, we present 8 specific examples of the attacks that were conducted on COMPAS, Adult, Marketing, and Communities.

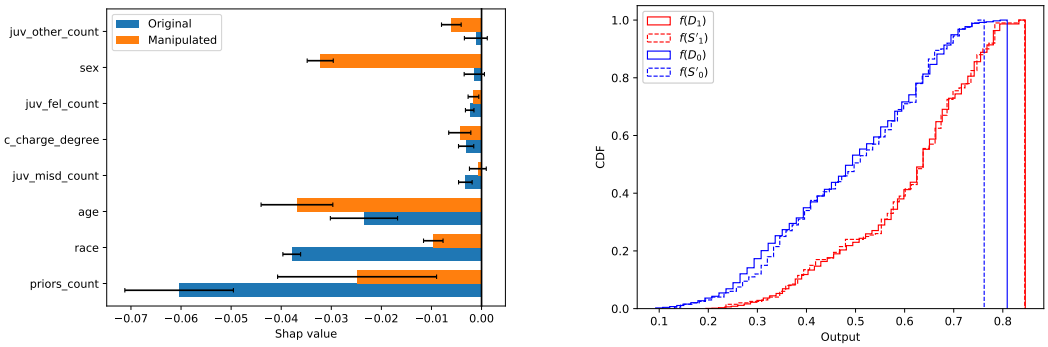

Figure 10: Attack of RF fitted on COMPAS. Left: GSV before and after the attack with $M = 200$. As a reminder, the sensitive attribute is `race`. Right: Comparison of the CDF of the misleading subsets $f(S'_0), f(S'_1)$ and the CDF over the whole data. $f(D_0), f(D_1)$.

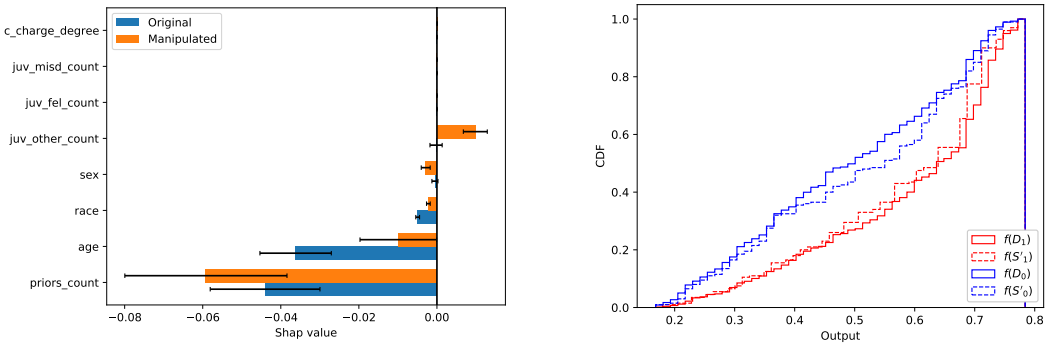

Figure 11: Attack of XGB fitted on COMPAS. Left: GSV before and after the attack with $M = 200$. As a reminder, the sensitive attribute is `race`. Right: Comparison of the CDF of the misleading subsets $f(S'_0), f(S'_1)$ and the CDF over the whole data. $f(D_0), f(D_1)$.

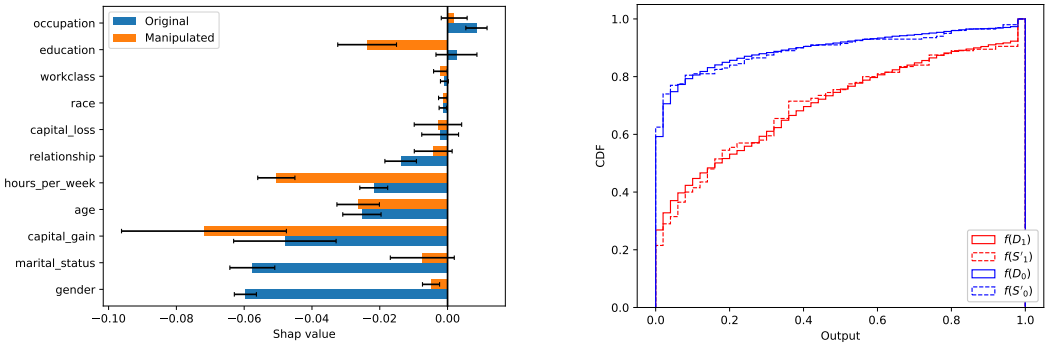

Figure 12: Attack of XGB fitted on Adults. Left: GSV before and after the attack with $M = 200$. As a reminder, the sensitive attribute is `gender`. Right: Comparison of the CDF of the misleading subsets $f(S'_0), f(S'_1)$ and the CDF over the whole data. $f(D_0), f(D_1)$.

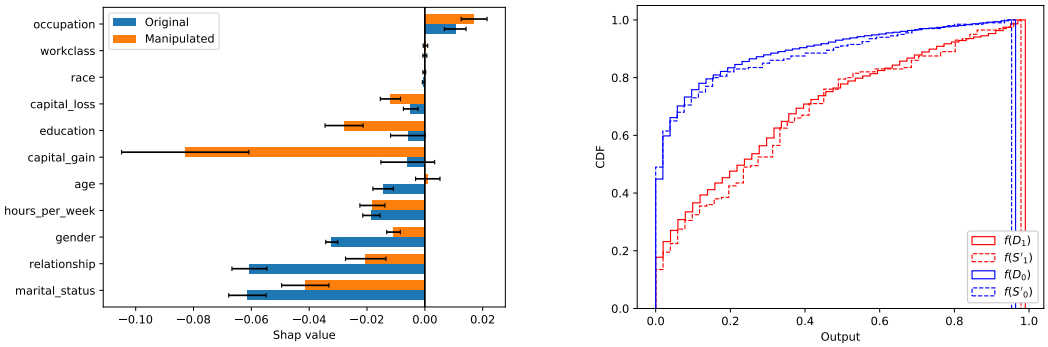

Figure 13: Attack of RF fitted on Adults. Left: GSV before and after the attack with $M = 200$. As a reminder, the sensitive attribute is `gender`. Right: Comparison of the CDF of the misleading subsets $f(S'_0), f(S'_1)$ and the CDF over the whole data. $f(D_0), f(D_1)$.

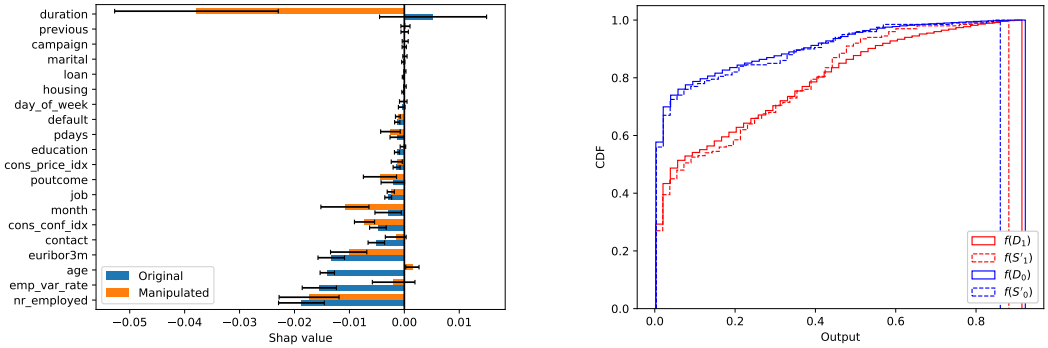

Figure 14: Attack of RF fitted on Marketing. Left: GSV before and after the attack with $M = 200$. As a reminder, the sensitive attribute is `age`. Right: Comparison of the CDF of the misleading subsets $f(S'_0), f(S'_1)$ and the CDF over the whole data. $f(D_0), f(D_1)$.

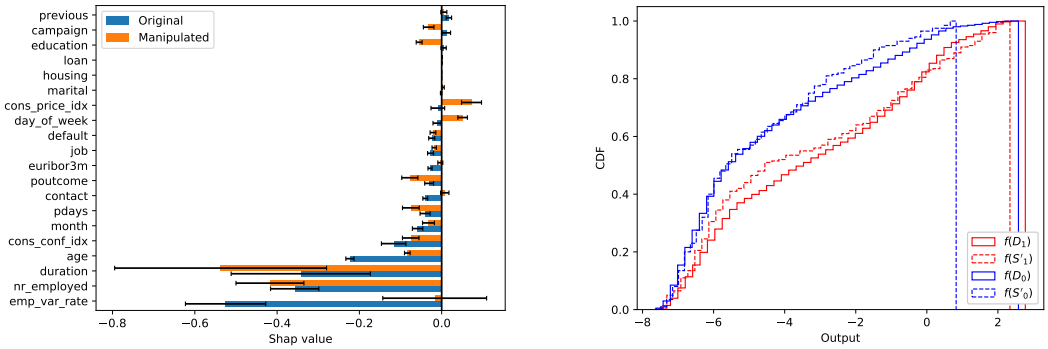

Figure 15: Attack of XGB fitted on Marketing. Left: GSV before and after the attack with $M = 200$. As a reminder, the sensitive attribute is `age`. Right: Comparison of the CDF of the misleading subsets $f(S'_0), f(S'_1)$ and the CDF over the whole data. $f(D_0), f(D_1)$. Since we used the `TreeExplainer` for this model, we had to explain its raw output which is a logit and not a probability. Hence the output is not constrained to the interval $[0, 1]$.

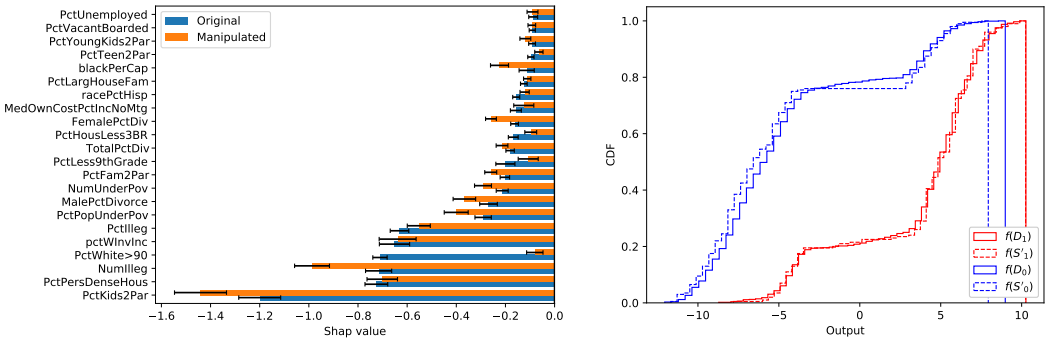

Figure 16: Attack of XGB fitted on Communities. Left: GSV before and after the attack with $M = 200$. As a reminder, the sensitive attribute is `PctWhite>90`. Right: Comparison of the CDF of the misleading subsets $f(S'_0), f(S'_1)$ and the CDF over the whole data. $f(D_0), f(D_1)$. Since we used the `TreeExplainer` for this model, we had to explain its raw output which is a logit and not a probability. Hence the output is not constrained to the interval $[0, 1]$.

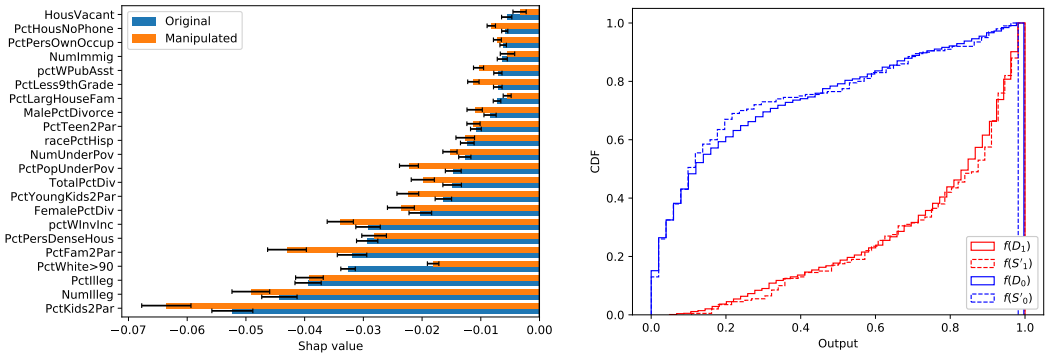

Figure 17: Attack of RF fitted on Communities. Left: GSV before and after the attack with $M = 200$. As a reminder, the sensitive attribute is `PctWhite>90`. Right: Comparison of the CDF of the misleading subsets $f(S'_0), f(S'_1)$ and the CDF over the whole data. $f(D_0), f(D_1)$.

E.3 Genetic Algorithm

This section motivates the use of stealthily biased sampling to perturb Shapley values in place of the method of Baniecki et al. (2021), which fools `SHAP` by perturbing the background dataset S'_1 via a genetic algorithm. In said genetic algorithm, a population of P fake background datasets $\{S_1'^{(k)}\}_{k=1}^{P}$ evolves iteratively following three biological mechanisms

- **Cross-Over:** Two parents produce two children by switching some of their feature values.
- **Mutation:** Some individuals are perturbed with small Uniform noise.
- **Selection:** The individuals $S_1'^{(k)}$ with the smallest amplitudes $|\Phi_s(f, S'_0, S_1'^{(k)})|$ are selected for the next generation.

Although the use of a genetic algorithm makes the method of Baniecki et al. (2021) very versatile, its main drawback is that there is no constraint on the similarity between the perturbed background and the original one. Moreover, the mutation and cross-over operations ignore the correlations between features and hence the perturbed dataset can contain unrealistic instances. Our methods solves both of these issues. Indeed, our objective is tuned to make sure that the Wasserstein distance between the original and perturbed background is kept in check. Moreover, since we do not generate new samples but rather apply non-uniform weights to pre-existing ones, we do not run into the risk of generating unrealistic data.

To illustrate these points, we have conducted an experiment on Adult-Income. For 5 different train/test splits, we have fitted a XGB model and run the genetic algorithm for 200 iterations in order to reduce the importance of the feature `gender`. At each iteration, we checked if the audit detector was able to identify the manipulation of S'_1. Results averaged over the five runs are shown in Figure 18. We see that the detector is able to systematically identify the fraud after around 50 iterations while the resulting decreases in amplitude of the sensitive feature remain small (about 30% decrease). On the other hand, results from Section 5.4 show that our attacks is undetectable and enables reductions in amplitude that range from 60% to 90% for XGB models fitted on Adults.

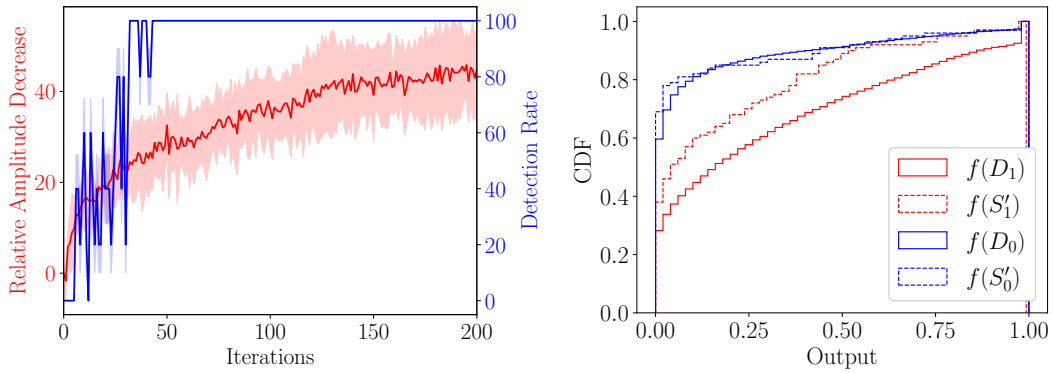

Figure 18: Genetic algorithm attacks of five XGBs fitted on Adult. Left: The relative decreases in amplitude and detection rates across five runs. Right: One example of CDF of the misleading subsets $f(S'_0), f(S'_1)$ and the CDF over the whole data. $f(D_0), f(D_1)$. Here the audit can detect the manipulation of S'_1.