A Additional Related Work

Fairness in Machine Learning. Much of the work in fairness in machine learning typically concerns the implementation of a new fairness notion in a given learning setting; either an individual fairness notion Dwork et al. (2012); Joseph et al. (2018), one based on equalizing a statistical rate across protected subgroups Hardt et al. (2016); Pleiss et al. (2017), or one based on an underlying causal model Kusner et al. (2017). With a given notion of fairness in hand, approaches to learning fair classifiers can be typically classified as "in-processing", or trying to simultaneously learn a classifier and satisfy a fairness constraint, "post-processing" which takes a learned classifier and post-processes it to satisfy a fairness definition Hardt et al. (2016), or most closely related to the motivation behind this paper, pre-processing the data to remove bias. Existing work on dataset bias serve as high level motivation for our work.

Feature Importance Notions. The local explanation methods mentioned in Section 1.2 include model-agnostic methods like LIME or SHAP Ribeiro et al. (2016); Lundberg & Lee (2017), methods like saliency maps Simonyan et al. (2013); Sundararajan et al. (2017); Baehrens et al. (2010) that require h to be differentiable in x, or model-specific methods that depend on the classifier. In addition to these explanation methods, there are also global methods that attempt to explain the entire model behavior and so can be run on the entire subgroup. Our `LIN-FID` method as described in Appendix F is a global method that relies on training an inherently interpretable model (linear regression) on the subgroup and inspecting its coefficients. Other inherently interpretable models that could be used to define a notion of subgroup importance include decision trees Quinlan (1986) and generalized additive models Liu et al. (2022).

Fairness and Interpretability. Although no existing work examines the role of feature importance notions in detecting disparities in rich subgroups, there is a small amount of existing work examining explainability in the context of fairness. The recent Grabowicz et al. (2022) formalizes induced discrimination as a function of the SHAP values assigned to sensitive features, and proposes a method to learn classifiers where the protected attributes have low influence. Begley et al. (2020) applies a similar approach, attributing a models overall unfairness to its individual features using the Shapley value, and proposing an intervention to improve fairness. Ingram et al. (2022) examines machine learning models to predict recidivism, and empirically shows tradeoffs between model accuracy, fairness, and interpretability.

Additionally, Lundberg (2020) decomposes feature attribution explanations and fairness metrics into additive components and observes the relationship between the fairness metrics and input features. Our work does not try to decompose fairness metrics into additive components and also focuses on non-additive feature explanations. Furthermore, our consideration of rich subgroups is a novel addition to the space.

B Proof of Theorem 1

We start by showing that for the unconstrained problem, computing the subgroup g_j^* that maximizes $\texttt{FID}(f_j, g, h)$ over $\mathcal{G}$ can be computed in two calls to $\text{CSC}_{\mathcal{G}}$ when F is separable.

Lemma 1. *If F is separable and $\text{CSC}_{\mathcal{G}}$ is a CSC oracle for $\mathcal{G}$, then for any feature f_j, g_j^* can be computed with two oracle calls.*

Proof. By definition $g_j^* = \text{argmax}_{g\in\mathcal{G}}\texttt{FID}(j, g) = \text{argmax}_{g\in\mathcal{G}}|F(f_j, X^n, h) - F(f_j, g, h)| = \text{argmax}_{g\in\{g^+,g^-\}}\texttt{FID}(j, g)$, where $g^+ = \text{argmax}_{g\in\mathcal{G}}F(f_j, g, h), g^- = \text{argmin}_{g\in\mathcal{G}}F(f_j, g, h)$. By the definition of separability, we can write

$$F(f_j, g(X^n), h) = \sum_{X\in g(X^n)} F'(f_j, X, h) = \sum_{i=1}^{n} g(X_i)F'(f_j, X_i, h)$$

Then letting $c_k^0 = 0$ and $c_k^1 = -F'(f_j, X_k, h)$ for $k = 1, \ldots n$, we see that $g^+ = \text{CSC}_g((c_k^0, c_k^1)), g^- = \text{CSC}_g((c_k^0, -c_k^1))$. This establishes the claim. □

Theorem 1: Let F be a separable notion, fix a classifier h, subgroup class $\mathcal{G}$, and oracle $\text{CSC}_{\mathcal{G}}$. Then fixing a feature of interest f_j, we will run Algorithm 1 twice; once with `FID` given by F, and

once with FID given by $-F$. Let $\hat{p}_{\mathcal{G}}^{T}$ be the distribution returned after $T = O(\frac{4n^2B^2}{\nu^2})$ iterations by Algorithm 1 that achieves the larger value of $\mathbb{E}[\texttt{FID}(j, g)]$. Then:

$$\begin{aligned} \texttt{FID}(j, g_j^*) - \mathbb{E}_{g\sim\hat{p}_{\mathcal{G}}^{T}}[\texttt{FID}(j, g)] \leq \nu \\ |\Phi_L(g)|, |\Phi_U(g)| \leq \frac{1+2\nu}{B} \end{aligned} \tag{3}$$

Proof. We start by transforming our constrained optimization into optimizing a $\min - \max$ objective. The $\min$ player, referred to as the *subgroup player* will be solving a CSC problem over the class $\mathcal{G}$ at each iteration, while the $\max$ player, called the *dual player*, will be adjusting the dual weights λ on the two constraints using the exponentiated gradient algorithm Kivinen & Warmuth (1997). By Lemma 2 Freund & Schapire (1996), we know that if each player implements a *no-regret* strategy, then the error of subgroup found after T rounds is sub-optimal by at most the average cumulative regret of both players. The regret bound for the exponentiated gradient descent ensures this occurs in *poly(n)* rounds.

As in Kearns et al. (2018); Agarwal et al. (2018), we first relax Equation 1 to optimize over all *distributions* over subgroups, and we enforce that our constraints hold in expectation over this distribution. Our new optimization problem becomes:

$$\begin{aligned} \min_{p_g\in\Delta(\mathcal{G})} \quad & \mathbb{E}_{g\sim p_g}[\sum_{i=1}^{n} g(x_i)F'(f_j, x_i, h)] \\ \text{s.t.} \quad & \mathbb{E}_{g\sim p_g}[\Phi_L(g)] \leq 0 \\ & \mathbb{E}_{g\sim p_g}[\Phi_U(g)] \leq 0 \end{aligned} \tag{4}$$

We note that while $|\mathcal{G}|$ may be infinite, the number of distinct labelings of X by elements of $\mathcal{G}$ is finite; we denote the number of these by $|\mathcal{G}(X)|$. Then since Equation 4 is a finite linear program in $|\mathcal{G}(X)|$ variables, it satisfies strong duality, and we can write:

$$\begin{aligned} (p_g^*, \lambda^*) &= \text{argmin}_{p_g\in\Delta(\mathcal{G})}\text{argmax}_{\lambda\in\Lambda}\mathbb{E}_{g\sim p_g}[L(g, \lambda)] = \text{argmin}_{p_g\in\Delta(\mathcal{G})}\text{argmax}_{\lambda\in\Lambda}L(p_g, \lambda) \\ \text{with} \quad L(g, \lambda) &= \sum_{x\in X} g(x)F(f_j, x, h) + \lambda_L\Phi_L + \lambda_U\Phi_U, \quad L(p_g, \lambda) = \mathbb{E}_{g\sim p_g}[L(g, \lambda)] \end{aligned}$$

As in Kearns et al. (2018) $\Lambda = \{\lambda \in \mathbb{R}^2 \mid \|\lambda\|_1 \leq B\}$ is chosen to make the domain compact, and does not change the optimal parameters as long as B is sufficiently large, i.e. $\|\lambda^*\|_1 \leq B$. In practice, this is a hyperparameter of Algorithm 1, similar to Agarwal et al. (2018); Kearns et al. (2018). Then we follow the development in Agarwal et al. (2018); Kearns et al. (2018) to show that we can compute (p_g^*, λ^*) efficiently by implementing *no-regret* strategies for the subgroup player (p_g) and the dual player (λ).

Formally, since $\mathbb{E}_{g\sim p_g}[L(g, \Lambda)]$ is bi-linear in p_g, λ, and $\Lambda, \Delta(\mathcal{G})$ are convex and compact, by Sion's minimax theorem Kindler (2005):

$$\min_{p_g\in\Delta(G)}\max_{\lambda\in\Lambda} L(p_g, \lambda) = \max_{\lambda\in\Lambda}\min_{p_g\in\Delta(G)} L(p_g, \lambda) = \text{OPT} \tag{5}$$

Then by Theorem 4.5 in Kearns et al. (2018), we know that if (p_g^*, λ^*) is a ν-approximate min-max solution to Equation 5 in the sense that

$$\begin{aligned} \text{if:} \quad & L(p_g^*, \lambda^*) \leq \min_{p\in\Delta(\mathcal{G})} L(p, \lambda^*) + \nu, L(p_g, \lambda) \geq \max_{\lambda\in\Lambda} L(p^*, \lambda), \\ \text{then:} \quad & F(f_j, p_g^*, h) \leq OPT + 2\nu, \quad |\Phi_L(g)|, |\Phi_U(g)| \leq \frac{1+2\nu}{B} \end{aligned} \tag{6}$$

So in order to compute an approximately optimal subgroup distribution p_g^*, it suffices to compute an approximate min-max solution of Equation 5. In order to do that we rely on the classic result of

Freund & Schapire (1996) that states that if the subgroup player best responds, and if the dual player achieves low regret, then as the average regret converges to zero, so does the sub-optimality of the average strategies found so far.

Lemma 2 (Freund & Schapire (1996)). *Let $p_1^\lambda, \ldots p_T^\lambda$ be a sequence of distributions over Λ, played by the dual player, and let $g^1, \ldots g^T$ be the subgroup players best responses against these distributions respectively. Let $\hat{\lambda}_T = \frac{1}{T}\sum_{t=1}^T p_t^\lambda, \hat{p}_g = \frac{1}{T}\sum_{t=1}^T g_t$. Then if*

$$\sum_{t=1}^T \mathbb{E}_{\lambda \sim p_t^\lambda}[L(g_t, \lambda)] - \min_{\lambda \in \Lambda} \sum_{t=1}^T [L(g_t, \lambda)] \leq \nu T,$$

Then $(\hat{\lambda}_T, \hat{p}_g)$ is a ν-approximate minimax equilibrium of the game.

To establish Theorem 1, we need to show (i) that we can efficiently implement the subgroup players best response using $\text{CSC}_\mathcal{G}$ and (ii) we need to translate the regret bound for the dual players best response into a statement about optimality, using Lemma 2. Establishing (i) is immediate, since at each round t, if $\lambda_{t,0} = \mathbb{E}_{p_t^\lambda}[\lambda_L], \lambda_{t,1} = \mathbb{E}_{p_t^\lambda}[\lambda_U]$, then the best response problem is:

$$\text{argmin}_{p_g \in \Delta(G)} \mathbb{E}_{g \sim p_g}[\sum_{x \in X} g(x) F(f_j, x, h) + \lambda_{t,0}\Phi_L + \lambda_{t,1}\Phi_U]$$

Which can further be simplified to:

$$\text{argmin}_{g \in G} \sum_{x \in X} g(x)(F(f_j, x, h) - \lambda_L + \lambda_U) \tag{7}$$

This can be computed with a single call of $\text{CSC}_\mathcal{G}$, as desired. To establish (ii), the no-regret algorithm for the dual player's distributions, we note that at each round the dual player is playing online linear optimization over 2 dimensions. Algorithm 1 implements the exponentiated gradient algorithm Kivinen & Warmuth (1997), which has the following guarantee proven in Theorem 1 of Agarwal et al. (2018), which follows easily from the regret bound of exponentiated gradient Kivinen & Warmuth (1997), and Lemma 2:

Lemma 3 (Agarwal et al. (2018)). *Setting $\eta = \frac{\nu}{2n^2 B}$, Algorithm 1 returns $\hat{p}_\lambda^T$ that is a ν-approximate min-max point in at most $O(\frac{4n^2 B^2}{\nu^2})$ iterations.*

Combining this result with Equation 5 completes the proof.

□

C Proof of AVG-SEPFID Primitive

In Section 3, we presented our approach that optimizing for FID constrained across a range of subgroup sizes will allow us to efficiently optimize for AVG-SEPFID. We provide a more complete proof of that claim here:

Let g^* be the subgroup that maximizes AVG-SEPFID. Without loss of generality, $g^* = \text{argmax}_{g \in \mathcal{G}} \frac{1}{n|g|} \sum g(x) F'(f_j, X, h)$ (we drop the absolute value because we can also set $F' = -F$). Then it is necessarily true, that g^* also solves the constrained optimization problem $\text{argmax}_{g \in \mathcal{G}} \frac{1}{n} \sum g(x) F'(f_j, X, h)$ such that $|g| = |g^*|$, where we have dropped the normalizing term $\frac{1}{|g|}$ in the objective function, and so we are maximizing the constrained FID.

Now consider an interval $I = [|g^*| - \alpha, |g^*| + \alpha]$, and suppose we solve $g_I^* = \text{argmax}_{g \in \mathcal{G}} \frac{1}{n} \sum g(x) F'(f_j, X, h)$ such that $g \in I$. Then since $g^* \in I$, we know that $\frac{1}{n} \sum g^* F'(f_j, X, h) \leq \frac{1}{n} \sum g_I^*(x) F'(f_j, X, h)$. This implies that:

$$\begin{aligned}\texttt{AVG-SEPFID}(g_I^*) &\geq \frac{1}{|g_I^*|}\frac{1}{n}\sum g^*(x)F'(f_j, X, h)\\ &= \texttt{AVG-SEPFID}(g^*) + (\frac{1}{|g_I^*|+\alpha} - \frac{1}{|g_I^*|})\texttt{FID}(g^*)\\ &= \texttt{AVG-SEPFID}(g^*) - \frac{\alpha}{|g^*|(|g^*|+\alpha)} \cdot \texttt{FID}(g^*)\end{aligned}$$

Given the above derivation, as $\alpha \to 0$, we have $\texttt{AVG-SEPFID}(g_I^*) \to \texttt{AVG-SEPFID}(g^*)$.

Hence we can compute a subgroup g that approximately optimizes the $\texttt{AVG-SEPFID}$ if we find an appropriately small interval I aroudn $|g^*|$. Since the discretization in Section 3 covers the unit interval, we are guaranteed for sufficiently large n to find such an interval.

D Cost Sensitive Classifier, $\text{CSC}_{\mathcal{G}}$

Definition 4. *(Cost Sensitive Classification) A Cost Sensitive Classification (CSC) problem for a hypothesis class $\mathcal{G}$ is given by a set of n tuples $\{(X_i, c_i^0, c_i^1)\}_{i=1}^n$, where c_i^0 and c_i^1 are the costs of assigning labels 0 and 1 to X_i respectively. A CSC oracle finds the classifier $\hat{g} \in \mathcal{G}$ that minimizes the total cost across all points:*

$$\hat{g} = \operatorname*{argmin}_{g \in \mathcal{G}} \sum_i \Big(g(X_i)c_i^1 + (1 - g(X_i))c_i^0 \Big) \tag{8}$$

Algorithm 2 $\text{CSC}_{\mathcal{G}}$

Input: Dataset $X \subset \mathbb{R}^{d_{sens}} \times \mathbb{R}^{d_{safe}}$, costs $(c^0, c^1) \in \mathbb{R}^n$
Let X_{sens} consist of the sensitive attributes x of each $(x, x') \in X$.
Train linear regressor $r_0 : \mathbb{R}^{d_{sens}} \to \mathbb{R}$ on dataset (X_{sens}, c^0) ▷ learn to predict the cost c^0
Train linear regressor $r_1 : \mathbb{R}^{d_{sens}} \to \mathbb{R}$ on dataset (X_{sens}, c^1) ▷ learn to predict the cost c^1
Define $g((x, x')) := \mathbf{1}\{(r_0 - r_1)(x) > 0\}$ ▷ predict 0 if the estimated $c_0 < c_1$
Return g

E NP-Completeness

We will show below that the fully general version of this problem (allowing any poly-time F) is NP complete. First, we will define a decision variant of the problem:

$$\delta_{X,F,h,A} = \max_{g \in \mathcal{G}, f_j} (|F(f_j, g, h) - F(f_j, X, h)|) \geq A$$

Note that a solution to the original problem trivially solves the decision variant. First, we will show the decision variant is in NP, then we will show it is NP hard via reduction to the max-cut problem.

Lemma 4. *The decision version of this problem is in NP.*

Proof. Our witness will be the subset g and feature f_j such that

$$(|F(f_j, g, h) - F(f_j, X, h)|) \geq A$$

Given these 2, evaluation of the absolute value is polytime given that F is polytime, so the solution can be verified in polytime. □

Lemma 5. *The decision version of this problem is NP hard.*

Proof. We will define our variables to reduce our problem to maxcut(Q, k). Given a graph defined with V, E as the vertex and edge sets of Q (with edges defined as pairs of vertices), we will define our F, X, G, A, and h as follows:

$$\begin{aligned} X &= V \\ h &= \text{constant classifier, maps every value to 1} \\ \mathcal{G} &= \mathcal{P}(V) \text{ i.e. all possible subsets of vertices} \\ F(f_j, g, h) &= |x \in E : x[0] \in g, x[1] \in g^c| \\ &\quad \text{–i.e. } F(j, g, h) \text{ returns the number of} \\ &\quad \text{edges cut by a particular subset, ignoring} \\ &\quad \text{its first and third argument.} \\ &\quad \text{(this is trivially computable in polynomial} \\ &\quad \text{time by iterating over the set of edges).} \\ A &= k \end{aligned}$$

Note that $F(f_j, X, h) = 0$ by definition, and that $F \geq 0$. Therefore, $|F(f_j, g, h) - F(f_j, X, h)| = F(f_j, g, h)$, and we see that $(|F(f_j, g, h) - F(f_j, X, h)|) \geq A$ if and only if g is a subset on Q that cuts at least $A = k$ edges. Therefore an algorithm solving the decision variant of the feature importance problem also solves maxcut. □

F Linear Feature Importance Disparity

The *non-separable* FID notion considered in this paper corresponds to training a model that is inherently interpretable on only the data in the subgroup g, and comparing the influence of feature j to the influence when trained on the dataset as a whole. Since all of the points in the subgroup can interact to produce the interpretable model, this notions typically are not separable. Below we formalize this in the case of linear regression, which is the non-separable notion we investigate in the experiments.

Definition 5. *(Linear Feature Importance Disparity). Given a subgroup g, let $\theta_g = \inf_{\theta \in \mathbb{R}^d} \mathbb{E}_{(X,y)\sim\mathcal{R}}[g(X)(\theta' X - y)^2]$, and $\theta_{\mathcal{R}} = \inf_{\theta \in \mathbb{R}^d} \mathbb{E}_{(X,y)\sim\mathcal{R}}[(\theta' X - y)^2]$. Then if e_j is the j^{th} basis vector in $\mathbb{R}^d$, we define the* linear feature importance disparity *(LIN-FID) by*

$$\texttt{LIN-FID}(j, g) = |(\theta_g - \theta_{\mathcal{R}}) \cdot e_j|$$

LIN-FID(j, g) is defined as the difference between the coefficient for feature j when training the model on the subgroup g, versus training the model on points from $\mathcal{R}$. Expanding Definition 5 using the standard weighted least squares estimator (WLS), the feature importance for a given feature f_j and subgroup $g(X)$ is:

$$F_{lin}(j, g) = \left((X g(X) X^T)^{-1} (X^T g(X) Y)\right) \cdot e_j, \tag{9}$$

Where $g(X)$ is a diagonal matrix of the output of the subgroup function. The coefficients of the linear regression model on the dataset X can be computed using the results from ordinary least squares (OLS): $(XX^T)^{-1}(X^T Y) \cdot e_j$.

We compute argmax$_{g\in G}$LIN-FID $=$ argmax$_{g\in G}|F_{lin}(j, X^n) - F_{lin}(j, g)|$ by finding the minimum and maximum values of $F_{lin}(j, g)$ and choosing the one with the larger difference. For the experiments in Section 4, we use logistic regression as the hypothesis class for g because it is non-linear enough to capture complex relationships in the data, but maintains interpretability in the form of its coefficients, and importantly because Equation 9 is then differentiable in the parameters θ of $g(X) = \sigma(X \cdot \theta), \sigma(x) = \frac{1}{1+e^{-x}}$. Since Equation 9 is differentiable in θ, we can use non-convex optimizers like SGD or ADAM to maximize Equation 9 over θ.

While this is an appealing notion due to its simplicity, it is not relevant unless the matrix $X g(X) X^T$ is of full rank. We ensure this first by lower bounding the size of g via a size penalty term $P_{size} = \max(\alpha_L - |g(X_{train})|, 0) + \max(|g(X_{train})| - \alpha_U, 0)$, which allows us to provide α constraints

in the same manner as in the separable approach. We also add a small l_2 regularization term ϵI to $X^T g(X) X$. This forces the matrix to be invertible, avoiding issues with extremely small subgroups. Incorporating these regularization terms, Equation 9 becomes:

$$F_{lin}(j, g) = \lambda_s \cdot \left((X\sigma(X \cdot \theta_L^T)X^T + \epsilon I)^{-1}(X^T\sigma(X \cdot \theta_L^T)Y) \cdot e_j\right) + \lambda_c \cdot P_{size} \tag{10}$$

We note that LIN-FID is a similar notion to that of LIME Ribeiro et al. (2016), but LIME estimates a local effect around each point which is then summed to get the effect in the subgroup, and so it is *separable*. It is also the case that F_{lin} is non-convex as shown below:

Lemma 6. *F_{lin} as defined in Equation 9 is non-convex.*

Proof. We will prove this by contradiction. Assume F_{lin} is convex, which means the Hessian is positive semi-definite everywhere. First we will fix $(Xg(X)X^T))^{-1}$ to be the identity matrix, which we can do without loss of generality by scaling g by a constant. This scaling will not affect the convexity of F_{lin}.

Now, we have the simpler form of $F_{lin} = (X^T g(X)Y) \cdot e_j$. We then can compute the values of the Hessian:

$$\frac{\partial F^2}{\partial^2 g} \quad = \quad (X^T g''(X)Y) \cdot e_j$$

Consider the case where X^T is a 2×2 matrix with rows $1, 0$ and $0, -1$ and Y is a vector of ones. If g weights the second column (i.e. feature) greater than the first, then the output Hessian will be positive semi-definite. But if g weights the first column greater than the first, then it will be negative semi-definite. Since the Hessian is not positive semi-definite everywhere, F_{lin} must be non-convex over the space of g. □

This means the stationary point we converge to via gradient descent may only be locally optimal. In Section 4, we optimize Equation 10 using the ADAM optimizer Kingma & Ba (2015). Additional details about implementation and parameter selection are in Appendix G. Despite only locally optimal guarantees, we were still able to find (feature, subgroup) pairs with high LIN-FID for all datasets.

G Experimental Details

G.1 Algorithmic Details

Separable Case. In order to implement Algorithm 1 over a range of $[\alpha_L, \alpha_U]$ values, we need to specify our dual norm B, learning rate η, number of iterations used T, rich subgroup class $\mathcal{G}$, and the associated oracle CSC$_\mathcal{G}$. We note that for each feature f_j, Algorithm 1 is run twice; one corresponding to maximizing $\texttt{FID}(f_j, g, h)$ and the other minimizing it. Note that in both cases our problem is a minimization, but when maximizing we simply negate all of the point wise feature importance values $F(f_j, x_i, h) \to -F(f_j, x_i, h)$. In all experiments our subgroup class $\mathcal{G}$ consists of linear threshold functions over the sensitive features: $\mathcal{G} = \{\theta \in \mathbb{R}^{d_{sens}} : \theta((x, x')) = \mathbf{1}\{\theta' x > 0\}$. We implement CSC$_\mathcal{G}$ as in Agarwal et al. (2018); Kearns et al. (2018) via linear regression, see Algorithm 2 in Appendix D. To ensure the dual player's response is strong enough to enforce desired size constraints, we empirically found that setting the hyperparameter $B = 10^4 \cdot \mu(f_j)$ worked well on all datasets, where $\mu(f_j)$ is the average absolute importance value for feature j over X. We set the learning rate for exponentiated gradient descent to $\eta = 10^{-5}$. Empirical testing showed that $\eta \cdot B$ should be on the order of $\mu(f_j)$ or smaller to ensure proper convergence. We found that setting the error tolerance hyperparameter $\nu = .05 \cdot \mu(f_j) \cdot n \cdot \alpha_L$ worked well in ensuring good results with decent convergence time across all datasets and values of α. For all datasets and methods we ran for at most $T = 5000$ iterations, which we observe empirically was large enough for FID values to stabilize and for $\frac{1}{T}\sum_{t=1}^{T} |g_t| \in [\alpha_L, \alpha_U]$, with the method typically converging in $T = 3000$ iterations or less. See Appendix M for a sample of convergence plots.

Non-Separable Case. For the non-separable approach, datasets were once again split into train and test sets. For Student, it was split 50-50, while COMPAS, Bank, and Folktables were split 80-20

train/test. The 50-50 split for Student was chosen so that a linear regression model would be properly fit on a small $g(X_{test})$. The parameter vector θ for a logistic regression classifier was randomly initialized with a PyTorch random seed of 0 for reproducability. We used an ADAM Kingma & Ba (2015) optimizer with a learning rate of .05 as our heuristic solver for the loss function.

To enforce subgroup size constraints, $\lambda_s P_{size}$ must be on a significantly larger order than $\lambda_c F_{lin}(j, g)$. Empirical testing found that values of $\lambda_s = 10^5$ and $\lambda_c = 10^{-1}$ returned appropriate subgroup sizes and also ensured smooth convergence. The optimizer ran until it converged upon a minimized linear regression coefficient, subject to the size constraints. Experimentally, this took at most 1000 iterations, see Appendix N for a sample of convergence plots. After solving twice for the minimum and maximum $F_{lin}(j, g)$ values and our subgroup function g is chosen, we fit the linear regression on both X_{test} and $g(X_{test})$ to get the final `FID`.

G.2 FID NOTIONS

LIME: A random forest model h was trained on dataset X^n. Then each data point along with the corresponding probability outputs from the classifier were input into the LIME Tabular Explainer Python module. This returned the corresponding LIME explanation values.

SHAP: This was done with the same method as LIME, except using the SHAP Explainer Python module.

Vanilla Gradient: Labeled as *GRAD* in charts, the vanilla gradient importance notion was computed using the Gradient method from the OpenXAI library Agarwal et al. (2022b). This notion only works on differentiable classifiers so in this case, h is a logistic regression classifier. We found there was no substantial difference between the choice of random forest or logistic regression for h when tested on other importance notions (see Section J). Due to constraints on computation time, this method was only tested on the COMPAS dataset (using `Two Year Recidivism` as the target variable).

Linear Regression: For the linear regression notion, the subgroup g was chosen to be in the logistic regression hypothesis class. For a given subgroup $g(X)$, the weighted least squares (WLS) solution is found whose linear coefficients θ_g then define the feature importance value $e_j \cdot \theta_g$.

For details on the consistency of these importance notions, see Appendix O.

G.3 DATASETS

These four datasets were selected on the basis of three criterion: (i) they all use features which could be considered *sensitive* to make predictions about individuals in a context where bias in a significant concern (ii) they are heavily used datasets in research on interpretability and fairness, and as such issues of bias in the datasets should be of importance to the community, and (iii) they trace out a range of number of datapoints and number of features and sensitive features, which we summarise in Table 4. For each dataset, we specified features that were "sensitive." That is, when searching for subgroups with high `FID`, we only considered rich subgroups defined by features generally covered by equal protection or privacy laws (e.g. race, gender, age, health data).

Student: This dataset aims to predict student performance in a Portugese grade school using demographic and familial data. For the purposes of this experiment, the target variable was math grades at the end of the academic year. Student was by far the smallest of the four datasets with 395 data points. The sensitive features in Student are `gender`, `parental status`, `address` (urban or rural), `daily alcohol consumption`, `weekly alcohol consumption`, and `health`. `Age` typically would be considered sensitive but since in the context of school, age is primarily an indicator of class year, this was not included as a sensitive feature. The categorical features `address`, `Mother's Job`, `Father's Job`, and `Legal Guardian` were one hot encoded.

COMPAS: This dataset uses a pre-trial defendant's personal background and past criminal record to predict risk of committing new crimes. To improve generalizability, we removed any criminal charge features that appeared fewer than 10 times. Binary counting features (e.g. `25-45 yrs old` or `5+ misdemeanors`) were dropped in favor of using the continuous feature equivalents. Additionally, the categorical variable `Race` was one-hot encoded. This brought the total number of features to 95. The sensitive features in COMPAS are `age`, `gender`, and `race` (Caucasian, African-American,

Asian, Hispanic, Native American, and Other). For COMPAS, we ran all methodologies twice, once using the binary variable, `Two Year Recidivism`, as the target variable and once using the continuous variable `Decile Score`. `Two Year Recidivism` is what the model is intended to predict and is labeled as *COMPAS R* in the results. Meanwhile, `Decile Score` is what the COMPAS system uses in practice to make recommendations to judges and is labeled as *COMPAS D* in the results.

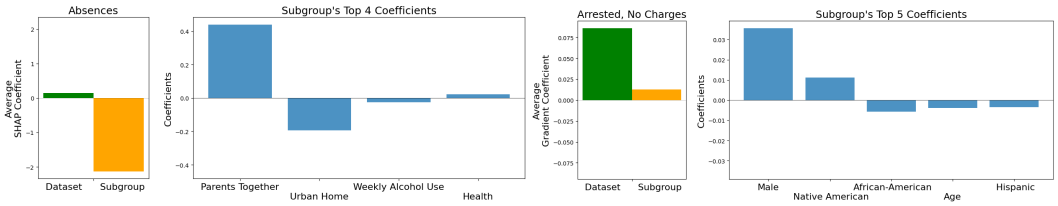
Bank: This dataset looks at whether a potential client signed up for a bank account after being contacted by marketing personnel. The sensitive features in Bank are `age` and `marital status` (married, single, or divorced). The `age` feature in Bank is a binary variable representing whether the individual is above the age of 25.

Folktables: This dataset is derived from US Census Data. Folktables covers a variety of tasks, but we used the ACSIncome task, which predicts whether an individual makes more than $50k per year. The ACSIncome task is meant to mirror the popular Adult dataset, but with modifications to address sampling issues. For this paper, we used data from the state of Michigan in 2018. To reduce sparseness of the dataset, the `Place of Birth` feature was dropped and the `Occupation` features were consolidated into categories of work as specified in the official Census dictionary Bureau (2020), (e.g. people who work for the US Army, Air Force, Navy, etc. were all consolidated into `Occupation=Military`). The sensitive features in Folktables are `age`, `sex`, `marital status` (married, widowed, divorced, separated, never married/under 15 yrs old), and `race` (Caucasian, African-American, Asian, Native Hawaiian, Native American singular tribe, Native American general, Other, and 2+ races).

Table 4: Summary of Datasets

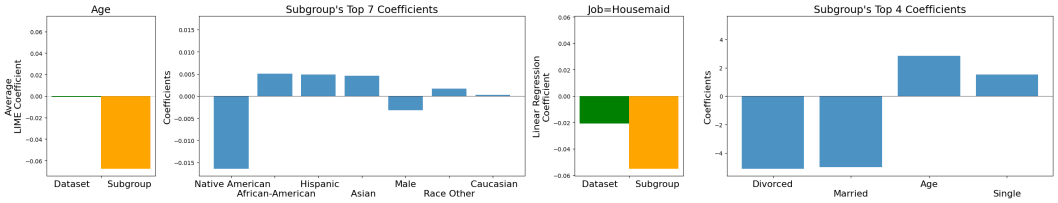
Dataset	Data Points	# of Features	# of Sensitive Features
Student	395	32	6
COMPAS	6172	95	8
Bank	30488	57	4
Folktables Income	50008	52	16

H More Discussion of High FID Subgroups



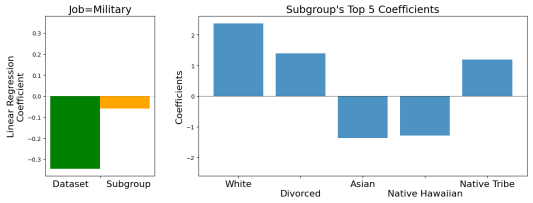
(a) Student: Predicting grade outcomes

(b) COMPAS: Predicting 2-year recidivism



(c) COMPAS: Predicting decile risk score

(d) Bank: Predicting whether bank deposit is made



(e) Folktables: Predicting income >$50k

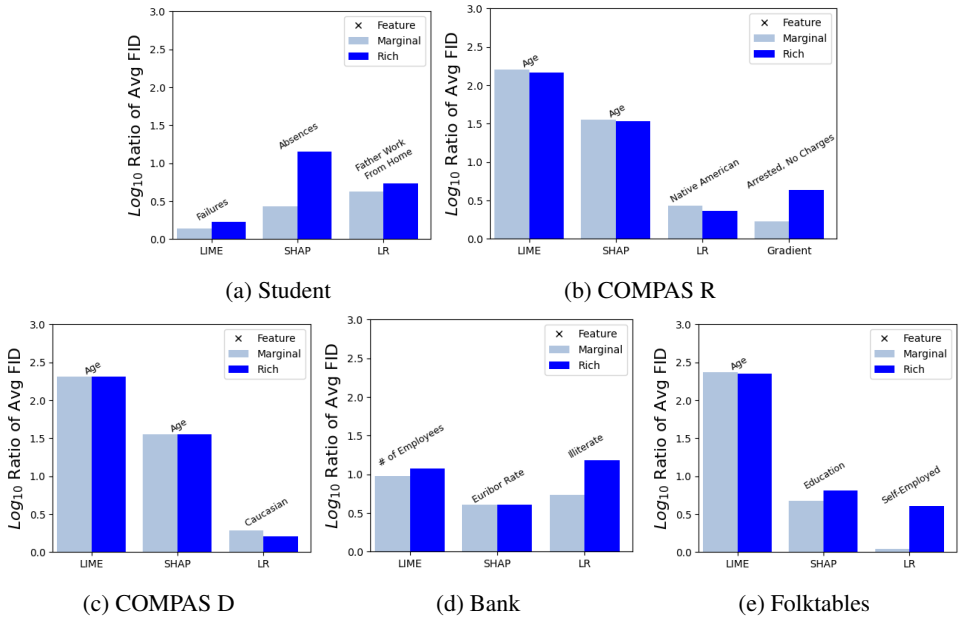
Figure 5: Exploration of key subgroup/feature pairs found for each dataset. The first graph shows the change in feature importance from whole dataset to subgroup. The second graph shows the main coefficients that define the subgroup.

In Figure 5, we highlight selections of an interesting (feature, subgroup, method) pair for each dataset. Figure 5a shows that on the Student dataset the feature `absences` which is of near zero importance on the dataset as a whole, is very negatively correlated with student performance on a subgroup whose top 2 features indicate whether a student's parents are together, and if they live in an rural neighborhood. Figure 5b shows that on the COMPAS dataset with method GRAD, the feature `arrested-but-with-no-charges` is typically highly important when predicting `two-year-recidivism`. However, it carries significantly less importance on a subgroup that is largely defined as Native American males. When predicting the decile risk score on COMPAS, LIME indicates that age is not important on the dataset as a whole; however, for non-Native American, female minorities, older age can be used to explain a lower `Decile Score`. On the Bank dataset using `LIN-FID`, we see that a linear regression trained on points from a subgroup defined by older, single individuals, puts more importance on `job=housemaid` when predicting likelihood in signing up for an account. Finally on Folktables, we see that `LIN-FID` assigns much lower weight to the `job=military` feature among a subgroup that is mainly white and divorced people than in the overall dataset when predicting income. These interesting examples, in conjunction with the results reported in Table 1, highlight the usefulness of our method in finding subgroups where a concerned analyst/domain expert could dig deeper to determine how biases might be manifesting themselves in the data and if/how to correct for them.

I Comparison of FID Values on Rich vs. Marginal Subgroups

To better justify the use of rich subgroups, we performed the same analysis but only searching over the marginal subgroup space. For each dataset and importance notion pair, we established the finite list of marginal subgroups defined by a single sensitive characteristic and computed the feature importance values on each of these subgroups. In Figure 6, we compare the maximal `AVG-SEPFID`

rich subgroups shown in Figure 2 to the maximal AVG-SEPFID marginal subgroup for the same feature. In about half of the cases, the AVG-SEPFID of the marginal subgroup was similar to the rich subgroup. In the other cases, expanding our subgroup classes to include rich subgroups defined by linear functions of the sensitive attributes enabled us to find a subgroup that had a higher AVG-SEPFID. For example, in Figure 6b, we can see that on the COMPAS R dataset using GRAD as the importance notion, `Arrested, No Charges` had a rich subgroup with AVG-SEPFID that was 4 times less than on the full dataset. However, we were unable to find any subgroup in the marginal space where the importance of the feature was nearly as different. In some cases in Figure 6, the marginal subgroup performs slightly better than the rich subgroup. This happens when using rich subgroups does not offer any substantial advantage over marginal subgroups, and the empirical error tolerance in Algorithm 1 stopped the convergence early.



(a) Student (b) COMPAS R

(c) COMPAS D (d) Bank (e) Folktables

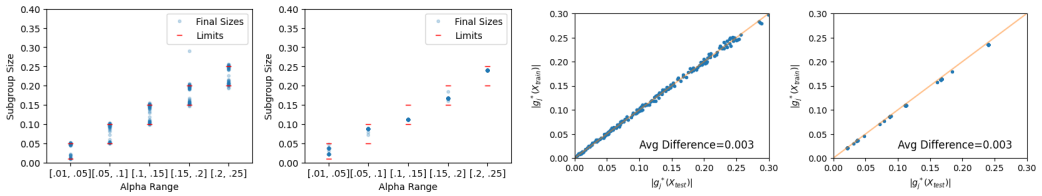
Figure 6: Comparison of the maximal FID rich subgroups from Figure 2 to the maximal FID marginal subgroup on the same feature. This is displayed as $|log_{10}(R)|$ where R is the ratio of average importance per data point for separable notions and the ratio of coefficients for the linear coefficient notion. The feature associated with the subgroups is written above each bar.

J Statistical Validity of Results: Generalization of FID and $|g|$

When confirming the validity of our findings, there are two potential concerns: (1) Are the subgroup sizes found in-sample approximately the same on the test set and (2) do the FID's found on the training set generalize out of sample? Taken together, (1) and (2) are sufficient to guarantee our maximal AVG-SEPFID values generalize out of sample.

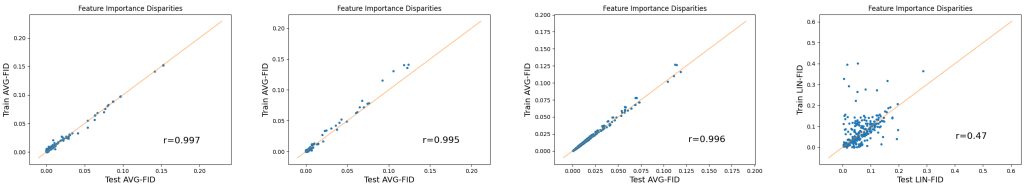
In Figure 7, we can see that when we take the maximal subgroup found for each feature f_j, g_j^*, and compute it's size $|g_j^*|$ on the test set, for both the separable and non-separable methods it almost always fell within the specified $[\alpha_L, \alpha_U]$ range; the average difference in $|g_j^*(X_{train})|$ and $|g_j^*(X_{test})|$ was less than .005 on all notions of feature importance and all datasets except for Student, which was closer to .025 due to its smaller size. A few rare subgroups were significantly outside the desired α range, which was typically due to the degenerate case of the feature importance values all being 0 for the feature in question. Additional plots for all (dataset, notion) pairs are in Appendix L.

In Figure 8, we compare AVG-SEPFID(f_j, g_j^*, X_{train}) to AVG-SEPFID(f_j, g_j^*, X_{test}), or LIN-FID in the case of the linear regression notion, to see how FID generalizes. The separable notions all generalized very well, producing very similar AVG-SEPFID values for in and out of sample tests. The non-separable method still generalized, although not nearly as robustly, with



(a) (b) (c) (d)

Figure 7: Generalizability of $|g|$ on the Folktables dataset. (a) Size outputs from Algorithm 1 for all features and separable notions and (b) from optimizing Equation 10 for LIN−FID show that our size constraints hold in-sample. (c) Plots the corresponding values of $|g_j^*(X_{train})|$ vs $|g_j^*(X_{test})|$ for separable notions and (d) for LIN−FID, showing that the subgroup size generalizes out of sample.



(a) LIME (b) SHAP (c) GRAD (d) Linear Regression

Figure 8: Out of sample generalization of the methods. Each dot represents a feature, plotting FID on X_{test} vs on X_{train}. All are computed on the Folktables dataset except (c) is computed on COMPAS R. The diagonal line represents perfect generalization and the Pearson correlation coefficient is displayed in figure. The non-separable approach suffers from the instability of the WLS method.

outlier values occurring. This was due to ill-conditioned design matrices for small subgroups leading to instability in fitting the least squares estimator. In Appendix O, we investigate the robustness of the feature importance notions, evaluated on the entire dataset. We find that the coefficients of linear regression are not as stable, indicating the lack of generalization in Figure 8 could be due to the feature importance notion itself lacking robustness, rather than an over-fit selection of g_j^*.

K Choice of Hypothesis Class

One ablation study we explored was the choice of classification model h. While the main experiments used a random forest model, we also explored using a logistic regression model. The logistic regression model was implemented with the default sklearn hyperparameters. We found that the results are roughly consistent with each other no matter the choice of h. In Table 5 and Table 6, we see that the features with the highest AVG−SEPFID, their subgroup sizes, and the AVG−SEPFID values are consistent between the choice of hypothesis class. We then looked further into the features that were used to define these subgroups. In Figure 9, we see that the subgroups with high AVG−SEPFID for the feature Age were both defined by young, non-Asians.

Similarly consistent results were found across all feature importance notions and datasets. As a result, all of the results presented in the main section of the paper used random forest as the hypothesis class.

L Subgroup Sizes Outcomes

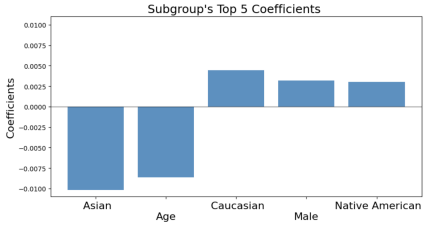
In Figure 10 we chart the subgroup sizes, $|g(X_{test})|$, outputted by the algorithms across all dataset and importance notion combinations. As a whole, the final subgroup sizes were generally within the specified α range. Occasionally, there were subgroups which were significantly outside the expected range. Usually this was due to most of the importance values, $F(f_j, X, h)$, being zero for a given feature.

h = Random Forest			h = Logistic Regression		
Feature	Size	AVG-SEPFID	Feature	Size	AVG-SEPFID
Age	$.05 - .1$	.144	Age	$.05 - .1$	.21
Priors Count	$.01 - .05$	.089	Priors count	$.01 - .05$	.092
Juv Other Count	$.01 - .05$	.055	Juv Other Count	$.01 - .05$	.055
Other Features	-	$< .025$	Other Features	-	$< .025$

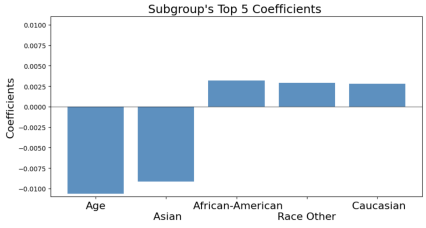
Table 5: Comparing results between using random forest and logistic regression as the hypothesis class for classifier h using LIME as the importance notion on the COMPAS R dataset. Here we display the features with the highest AVG-SEPFID, the subgroup size $|g|$, and the AVG-SEPFID. We can see that the choice of hypothesis class h does not substantially affect the output. We used random forest for all of our main experiments.

h = Random Forest			h = Logistic Regression		
Feature	Size	AVG-SEPFID	Feature	Size	AVG-SEPFID
Age	$.01 - .05$	.4	Age	$.01 - .05$	.21
Priors Count	$.01 - .05$	.11	Priors count	$.01 - .05$	.14
Other Features	-	$< .05$	Other Features	-	$< .05$

Table 6: Same as Table 5 except using SHAP as the importance notion. With SHAP, there were fewer features with significant AVG-SEPFID before dropping off but in both cases, the choice of h did not significantly affect the outcome.



(a) h = Random Forest, f_{j^*} = Age



(b) h = Logistic Regression, f_{j^*} = Age

Figure 9: Comparing the choice of hypothesis class of h. Here we show the defining coefficients for the highest AVG-SEPFID subgroup found on the COMPAS R dataset using LIME as the feature importance notion. For the feature Age, we find that young and non-Asian were the two most defining coefficients for g^* no matter which choice of h.

In Figure 11 we compare $|g_j^*(X_{train})|$ and $|g_j^*(X_{test})|$, outputted by the algorithms across all dataset and importance notion combinations. As we can see, the subgroup sizes were very consistent between the train and test set meaning $|g|$ generalized very well. The average difference was only somewhat large on the Student dataset, due to the fact that it is a smaller dataset.

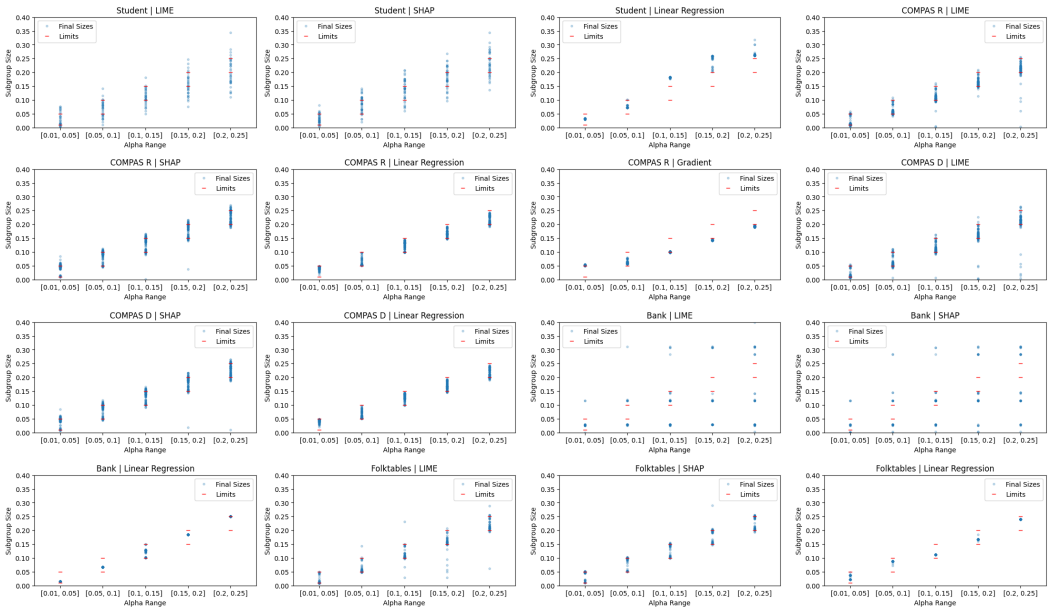
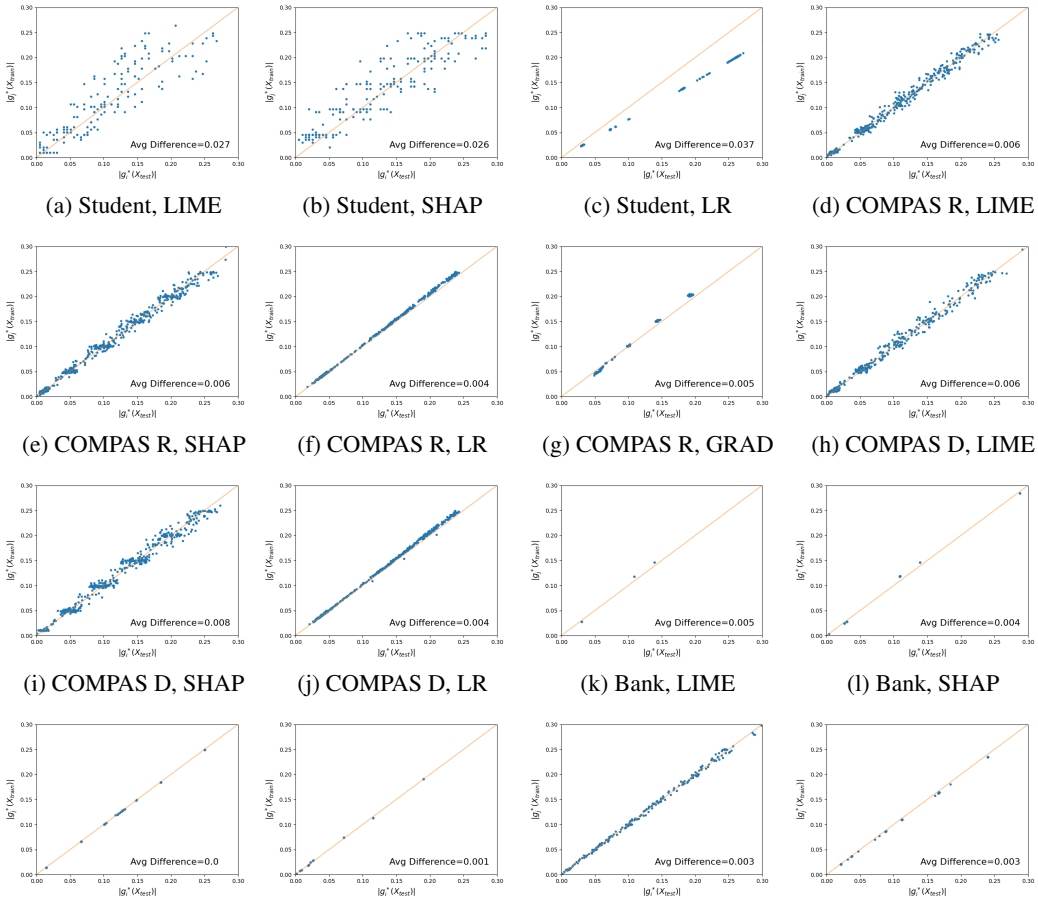


Figure 10: Final subgroup sizes of $g(X_{test})$ compared with α range. These almost always fall within the correct size range. Student has the largest errors, mostly due to the fact that the dataset itself is small.



(a) Student, LIME (b) Student, SHAP (c) Student, LR (d) COMPAS R, LIME

(e) COMPAS R, SHAP (f) COMPAS R, LR (g) COMPAS R, GRAD (h) COMPAS D, LIME

(i) COMPAS D, SHAP (j) COMPAS D, LR (k) Bank, LIME (l) Bank, SHAP

(m) Bank, LR (n) Folk, LIME (o) Folk, SHAP (p) Folk, LR

Figure 11: Comparing $|g_j^*(X_{train})|$ and $|g_j^*(X_{test})|$. We can see that the size of the subgroup was consistent between the train and test set.

M ALGORITHM 1 OPTIMIZATION CONVERGENCE

Here are additional graphs showing examples of the convergence of Algorithm 1. Data was tracked every 10 iterations, recording the Lagrangian values (to compute the error $v_t = max(|L(\hat{p}^t_{\mathcal{G}}, \hat{p}^t_{\lambda}) - \underline{L}|, |\overline{L} - L(\hat{p}^t_{\mathcal{G}}, \hat{p}^t_{\lambda})|)$), the subgroup size, and `AVG-SEPFID` value, graphed respectively in Figure 12. We can see `AVG-SEPFID` value moving upward, except when the subgroup size is outside the α range, and the Lagrangian error converging upon the set error bound v before terminating.

While Theorem 1 states that convergence time may grow quadratically, in practice we found that computation time was not a significant concern. The time for convergence varied slightly based on dataset but for the most part, convergence for a given feature was achieved in a handful of iterations that took a few seconds to compute. Features which took several thousand iterations could take around 30 minutes to compute on larger datasets.

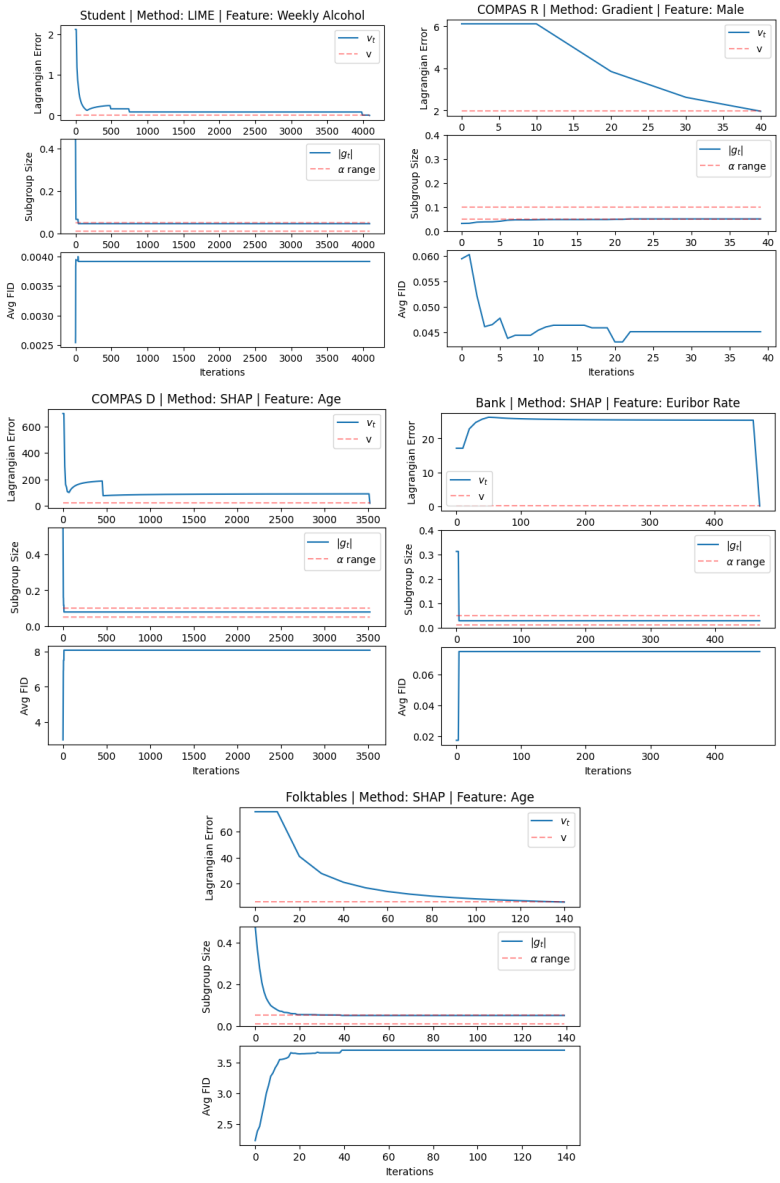


Figure 12: Plots detailing the convergence of Algorithm 1. The top plot shows the error convergence, i.e. the max difference in Lagrangian values between our solution and the min/max-players' solution. The other two plots display the subgroup size and AVG−SEPFID of the solution. Convergence almost always happened in fewer than 5000 iterations, allaying concerns about theoretical run time.

N NON-SEPARABLE OPTIMIZATION CONVERGENCE

Here are additional graphs showing the convergence in the non-separable approach. Using the loss function that rewards minimizing the linear regression coefficient (or maximizing it) and having a size within the alpha constraints, we typically reach convergence after a few hundred iterations. In Figure 13, we can see in the respective upper graphs that the subgroup size converges to the specified α range and stays there. Meanwhile, in the lower graph, we see the LIN-FID attempt to maximize but oscillates as the appropriate size is found.

Convergence using this method was almost always achieved in under 1000 iterations. Running this for all features took around 2 hours to compute on the largest datasets. The optimization was run using GPU computing on NVIDIA Tesla V100s.

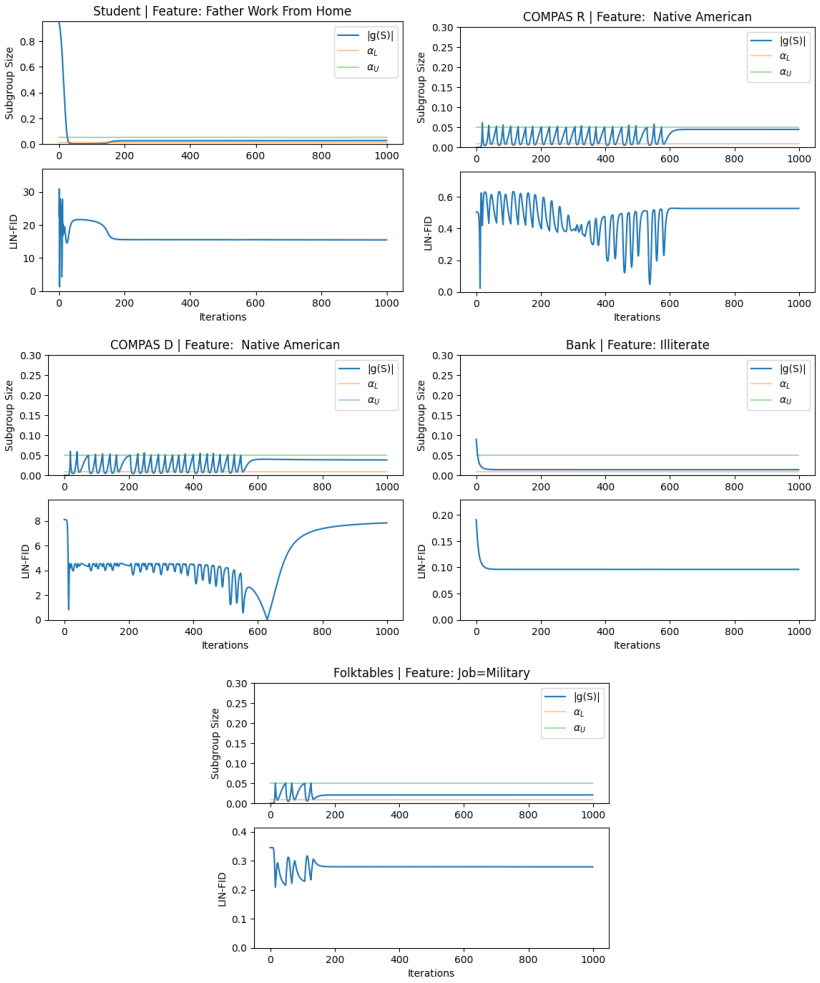


Figure 13: Plots of subgroup size and linear regression coefficient of g over the training iterations of the Adam optimizer. For each dataset, the feature with the highest LIN-FID was displayed.

O IMPORTANCE NOTION CONSISTENCY

To see how consistent importance notion methods were, we plotted the values of $F(f_j, X_{test}, h)$ against $F(f_j, X_{train}, h)$ with each point representing a feature f_j of the COMPAS dataset. The closer these points track the diagonal line, the more consistent a method is in providing the importance values. As we can see in Figure 14, LIME and GRAD are extremely consistent. Linear regression is less consistent, due to instability in fitting the least squares estimator on ill-conditioned design matrices. SHAP is also inconsistent in its feature importance attribution, however the `AVG-SEPFID` still generalized well as seen in Figure 8. This could mean that while SHAP is inconsistent from dataset to dataset, it is consistent relative to itself. i.e. if $F(j, X_{train}) > F(j, X_{test})$ then $F(j, g(X_{train})) > F(f, X_{test})$ meaning the `AVG-SEPFID` value would remain the same.

These inconsistencies seem to be inherent in some of these explainability methods as noted in other research Krishna et al. (2022); Dai et al. (2022); Agarwal et al. (2022a); Alvarez-Melis & Jaakkola (2018); Bansal et al. (2020). Exploring these generalization properties would be an exciting future direction for this work.

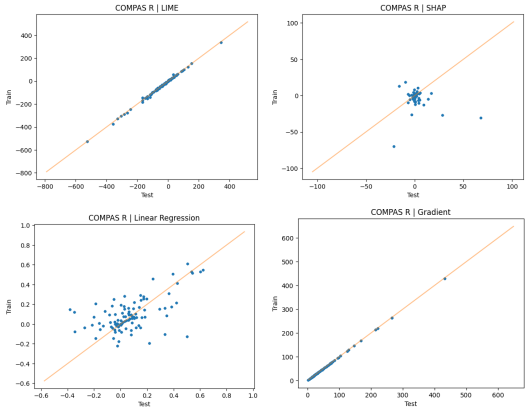


Figure 14: Consistencies of importance notions. Each point represents a feature, the x-value is $F(j, X_{test})$, and y-value is $F(j, X_{train})$. The closer the points are to the diagonal, the more consistent the notion is.