D-BIAS: A Causality-Based Human-in-the-Loop System for Tackling Algorithmic Bias

Bhavya Ghai and Klaus Mueller, *Senior Member, IEEE*

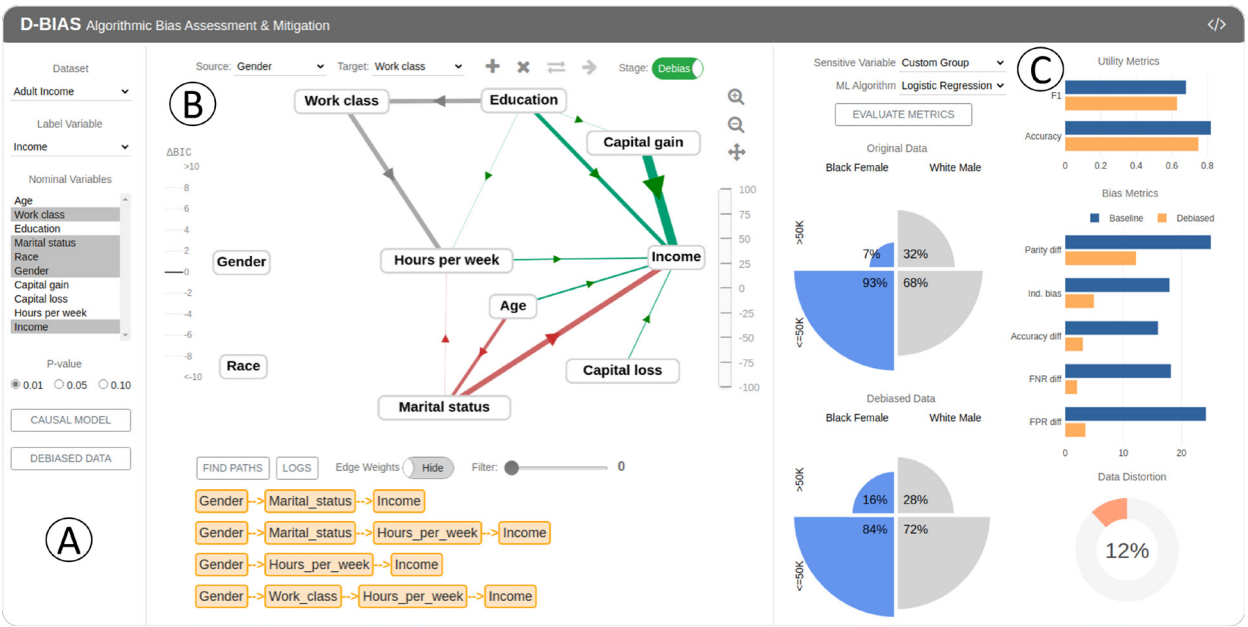

Fig. 1. The visual interface of our D-BIAS tool. (A) The Generator panel: used to create the causal network and download the debiased dataset (B) The Causal Network view: shows the causal relations between the attributes of the data, allows user to inject their prior in the system (C) The Evaluation panel: used to choose the sensitive variable, the ML model and displays different evaluation metrics.

Abstract— With the rise of AI, algorithms have become better at learning underlying patterns from the training data including ingrained social biases based on gender, race, etc. Deployment of such algorithms to domains such as hiring, healthcare, law enforcement, etc. has raised serious concerns about fairness, accountability, trust and interpretability in machine learning algorithms. To alleviate this problem, we propose D-BIAS, a visual interactive tool that embodies human-in-the-loop AI approach for auditing and mitigating social biases from tabular datasets. It uses a graphical causal model to represent causal relationships among different features in the dataset and as a medium to inject domain knowledge. A user can detect the presence of bias against a group, say females, or a subgroup, say black females, by identifying unfair causal relationships in the causal network and using an array of fairness metrics. Thereafter, the user can mitigate bias by refining the causal model and acting on the unfair causal edges. For each interaction, say weakening/deleting a biased causal edge, the system uses a novel method to simulate a new (debiased) dataset based on the current causal model while ensuring a minimal change from the original dataset. Users can visually assess the impact of their interactions on different fairness metrics, utility metrics, data distortion, and the underlying data distribution. Once satisfied, they can download the debiased dataset and use it for any downstream application for fairer predictions. We evaluate D-BIAS by conducting experiments on 3 datasets and also a formal user study. We found that D-BIAS helps reduce bias significantly compared to the baseline debiasing approach across different fairness metrics while incurring little data distortion and a small loss in utility. Moreover, our human-in-the-loop based approach significantly outperforms an automated approach on trust, interpretability and accountability.

Index Terms—Algorithmic Fairness, Causality, Debiasing, Human-in-the-loop, Visual Analytics

1 Introduction

When computer systems discriminate based on an individual's inherent characteristic such as gender, or acquired traits such as nationality, which are protected classes under law and are irrelevant to the decision making process, it constitutes algorithmic bias. A simple way to deal with this problem can be to remove the sensitive attribute such as race before training the machine learning (ML) model. However, algorithmic bias can still persist via proxy variables such as zipcode that are correlated with the sensitive attribute. Recent years have seen a huge surge in research papers that deal with this problem. These papers have largely focused on pure algorithmic means to detect and remove bias at different stages of the ML pipeline. However, fairness is contextual and thus cannot be achieved using fully automated methods [40]. Moreover, existing techniques are largely black boxes, offering only limited insight on the proxy variables and how bias is mitigated. Finally, they

• *Bhavya Ghai and Klaus Mueller are affiliated to the Computer Science department, Stony Brook University. E-mail: {bghai, mueller}@cs.stonybrook.edu.*

Manuscript received 31 March 2022; revised 1 July 2022; accepted 8 August 2022. Date of publication 26 September 2022; date of current version 2 December 2022. This article has supplementary downloadable material available at https://doi.org/10.1109/TVCG.2022.3209484, provided by the authors.
Digital Object Identifier no. 10.1109/TVCG.2022.3209484

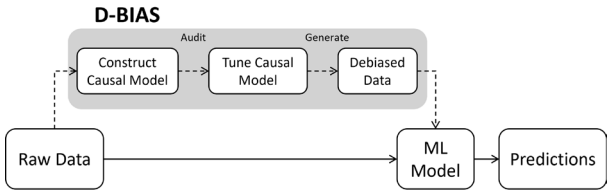

Fig. 2. This figure shows how D-BIAS fits into the typical ML pipeline. D-BIAS allows users to act on the raw data and returns its debiased version which can be fed into a ML model for fairer predictions.

are also limited in providing capabilities that allow users to actively steer and control the debiasing process. Given this limited transparency and human control, accountability and trust become a major concern.

To address these needs, we hypothesize that a human-in-the-loop (HITL) approach is the way forward. A human expert can determine what fairness means in a given context. Such domain knowledge can be incorporated effectively via the HITL approach and hence improve perceived fairness. Introducing a human into the loop will only be effective when a person can understand the underlying state of the system and provide useful feedback. Hence, this approach is naturally inclined towards interpretability. On the trust aspect, people are more likely to trust a system if they can tinker with it, even if it meant making it perform imperfectly [14]. Human interaction is a core part of the HITL approach, so it might instill more trust. Lastly, this approach should also foster accountability as the human has a much bigger role to play, which can significantly impact the results.

We present D-BIAS, a visual interactive tool that embodies a HITL approach for bias identification and mitigation. Given a tabular dataset with meaningful column names as input, D-BIAS assists users in auditing the data for different biases and then helps generate its debiased version (see Fig. 2). It uses a graphical causal model as a medium for users to visualize the causal structures inherent in the data and to inject their domain knowledge. We have made use of causal models since discrimination is inherently causal, and causal models can also be easy to interpret [24, 46, 49]. Apart from causal model, D-BIAS also includes multiple statistical fairness metrics to help identify bias. Users can choose to compare between two groups based on a single variable say gender (Male, Female) or a combination of attributes say race and gender (Black Females, White Males). Thereafter, they can inject their domain knowledge by acting on the edges of the causal network, for instance by deleting or weakening biased causal edges. Since the causal model encodes the data-generating mechanism, any user intervention modifies that process. Following each change, the system generates a new dataset based on the current causal model while keeping track and visualizing the impact of the user interventions on utility, data distortion and various fairness metrics. Users can interact with the system until they are satisfied with the outcome and then download the debiased dataset for use in any downstream ML application to achieve fairer predictions. The major contributions of our work are:

- A novel human-in-the-loop method to debias tabular datasets.
- An end-to-end visual interactive tool for algorithmic bias identification and mitigation.
- A demonstration of the effectiveness of our tool in reducing bias using three datasets.
- A user study to evaluate our tool on human-centric measures like usability, trust, interpretability, accountability, etc.

2 Background and Related Work

2.1 Bias Identification

The existing literature on bias identification mostly revolves around different fairness metrics. Numerous fairness metrics have been proposed which capture different facets of fairness, such as group fairness, individual fairness, counterfactual fairness, etc. [4, 5, 16, 28, 33]. Another way to classify fairness metrics can be on the level they operate on. For eg., dataset based metrics are solely computed using the dataset and are independent of any ML model, such as statistical parity difference. On the other hand, classifier based metrics are computed over the predictions of a trained ML model, such as false negative rate difference. So far, there is not a single best fairness metric. Moreover, some of the fairness metrics can be mutually incompatible, i.e., it is impossible to optimize different metrics simultaneously [27]. In line with existing visual tools [5, 46], our tool also uses a diverse set of fairness metrics to present a more comprehensive picture.

Many fairness metrics solely focus on the aggregate relationship between the sensitive attribute and the output variable. This can lead to misleading conclusions as the aggregate trend might disappear or reverse when accounting for other relevant factors. A prime example of this phenomenon, also known as Simpson's paradox [35], is the Berkeley Graduate Admission dataset [6]. There it appeared as if the admission process was biased against women since the overall admit rate for men (44%) was significantly higher than for women (30%) [2]. However, this correlation/association did not account for the fact that women typically applied for more competitive departments than men. After correcting for this factor, it was found that the admission process had a small but statistically significant bias *in favor of* women [6]. Causal models can be an effective tool for dealing with such a situation as they can decipher the different intermediate factors (indirect effects) along with their respective contributions behind an aggregate trend. Hence, our tool also employs causal model for bias identification.

2.2 Bias Mitigation

The existing literature on algorithmic bias mitigation can be broadly categorized into the three different stages in which they operate within the ML pipeline, namely pre-processing, in-processing and post-processing. In the pre-processing stage, the dataset is modified such that the bias with respect to an attribute or set of attributes is reduced or eliminated [10, 22, 25, 36, 48]. This can be achieved by either modifying the output label [25, 26] or by modifying the input attributes [10, 48]. In the in-processing stage, the algorithm is designed to take in biased data but still generate unbiased predictions. This can be achieved by tweaking the learning objectives such that accuracy is optimized while discrimination is minimized [3]. Finally, in the post-processing stage, predictions from ML algorithms are modified to counter bias [29]. Our work relates closely with the pre-processing stage where we make changes to the input attributes and the output label.

There is also a growing set of work at the intersection of bias mitigation and causality [11, 43, 50]. The general idea is to generate the causal network, modify it, and then simulate debiased data. These approaches fully rely on automated techniques to yield the true causal network, and assume a priori knowledge about fair/unfair causal relationships. Our work draws inspiration from this line of work and presents a more general solution where human domain knowledge is leveraged to refine the causal network and generate the debiased dataset.

2.3 Visual Tools

Recent years have seen visual tools like *Silva* [46], *FairVis* [8], *FairRankVis* [44], *DiscriLens* [42], *FairSight* [1], *WordBias* [18], etc. which are all aimed at tackling algorithmic bias. Although most of these tools are focused on bias identification, a few of them, such as FairSight, also permit debiasing. However, the debiasing strategy used in such tools is fairly basic, like eliminating proxy variable(s). Simple measures like this can lead to high data distortion and can have a high negative impact on data utility. Our work relates more closely with *Silva* which also features a graphical causal model in its interface. Using an empirical study, it showed that users can interpret causal networks and found them helpful in identifying algorithmic bias. However like most other visual tools, Silva is limited to bias identification. Our work advances the state of the art by presenting a tool that supports both bias identification and mitigation, with our debiasing strategy being more nuanced and sensitive toward data distortion.

3 Methodology

The workflow of our system can be understood from Fig. 3. A detailed discussion for each stage follows next.

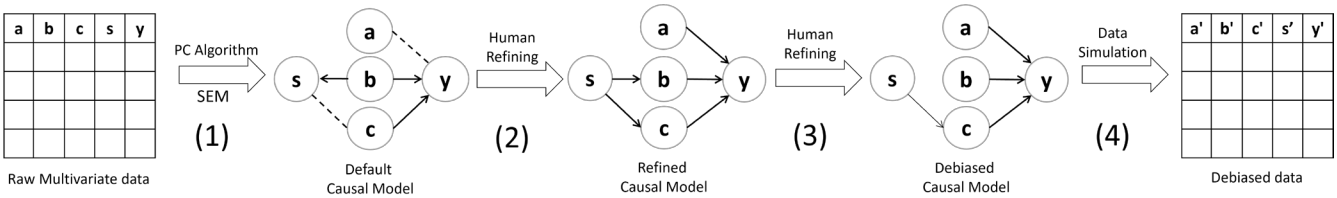

Fig. 3. The workflow of D-BIAS has 4 parts: (1) The default causal model is inferred using the PC algorithm and SEM. Here, dotted edges represent undirected edges (2) Refine stage - the default causal model is refined by directing undirected edges, reversing edges, etc. (3) Debiasing stage - the causal model is debiased by deleting/weakening biased edges (4) The debiased causal model is used to simulate the debiased dataset.

3.1 Generating the Causal Model

A causal model generally consists of two parts: causal graph (skeleton) and statistical model. Given a tabular dataset as input, we first infer an initial causal graph (directed acyclic graph) using a popular causal discovery algorithm, namely the PC algorithm [13, 39]. This algorithm relies on conditional independence tests and a set of orientation rules to yield the causal graph (for details, see appendix B). Each node in the causal graph represents a data attribute and each edge represents a causal relation. For example, a directed edge from X to Y signifies that X is the cause of Y, i.e., a change in X will cause a change in Y. The PC algorithm provides qualitative information about the presence of causal relationship between variables. To quantitatively estimate the strength of each causal relationship, we use linear structural equation models (SEM) where a node is modeled as a linear function of its parent nodes.

$$y = \sum_{i}^{parents(y)} \beta_i x_i + \varepsilon \tag{1}$$

In the above equation, variable y is modeled as a linear combination of its parents x_i, their regression coefficients (β_i) and the intercept term (ε). If y is a numerical variable, we use linear regression else we use the multinomial logit model to estimate the values of β_i and ε. Here, β_i represents the strength of causal relation between x_i and y. We repeat this modeling process for each node with non-zero parent nodes. Such nodes that have at least one edge leading into them are termed as endogenous variables. Other nodes correspond to exogenous variables or independent variables that have no parent nodes. In Fig. 3 (Refined Causal Model), s and a are exogenous variables while b, c and y are endogenous variables. Here, y will be modeled as a function of its parent nodes, i.e., a, b and c. Similarly, b and c will be modeled as function of s. After this process, we arrive at a causal graph whose different edges are parameterized using SEM. This constitutes a Structural Causal Model (SCM) or simply causal model (see Fig. 3 (1)).

This causal model, generated using automated algorithm, might have some undirected/erroneous causal edges due to different factors like sampling bias, missing attributes, etc. To achieve a reliable causal model, we have taken a similar approach as Wang and Mueller [41] where we leverage user knowledge to refine the causal model via operations like adding, deleting, directing, and reversing causal edges (see Fig. 3 (2)). For every operation, the system computes a score (Bayesian Information Criterion (BIC)) of how well the modified causal model fits the underlying dataset. Similar to [41], our system assists the user in refining the causal model by providing the difference in BIC score before and after the change. A negative BIC score suggests a better fit. After achieving a reliable causal model, we enter into the debiasing stage where any changes made to the causal model will reflect on the debiased dataset (see Fig. 3 (3)).

3.2 Auditing and Mitigating Social Biases

From a causal perspective, discrimination can be defined as an unfair causal effect of the sensitive attribute (say, race) on the outcome variable (say, getting a loan) [34]. A direct causal path from a sensitive variable to the output variable constitutes disparate treatment while an unfair indirect path via some proxy attribute (say, zipcode) constitutes disparate impact [12]. A direct path is certainly unfair but an indirect path may be fair or unfair (as in the case of Berkeley Admission dataset [35]). Our system computes all causal paths and lets the user decide if a causal path is fair or unfair. If a causal path is unfair, the user should identify which constituting causal relationship(s) are unfair and act on them. The user can do this by deleting or weakening such biased causal relationships to reduce/mitigate the impact of the sensitive attribute on the outcome variable. For example, in Fig.3 (Refined Causal Model), s is the sensitive attribute and y is the outcome variable. Here, the user deletes the edge $s \rightarrow b$ and weakens the edge $s \rightarrow c$ (shown as a thin edge). Once the user has dealt with the biased causal relationships, we achieve what we call the *Debiased Causal Model*.

Algorithm 1 Generate Debiased Dataset

$D \leftarrow$ Original Dataset
$G(V,E) \leftarrow$ refined causal model
$E_a \leftarrow$ set of edges added during the debiasing stage
$E_m \leftarrow$ subset of E that were deleted/strengthened/weakened during the debiasing stage

for each e in E_a **do**
  $n \leftarrow$node pointed by head of e
  retrain linear model for n as a function of its parents
end for

$V_{sim} \leftarrow \emptyset$ // Attributes that need to be simulated
for each e in $(E_m \cup E_a)$ **do**
  $n \leftarrow$node pointed by head of e
  $V_{sim} \leftarrow V_{sim} + n + all\_descendent\_nodes(n)$
end for
$V_{sim} \leftarrow remove\_duplicates(V_{sim})$

$D_{deb} \leftarrow \emptyset$ // Debiased Dataset
for each v in topological_sort(V) **do**
  if v present in V_{sim} **then**
    $D_{deb}[v] \leftarrow$ generate values based on Equation 2
  else
    $D_{deb}[v] \leftarrow D[v]$
  end if
end for

// Rescale values for the simulated attributes
for each v in V_{sim} **do**
  if v is a categorical variable **then**
    $D_{deb}[v] \leftarrow$ rescale values based on Algorithm 2
  else
    // v is a numeric variable
    $\mu, \sigma^2 \leftarrow mean(D[v]), variance(D[v])$
    $\mu_{deb}, \sigma^2_{deb} \leftarrow mean(D_{deb}[v]), variance(D_{deb}[v])$
    $D_{deb}[v] \leftarrow \mu + (D_{deb}[v] - \mu_{deb})/\sigma_{deb} * \sigma$
  end if
end for
Result D_{deb}

3.3 Generating the Debiased Dataset

We simulate the debiased dataset based on the debiased causal model (see Fig. 3 (4)). The idea is that if the user weakens/removes biased edges from the causal network, then the simulated dataset might also contain less biases. The standard way to simulate a dataset based

Algorithm 2 Rescale Categorical Variable v

```
1: D[v] ← categorical variable v in the original dataset
2: prob_mat ← probability matrix for v computed using Eq. 2
3:
4: lr ← 0.1                                              // learning rate
5: iterations ← 0
6: loop
7:    // DPD: Discrete Probability Distribution
8:    dist_ori ← DPD( D[v] )
9:    dist_deb ← DPD( argmax(prob_mat) )
10:   diff ← ∑ ||(dist_ori − dist_deb)/dist_deb||
11:   scale_factor ← 1 + lr * ∑ (dist_ori − dist_deb)/dist_deb)
12:   prob_mat ← scale_factor * prob_mat
13:   dist_deb ← DPD( argmax(prob_mat) )
14:   new_diff ← ∑ ||(dist_ori − dist_deb)/dist_deb||
15:   if new_diff > diff or iterations > 50 then
16:      break
17:   end if
18:   iterations ← iterations + 1
19: end loop
20: D_deb[v] ← argmax(prob_mat)
21: Result D_deb[v]
```

on a causal model is to generate random numbers for the exogenous variables [38]. Thereafter, each endogenous variable is simulated as a function of its parent nodes (variables) in the causal network. In this work, we have adapted this procedure to suit our needs, i.e., simulating a fair dataset while having minimum distortion from the original dataset.

Our approach to generate the debiased dataset, as illustrated in Algorithm 1, can be broken down into 4 steps. At step 1 (lines 6–9), we focus on the set of edges added during the debiasing stage (E_a). We retrain regression models corresponding to each of the target nodes of E_a. This will update the weights (regression coefficients (β_i)) for all edges leading into any target node of E_a. At step 2 (lines 11–16), we identify the set of nodes (attributes) that need to be simulated (V_{sim}). Unlike the standard procedure, we only simulate selective nodes that are directly/indirectly impacted by the user's interaction to minimize distortion. This set includes the target nodes of all edges that the user has interacted with along with their descendent nodes. For example, in Fig. 3(3), the user deletes the edge $s \rightarrow b$ and weakens the edge $s \rightarrow c$. So, we will only simulate variables b, c and y. At step 3 (lines 18–25), we actually simulate all nodes that are a part of V_{sim} using Equation 2. All other nodes are left untouched and their values are simply copied from the original dataset into the debiased dataset. It should be noted that all parent nodes must be simulated before their child node as the values for a node are computed using their parent nodes. So, we simulate all endogenous variables in a topological order. For eg., in Fig. 3(4), nodes b and c will be simulated before node y.

$$y = \sum_{i}^{parents(y)} \alpha_i \beta_i x_i \; + \; \varepsilon \; + \sum_{i}^{parents(y)} (1-\alpha_i)\beta_i r_i \qquad (2)$$

In the above equation, node y is simulated as a sum of 3 terms. The first term is the weighted linear combination of parent nodes. Here, α_i is the scaling factor that has a default value of 1 and can range between [0, 2] as determined by user interaction. Strengthening an edge, sets $\alpha_i > 1$; weakening an edge, sets $\alpha_i < 1$. For example, if the user weakens the edge between node x_i and y by -35%, then α_i=0.65, strengthening it by +35% will set α_i=1.35, and deleting it will set α_i=0. The second term is the intercept that was computed when the regression model for y was last trained. The third term adds randomness in proportion to the degree to which the user has altered an incoming edge. Here, r_i is a normal random variable that has a similar distribution as x_i ($r_i \sim \mathcal{N}(\mu_{x_i}, \sigma^2_{x_i})$). This term adds fairness as it is random and alleviates distortion for y.

Fig. 4 illustrates the case where the node *Job* has a single parent node (*Gender*). Let's say that the user chooses to delete this edge (α=0). Going the conventional route (without the third term), the attribute *Job* will be reduced to a constant value (the intercept (ε)) which is undesirable. Adding the (third) random term generates the *Job* distribution below the '+' node which is far more balanced (fair) in terms of *Gender* than the 'Original' distribution on the top right. However, the number of people getting the job (or not) is distorted compared the 'Original' distribution.

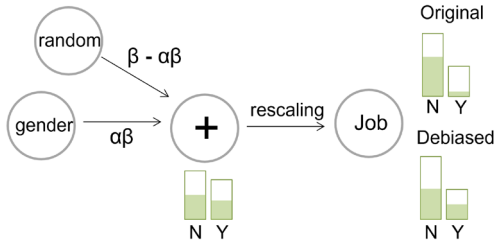

Fig. 4. Illustration of the data debiasing process using a toy example where the node Job is caused by a single variable, namely gender. Green color marks the proportion of females who got the job (Y) or not (N).

This is corrected in Step 4 (lines 27–37) in Algorithm 1, where we rescale each simulated variable so that its distribution is close to its original distribution. For numerical variable(s), we simply standardize values to their original mean and standard deviation. For categorical variable(s), the model returns a set of probability scores corresponding to each possible output label for each data point. We iteratively scale such a probability matrix so that the resulting debiased distribution inches toward the original distribution (see Algorithm 2). We continue this process until a fixed number of iterations or when the difference between original and debiased distribution starts increasing. In Fig. 4, we observe that the resulting 'Debiased' distribution matches the 'Original' distribution in terms of *Job* allocation, while maintaining gender parity. It should be noted that simulating each attribute adds a corresponding modeling error to the process. This modeling error is typically small but it can potentially overpower the impact of the user's intervention, especially when a user makes a small change, say weakening an edge by 5%. In such a case, the results may not be in strict accordance with the user's expectations.

3.4 Evaluation Metrics

Once the debiased dataset is generated, it can be evaluated using different metrics that operate at the dataset level and the classifier level. For the second case, the debiased dataset is used to train a ML model chosen by the user, and a set of metrics are computed over the model's predictions. Here, the idea is to evaluate the downstream effects of debiasing. All the evaluation metrics can be grouped into three broad categories, namely utility, fairness and distortion. It should be noted that there might be a trade-off among the three categories. For eg., reducing bias might cause high data distortion or lower utility. For comparison, we have used a baseline debiasing strategy which just removes the sensitive attribute(s) from the dataset.

Fairness. Our tool presents a diverse set of 5 popular fairness metrics, namely statistical parity difference (Parity diff), individual fairness (Ind. bias), accuracy difference (Accuracy diff), false negative rate difference (FNR diff) and false positive rate difference (FPR diff) [5]. Two of these metrics operate at the dataset level (Parity diff, Individual bias) and the rest operate on the classifier's predictions. Here, Ind. bias is defined as the mean percentage of a data point's k-nearest neighbors that have a different output label. A lower value for Ind. bias is desirable as it means that similar data points have similar output labels. For the 4 other fairness metrics, we compute some statistic for the two groups say males and females, and then report their absolute difference. This statistic can be ML model dependent, such as accuracy, or model independent, such as the likelihood of getting a positive output label. Lower values for such metrics suggests more equality between groups. For computing model based metrics, we omit the sensitive attributes(s) and perform 3-fold cross validation using the user-specified ML model with 50:50 train test split ration, and then report the mean absolute difference between groups across the 3 folds.

Utility. The utility of ML models is typically measured using met-

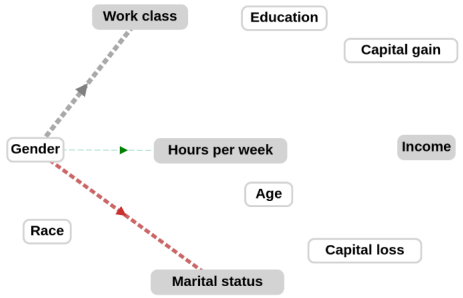

Fig. 5. Logs view highlighting the changes made to the causal network for the Adult Income dataset. Dotted lines represent deleted edges and nodes in grey represent the impacted nodes.

rics like accuracy, F1 score, etc. In our context, we are interested in measuring the utility of a ML model when it is trained using the debiased dataset instead of the original dataset. To compute the utility for the original dataset, we perform 3-fold cross validation using the user-specified ML model and report the mean accuracy and F1 score. For the debiased dataset, we follow a similar procedure where we train the user-specified ML model using the debiased dataset. However, we use the outcome variable from the original dataset as the ground truth for validation. Sensitive attribute(s) are removed from both datasets before training. Ideally, we would like the utility metrics for the debiased dataset to be close to the corresponding metrics for the original dataset.

Data Distortion. Data distortion is the magnitude of deviation of the debiased dataset from the original dataset. Since the dataset can have a mix of continuous and categorical variables, we have used the *Gower distance* [21] metric. We compute the distance between corresponding rows of the original and debiased dataset, and then report the mean distance as data distortion. This metric is easy to interpret as it has a fixed lower and upper bound ([0,1]). It is 0 if the debiased dataset is the same as the original while higher values signify higher distortion. Lower values for data distortion are desirable.

4 The D-BIAS Tool

4.1 Generator Panel

This is the first component of the tool the user interacts with (see Fig.1 (A)). The user starts off by choosing a dataset from its dropdown menu. The user then selects the label variable which should be a binary categorical variable as we are considering a classification setting. Next, the user selects all nominal variables which is required for fitting the SEM model. Lastly, the user chooses a p-value and clicks the 'Causal Model' button to generate the causal network. Here, the p-value is used by the PC algorithm to conduct independence tests. We set $p = 0.01$ for all our demonstrations. It can be changed to 0.05 or 0.10 for smaller datasets. The 'Debiased Data' button downloads the debiased dataset.

4.2 Causal Network View

This is the most critical piece of the interface where the user will spend most of the time (see Fig.1(B)). The center of this view contains the actual causal network which is surrounded by 4 panels on all sides.

Causal Network. All features in the dataset are represented as nodes in the network and each edge represents a causal relation. The width of an edge encodes the magnitude of the corresponding standardized beta coefficient. It signifies the feature importance of the source node in predicting the target node. The color of an edge encodes the sign of the corresponding standardized beta coefficient. Green (red) represents positive (negative) influence of the source node on the target node. If an edge is undirected, it does not have a beta coefficient and is colored orange. Finally, gray color encode edges that represent relationships which can not be represented by a single beta coefficient. This occurs when the target node is a categorical variable with more than 2 levels.

The causal network supports many interactions to enhance the user's overall experience and productivity. It supports operations like zooming and panning. The user can move nodes around if they are not satisfied with the default layout. On clicking a node, all directly connected edges and nodes are highlighted. Similarly, on clicking an edge, its source and target nodes are highlighted. Moreover, clicking a node or an edge also visualizes their distribution in the *Comparison View* (see Sect. 4.3).

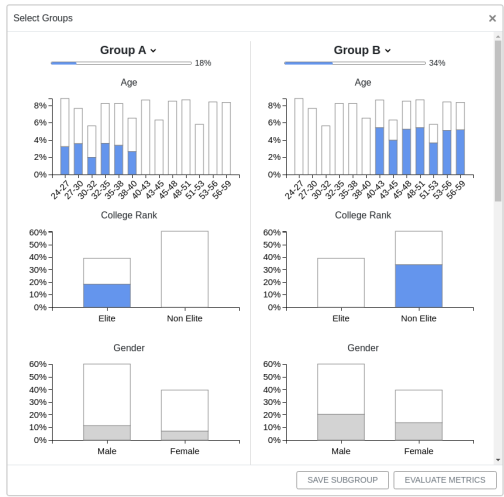

Fig. 6. The visual interface for selecting subgroups (Group A and B). Each column consists of a list of bar charts/histograms representing all attributes in the dataset. By default, all bars are colored gray. The user can click on multiple bars to select a subgroup. Selected bars are colored blue. Each bar is filled in proportion of their representation in the selected subgroup as a ratio of the entire dataset. In this picture, Group A consists of individuals who went to elite universities and whose age lies between 24-40. It represents 18% of the dataset.

Top panel. The panel right above the causal network (as shown above) allows the selection of an edge by choosing the source and target nodes. Next to the dropdown menus are a series of buttons which allow a user to inject their prior into the system. Going from left to right, they represent operations like adding an edge, deleting an edge, reversing the edge direction and directing an undirected edge. The toggle at the end represents the current stage (Refine/Debias) and helps transitioning from one to the other.

Left panel. The bar to the left of the causal network shows the change in BIC score. This bar is updated each time the user performs operations like directing an undirected edge, adding/deleting an edge, etc. A negative value means that the change made to the causal network is in sync with the underlying dataset; positive values mean the opposite. Negative (positive) values are represented in green (red).

Right panel. The panel to the right of the causal network offers four functionalities. Going from top to bottom, they represent zooming in, zooming out, reset layout and changing weight of an edge. The slider at the bottom gets activated when the user clicks on an edge during the debiasing stage. It allows the user to weaken/strengthen an edge depending on the selected value between -100% to 100%. Moving the slider changes α_i and also impacts the effective beta coefficient for the selected edge ($\alpha_i \beta_i$). This change manifests visually in the form of proportional change in the corresponding edge width. Moving the slider to -100% will result in deletion of the selected edge.

Bottom panel. As shown below, the bottom panel offers 4 functionalities. The "Find Paths" button triggers depth first traversal of the causal graph to compute all directed paths between the source and target node as selected in the top panel. This will be especially helpful when the graph is big and complex. All the computed paths are then displayed below the bottom panel. A user can click on a displayed path to highlight it and see an animated view of the path going from the source to the target node. The "Logs" button highlights changes made to the causal network during the debiasing stage (see Fig.5). All edges are hidden except for the newly added edges, deleted edges and edges that were weakened/strengthened. Nodes impacted by such operations

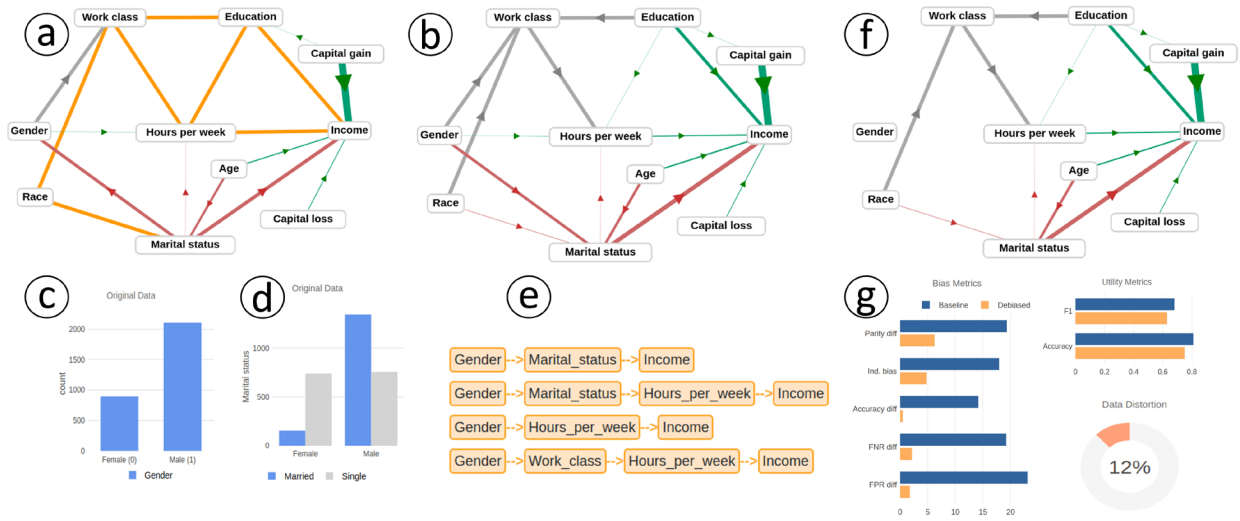

Fig. 7. Case study: Adult Income dataset (a) Causal model generated using automated techniques (b) Refined causal model (c) Clicking the Gender node visualizes its distribution as a bar chart (d) Bivariate distribution between Gender and Marital Status (e) All paths from Gender to Income in the refined causal model (f) Debiased causal model (g) Evaluation metrics to compare our results against the baseline debiasing approach.

(V_{sim}) are highlighted in grey.

If a user is interested in knowing the exact beta coefficients for all edges, the edge weights toggle will help in doing just that. By default, it is set to 'hide' to enhance readability. If turned to 'show', the beta coefficients will be displayed on each edge. The filter slider helps user focus on important edges by hiding edges with absolute beta coefficients less than the chosen value.

4.3 Evaluation Panel

This panel, at the right (see Fig.1(C)), provides different options and visual plots for comparing and evaluating the changes made to the original dataset. From the top left, users can select the sensitive variable and the ML algorithm from their respective dropdown menus. This selection will be used for computing different fairness and utility metrics. For the sensitive variable, the dropdown menu consists of all binary categorical variables in the dataset along with a *Custom Group* option. Selecting the *Custom Group* option opens a new window (see Fig. 6) which allows the user to select groups composed of multiple attributes. This interface facilitates comparison between intersectional groups say black females and white males.

Clicking the "Evaluate Metrics" button triggers the computation of evaluation metrics that are displayed on the right half of this panel (Performance View). It also visualizes the relationship between the sensitive attribute or selected groups and the outcome variable using a 4-fold display [17] on the left half of this panel (Comparison View).

Comparison View. This view comprises two plots aligned vertically where the top plot represents the original dataset and the bottom plot represents the debiased dataset. This view has two functions. It first aids the user in the initial exploration of different features and relationships. When a node or edge is clicked in the causal network, the summary statistics of the corresponding attributes is visualized (see Fig. 7(c) for an example). For binary relationships, we use either a scatter plot, a grouped bar chart or an error bar chart depending on the data type of the attributes. The second function of the Comparison View is to visualize the differences between the original and the debiased dataset. Initially, the original data is the same as the debiased data and so identical plots are displayed. However, when the user injects their prior into the system, the plots for the original and debiased datasets start to differ. We added this view to provide more transparency and interpretability to the debiasing process and also help detect sampling bias in the data.

When the user clicks the "Evaluate Metrics" button, the Comparison View visualizes the binary relation between the sensitive attribute or selected groups and the outcome (label) variable via the 4-fold plot [17] as shown in Fig.1(C), left panel). We chose a 4-fold display over a more standard brightness-coded confusion matrix since the spatial encoding aids in visual quantitative assessments. The left/right half of this display represents two groups based on the chosen sensitive variable (say males and females) or as defined in the Custom Group interface, while the top/bottom half represents different values of the output variable say getting accepted/rejected for a job. Here, symmetry along the y-axis means better group fairness.

Performance View. This view houses all the evaluation metrics as specified in Sec.3.4. It uses a horizontal grouped bar chart to visualize 2 utility and 5 fairness metrics. Lower values for the fairness metrics mean better fairness. Higher values for utility metrics means better utility. Data distortion is visualized using a donut chart. On hovering over any of these charts, a tooltip shows the exact values.

5 Case Study

We demonstrate the utility of our tool for bias identification and mitigation using the Adult Income dataset. Each data point in the dataset represents a person described by 14 attributes recorded from the US 1994 census. Here, the prediction task to classify if a person's income will be greater or less than $50k based on attributes like age, sex, education, marital status, etc. We chose this dataset as it is widely used in the algorithmic fairness literature [3, 10, 20]. Here, we have chosen a random sample of 3000 points from this dataset for faster computation.

Generating the causal network. We start off by selecting the Adult Income dataset from the respective dropdown menu in the Generator panel. We select *Income* as the label variable and *Work class*, *Marital Status*, *Race*, *Gender*, *Income* as the nominal variables. Next, we click on the *Causal Model* button which generates the default causal model (see Fig. 7 (a)). Here, we examine different edges of the causal model and act on them as needed to reach to a reliable causal model. We start with the 7 undirected edges encoded in orange. We direct edges based on our domain knowledge like *Hours per Week* $\rightarrow$ *Income*, *Education* $\rightarrow$ *Income*, *Education* $\rightarrow$ *Hours per week*, etc. After each of these operations, we observe a green bar in the left panel of the Causal Network View. This indicates that the resulting causal model is a better fit over the underlying dataset. Next, we examine other directed edges. Many of them align with our domain knowledge like *Capital Gain* $\rightarrow$ *Income*, *Age* $\rightarrow$ *Income*, etc. However, we found a couple of them to be counter-intuitive, namely *Capital Gain* $\rightarrow$ *Education* and *Marital Status* $\rightarrow$ *Gender*. In a causal relation, cause always precedes effect. Hence, immutable personal characteristics like sex, race, etc., which are assigned at birth, can not be the effect of a later life event like marriage or work class. So, in this case, we chose to reverse both these edges to

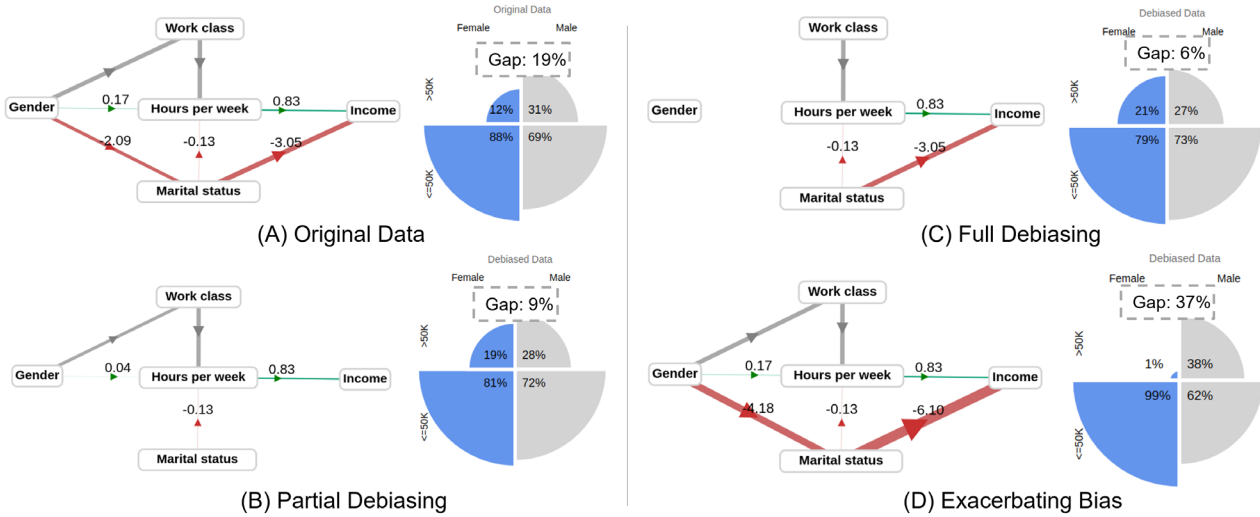

Fig. 8. The above picture shows the impact of 3 types of user interaction as captured by the 4-fold display. Due to space constraints, we have only shown a subset of the causal network which connects the Gender node with the Income node. For details, please refer to the description in Section 5.

get to the refined causal model (see Fig. 7(b)).

Auditing for social biases. Once we have reached a reliable causal model, we start auditing for different kinds of biases. We click on different nodes and edges to explore their distributions. For eg., clicking the *Gender* node visualizes its distribution in the *Comparison View*. We observe that females are underrepresented in the dataset (895 females vs 2105 males) (see Fig. 7 (c)). Given this representation bias and the fact that gender pay gap is a well-known issue, we decided to investigate further. We found an indirect path from *Gender* to *Income* via *Hours per week*. This indicates a possible disparity in income based on gender. To probe further, we click the "Evaluate Metrics" button with *Gender* as the sensitive variable to compute different fairness metrics. We observe significant gender bias as captured by metrics like Accuracy diff (14%), FPR diff (22%), FNR diff (17%), etc. The 4-fold display for the original data in Fig. 8 (A) reveals that only 12% of the females earn more than \$50k compared to 31% for males. Thus there is a significant income disparity based on gender. To have a more comprehensive understanding of the issue, we search for all possible paths by selecting *Gender* and *Income* as the source and target from the Top panel and then clicking the "Find Paths" button from the bottom panel. This populates 4 different causal paths below the bottom panel (see Fig. 7 (e)). We will now focus on the different causal relationships in these paths and try to make minimal changes to achieve more fairness.

Bias mitigation. To mitigate gender bias, we first enter the debiasing stage by flipping the Stage toggle from Refine to Debias. From here on, any changes to the causal network will simulate a new debiased dataset. Among the 4 paths we previously discovered, the top 2 paths have a common causal edge, i.e., *Gender* $\rightarrow$ *Marital status*. On clicking this edge, we find that most males in the dataset are married while most females are single (see Fig. 7(d)). This pattern indicates sampling bias. Ideally, we would like no relation between these attributes so we delete this causal edge. Next, we assess the remaining two causal paths. Based on our domain knowledge, we find the causal edge *Hours per week* $\rightarrow$ *Income* to be socially desirable, and the edges *Gender* $\rightarrow$ *Work class* and *Gender* $\rightarrow$ *Hours per week* to be socially undesirable. We delete the two biased edges to get to the debiased causal model (see Fig. 7(f)). To verify all changes made so far, we click on the *Logs* button. As shown in Fig. 5, it shows 3 dotted lines for the removed edges and highlight the impacted attributes.

Lastly, we click on the "Evaluate Metrics" to see the effect of our interventions. The 4-fold display for the debiased data in Fig. 8(C) shows that the disparity between the two genders has now decreased from 19% to 6% (compare Fig. 8(A)). The percentage of females who make more than \$50k have undergone a massive growth of 75% (from 12% to 21%). Moreover, as shown in Fig. 7(g), we find that all fairness metrics have vastly improved, with only a slight decrease in the utility metrics and an elevated distortion (12%). These results clearly indicate the efficacy of our debiasing approach.

Partial debiasing. Considering the tradeoff between different metrics, one might choose to debias data partially based on their context, i.e., weaken biased edges instead of deleting them or keeping certain unfair causal paths from the sensitive variable to the label variable intact. For eg., one might choose to delete the edge *Gender* $\rightarrow$ *Marital status* and weaken the edges *Gender* $\rightarrow$ *Work class* and *Gender* $\rightarrow$ *Hours per week* by 25% and 75%, respectively. On evaluation, we find this setup to sit somewhere between the original and the fully debiased case (see Fig. 8 (B)). It performs better on fairness than the original dataset (gap: 9% vs 19%) but worse than the full debiased version (gap: 9% vs 6%). Similarly, it incurs more distortion than the original dataset but less than the full debiased version (11% vs 12%).

Intersectional groups. D-BIAS facilitates auditing for biases against intersectional groups using the "Custom Group" option from the sensitive variable dropdown. Here, we choose *Black Females* and *White Males* as the two groups. At the outset, there is a great disparity between the groups as reflected in the 4-fold display and the fairness metrics (see Fig. 1). As these subgroups are defined by *Gender* and *Race*, we focus on the unfair causal paths from these nodes to the label variable (*Income*). For debiasing, we first perform the same operations we did for gender debiasing. Thereafter, we reduce the impact of race by deleting the edges *Race* $\rightarrow$ *Work class* and *Race* $\rightarrow$ *Marital status* which we deem as socially undesirable. On evaluation (see Fig. 1), we find a significant decrease in bias across all fairness metrics for the debiased dataset compared to the conventional debiasing practice (blue bars) which just trains the ML model with the sensitive attributes (here *Gender* and *Race*) simply removed. Finally, the two 4-fold displays reveal that the participation of the disadvantaged group more than doubled, while the privileged group experienced only a modest loss.

Exacerbating bias. The flexibility offered by D-BIAS to refine the causal model can be misused to increase bias as well. Bias can be exacerbated by strengthening/adding biased causal edges and weakening/deleting other relevant causal edges. For eg., one can exacerbate gender bias by strengthening the edges *Gender* $\rightarrow$ *Marital status* and *Marital status* $\rightarrow$ *Income* by a 100%. On evaluation, we find that the proportion of females making >\$50k has shrunk to just 1% while the proportion of males has surged to 38%. In effect, this has broadened the gap between males and females making more than \$50k by about 2x from 19% to 37% (see Fig. 8 (D)).

Results. Apart from the Adult Income dataset, we also tested our tool using the synthetic hiring dataset and the COMPAS recidivism dataset (see appendix D for details). The evaluation metrics for all 3 datasets after full debiasing are reported in Table 1. As we can observe, our tool is able to reduce bias significantly compared to the

Table 1. Evaluation metrics to compare the debiased dataset generated using our tool against the baseline debiasing approach for different datasets.

Dataset	Sensitive attribute	version	ML model	Accuracy	F1	Parity difference	Individual Bias	Accuracy difference	FNR difference	FPR difference	Data Distortion
Synthetic Hiring	Gender	baseline	SVM	77%	0.59	11.12	19.09	4.14	14.26	6.82	0%
		debiased		77%	0.60	1.66	12.93	2.99	1.37	3.63	6%
Adult Income	Gender	baseline	Logistic Regression	82%	0.69	19.32	17.92	14.35	17.98	22.53	0%
		debiased		75%	0.63	6.24	4.8	0.88	2.33	1.9	12%
COMPAS	Race	baseline	Random Forest	67%	0.64	12.07	33.9	0.17	23.07	16.89	0%
		debiased		63%	0.59	11.12	2.19	0.44	0.68	1.55	13%

baseline approach across the 3 datasets for a small loss in utility and data distortion. *These results validate the potential of HITL approach in mitigating bias*. It is interesting to observe that the F1 score for the synthetic hiring dataset is slightly higher than the baseline approach. However, this is line with the existing literature [20] where similar instances have been recorded.

6 User Study

We conducted a user study to evaluate two primary goals: (1) usability of our tool, i.e., if participants can comprehend and interact with our tool effectively to identify and mitigate bias, (2) compare our tool with the state of the art in terms of human-centric metrics like accountability, interpretability, trust, usability, etc.

Participants. We recruited 10 participants aged 24-36; gender: 7 Male and 3 Female; profession: 8 graduate students and 2 software engineers. The majority of the participants are computer science majors with no background in data visualization or algorithmic fairness. 80% of the participants trust AI and ML technologies in general. The participation was voluntary with no compensation.

Baseline Tool. For an even comparison, we looked for existing tools with a visual interface that support bias identification and mitigation. This led us to IBM's AI Fairness 360 [5] toolkit whose visual interface can be publicly accessed online[1]. However, we didn't go further with this toolkit as the baseline (control group) because it has a significantly different look and feel which is difficult to control for. Instead, we took inspiration from this toolkit and built a baseline visual tool (not to be confused with the baseline debiasing strategy) which mimics its workflow but matches the design of our D-BIAS tool (see Fig.9).

IBM's AI Fairness toolkit allows the user to choose from a set of fairness enhancing interventions with varying impact on the evaluation metrics. However, this study is focused on other important aspects such as trust, accountability, etc. So, in order to have a tightly controlled experiment, we imagine a hypothetical automated debiasing algorithm whose performance exactly matches the peak performance of our tool for all evaluation metrics. Here, peak performance refers to the state where all unfair causal edges are deleted.

Using the baseline tool is quite simple (see appendix E). The user first selects the dataset, label variable, etc. They can then audit for different biases by selecting the sensitive attribute and then clicking on the 'Check bias' button. This will compute and present a set of fairness metrics in the same fashion as the D-BIAS tool. Lastly, the user can click the 'Debias & Evaluate' button to debias the dataset and generate its evaluation metrics. A small lag is introduced before displaying evaluation metrics to mimic a real debiasing algorithm.

Study design. We conducted a within subject study where each participant was asked to use the baseline tool and D-BIAS in random order. The study was conducted remotely, i.e., each participant could access and interact with the tools via their own machine. For each tool, a small tutorial was given using the Synthetic Hiring dataset[2] to demonstrate the workflow and features of the tool. Each participant was then given some time to explore and interact with each system. Next, the participants were asked to identify and mitigate bias for the Adult Income dataset. For the D-BIAS tool, participants were also asked to complete a set of 5 tasks to evaluate usability. Each task was carefully designed to cover our testing goals and had a verifiable correct solution. Tasks included: generate a causal network, direct undirected edges, identify if bias exists with respect to an attribute, identify proxy variables and finally debias the dataset. After using each tool, the participants were asked to answer a set of survey questions. Lastly, we collected subjective feedback from each participant regarding their overall experience with both the tools. Throughout the study, participants were in constant touch with the moderator for any assistance. Participants were encouraged to think aloud during the user study.

Survey Design. Each participant was asked to answer a set of 13 survey questions to quantitatively measure usability, interpretability, workload, accountability and trust. All of these questions can be answered on a 5-point Likert Scale. To capture cognitive workload, we selected two applicable questions from the NASA-LTX task load index [23], i.e., "How mentally demanding was the task? and "How hard did you have to work to accomplish your level of performance?". Participants could choose between 1 = very low to 5 = very high. For capturing usability, we picked 3 questions from the System Usability Scale (SUS) [7]. For e.g., "I thought the system was easy to use", "I think that I would need the support of a technical person to be able to use this system". Participants could choose between 1 = Strongly disagree to 5 = Strongly agree. To capture accountability, we asked two questions based on previous studies [9, 19]. For e.g., "The credit/blame for mitigating bias effectively is totally due to" (1 = System's capability, 5 = My input to the system). To capture interpretability, we consulted Madsen Gregor scale [30] and adopted 3 questions for our application. For e.g., "I am satisfied with the insights and results obtained from the tool?", "I understand how the data was debiased?" Answers could lie between 1 = Strongly disagree to 5 = Strongly agree. For measuring trust, we referred to McKnight's framework on Trust [31, 32] and other studies [15, 19] to come up with 3 questions for our specific case. For e.g., "I will be able to rely on this system for identifying and debiasing data" (1 = Strongly disagree, 5 = Strongly agree).

Results. Despite not having a background in algorithmic fairness or data visualization, all participants were able to complete all 5 tasks using the D-BIAS tool. This indicates that our tool is easy to use.

The survey data was analyzed to calculate usability, interpretability, workload, accountability and trust ratings for each tool by each participant. The mean ratings along with their standard deviations are plotted in Fig.10. Using t-test, we found statistically significant differences for all measures with $p<0.05$. *We found that D-BIAS outperforms the baseline tool in terms of trust, accountability and interpretability.* However, it lags in usability and cognitive workload. So, if someone is looking for a quick fix or relies on ML algorithms more than humans, automated debiasing is the way to go. Conversely, if trust, accountability or interpretability is important, D-BIAS should be the preferred option. Looking at these results in conjunction with the results reported in Table 1, *we find that our tool enhances fairness while fostering accountability, trust and interpretability.*

Subjective feedback. After the study, we gathered feedback from

[1] https://aif360.mybluemix.net/data

[2] We generated a synthetic hiring dataset fraught with gender and racial bias to better evaluate our tool. For details, please refer to appendix C .

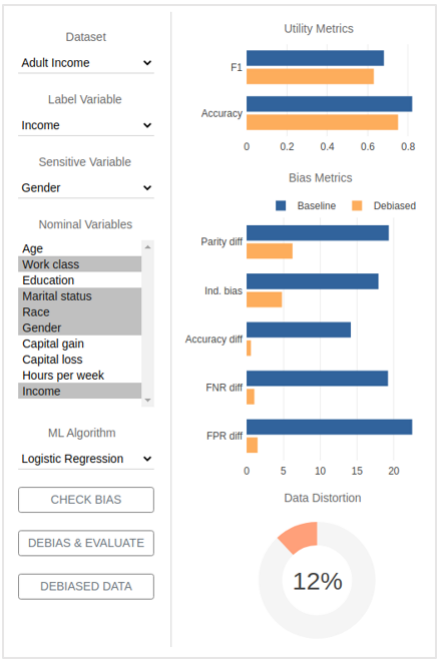

Fig. 9. Baseline visual tool used as a benchmark in the user study.

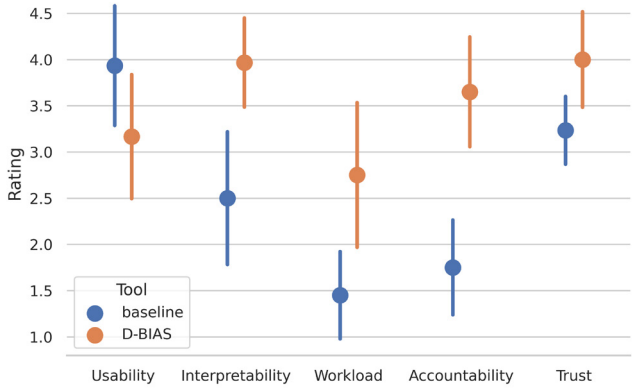

Fig. 10. Mean user ratings from the survey data along with their standard deviation for different measures.

each participant about what they liked or disliked about D-BIAS. Most participants liked the overall design, especially the causal network. We got comments like "*Interface is user friendly*", "*Causal network gives control and flexibility*", "*Causal network is very intuitive and easy to understand*", "*Causal network is a great way to understand relationships between features*". Most participants agreed that after a tutorial session, it should be fairly easy for even non-experts to play with the system. Another important aspect which received a lot of attention was our human-in-the-loop approach. Participants felt that they had a lot more control over the system and that they could change things around. One of the participants commented "*It feels like I have a say*". Some of the participants said they felt more accountable because the system offered much flexibility and that they had a choice to make.

Many of the participants strongly advocated for D-BIAS over the baseline tool. For e.g., "*D-BIAS better than automated debiasing any day*", "*D-BIAS hand's down!*". Few of the participants had a more nuanced view. They were of the opinion that the baseline tool might be the preferred option if someone is looking for a quick fix. We also received concerns and suggestions for future improvement. Two of the participants raised concern about the tool's possible misuse if the user is biased. Another participant raised concern about scalability for larger datasets. Most participants felt that adding tooltips for different UI components especially the fairness metrics will be a great addon. Two participants wished they could see the exact changes in CSV file in the visual interface itself.

7 Discussion, Limitations, and Future Work

Efficacy. The efficacy of our tool depends on how accurately the causal model captures the underlying data generating process and the ensuing refining/debiasing process. As our tool is based on causal discovery algorithms, it inherits all its assumptions such as the causal markov condition and its limitations like sampling biases, etc. [37]. For eg., Caucasians had a higher mean age than African Americans in the COMPAS dataset. So, the PC algorithm falsely detected a causal edge between *Age* and *Race* (see appendix D.2). From our domain knowledge, we can deduce that this error is due to sampling bias. Ideally, one should use such an insight to gather additional data points for the under sampled group. However, it may not always be possible. In such cases, our tool can be leveraged to remove such patterns in the debiasing stage. We dealt with a similar case (*Gender* $\rightarrow$ *Marital status*) for the Adult Income data (see Sec. 5).

It is also worth noting that our tool is able to reduce disparity between the privileged and the unprivileged group but it may not be able to close the gap entirely. This can be due to missing proxy variables whose influence is unaccounted for or maybe because linear models are too simple to capture the relationship between a node and its parents. Future work might use non-linear SEMs and causal discovery algorithms like Fast Causal Inference (FCI) that can better deal with missing attributes.

Scalability. As the size of the dataset increases in terms of features and rows, scalability can become an issue. With dataset size, the time to generate a causal network, debiasing data, finding paths between nodes and computing evaluation metrics will all increase proportionally. Among all these steps, the process to generate the default causal network might be the most computationally expensive. So, future work should employ GPU-based parallel implementation of PC algorithm like cuPC [47] or use inherently faster causal discovery algorithms like F-GES [45] to better scale to larger datasets. On the front end, the causal network will become big and complex as the number of features increase. With limited screen space, the user might find it difficult to comprehend the causal network. To alleviate this issue, we have implemented different visual analytics techniques like zooming, panning, filtering weak edges, finding paths, etc. Future work might optimize graph layout algorithm and explore other visual analytics techniques like node aggregation to help navigate larger graphs better [45].

Applications. In this paper, we have emphasized how our tool can help identify and remove social biases. However, our approach and tool is not limited to social biases. Our tool can incorporate human feedback to realize policy and institutional goals as well (see appendix D.1). For eg., one might strengthen the edge between the nodes *Education* and *Income* to implant a policy intervention where people with higher education are incentivized. A ML model trained over the resulting dataset is likely to reflect such policy intervention in its predictions. Next, we plan to extend our HITL methodology to tackle biases in other domains such as word embeddings.

Human factors. Involving a human in the loop for identifying and debiasing data is a double edged sword. On one hand, it is a key strength of our tool as it provides real world domain knowledge and fosters accountability and trust. On the other hand, it can also be its main weakness if the human operating this tool intentionally/unconsciously injects social biases. A user can misuse the system in two ways. Firstly, the user can choose to ignore the social biases inherent in the dataset by not acting on the unfair causal edges. Such behaviour renders the system ineffective. Secondly, a biased user can explicitly introduce their own biases in the system by adding/strengthening unfair causal edges. Since this is a human aided tool, the biases that are inherent to the human user cannot be avoided. Hence, we recommend choosing the user responsibly. Moreover, we can always check the system logs and hold the person responsible for their action/inaction.

Acknowledgments

This work was partially funded by NSF grants CNS 1900706, IIS 1527200, IIS 1941613, and NSF SBIR contract 1926949.

REFERENCES

[1] Y. Ahn and Y.-R. Lin. Fairsight: Visual analytics for fairness in decision making. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1086–1095, 2019.

[2] S. Barocas, M. Hardt, and A. Narayanan. Fairness in machine learning. *Nips tutorial*, 1:2017, 2017.

[3] Y. Bechavod and K. Ligett. Learning fair classifiers: A regularization-inspired approach. *arXiv:1707.00044*, 2017.

[4] Y. Bechavod and K. Ligett. Penalizing unfairness in binary classification. *arXiv preprint arXiv:1707.00044*, 2017.

[5] R. K. Bellamy, K. Dey, M. Hind, et al. Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv*, 2018.

[6] P. J. Bickel, E. A. Hammel, and J. W. O'Connell. Sex bias in graduate admissions: Data from berkeley. *Science*, 1975.

[7] J. Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[8] Á. A. Cabrera, W. Epperson, F. Hohman, M. Kahng, J. Morgenstern, and D. H. Chau. Fairvis: Visual analytics for discovering intersectional bias in machine learning. *arXiv*, 2019.

[9] C. J. Cai, J. Jongejan, and J. Holbrook. The effects of example-based explanations in a machine learning interface. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, 2019.

[10] F. P. Calmon, D. Wei, K. N. Ramamurthy, and K. R. Varshney. Optimized data pre-processing for discrimination prevention. *arXiv preprint arXiv:1704.03354*, 2017.

[11] S. Chiappa. Path-specific counterfactual fairness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[12] S. Chiappa and W. S. Isaac. A causal bayesian networks viewpoint on fairness. In *IFIP International Summer School on Privacy and Identity Management*, pp. 3–20. Springer, 2018.

[13] D. Colombo and M. Maathuis. Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, 2014.

[14] B. J. Dietvorst, J. P. Simmons, and C. Massey. Overcoming algorithm aversion: People will use imperfect algorithms if they can (even slightly) modify them. *Management Science*, 2016.

[15] J. Drozdal, J. Weisz, D. Wang, G. Dass, B. Yao, C. Zhao, M. Muller, L. Ju, and H. Su. Trust in automl: exploring information needs for establishing trust in automated machine learning systems. In *International Conference on Intelligent User Interfaces*, 2020.

[16] C. Dwork and C. Ilvento. Group fairness under composition. In *Conference on Fairness, Accountability, and Transparency (FAT* 2018)*.

[17] M. Friendly. A fourfold display for 2 by 2 by k tables. Technical report, Citeseer, 1994.

[18] B. Ghai, M. N. Hoque, and K. Mueller. Wordbias: An interactive visual tool for discovering intersectional biases encoded in word embeddings. In *ACM CHI, Extended Abstracts*, 2021.

[19] B. Ghai, Q. V. Liao, Y. Zhang, R. Bellamy, and K. Mueller. Explainable active learning (xal) toward ai explanations as interfaces for machine teachers. *Proc. ACM CSCW*, 2021.

[20] B. Ghai, M. Mishra, and K. Mueller. Cascaded debiasing: Studying the cumulative effect of multiple fairness-enhancing interventions. *arXiv preprint arXiv:2202.03734*, 2022.

[21] J. C. Gower. A general coefficient of similarity and some of its properties. *Biometrics*, pp. 857–871, 1971.

[22] S. Hajian and J. Domingo-Ferrer. A methodology for direct and indirect discrimination prevention in data mining. *IEEE transactions on knowledge and data engineering*, 2013.

[23] S. G. Hart and L. E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in Psychology*, vol. 52, pp. 139–183. 1988.

[24] M. N. Hoque and K. Mueller. Outcome-explorer: A causality guided interactive visual interface for interpretable algorithmic decision making. *arXiv preprint arXiv:2101.00633*, 2021.

[25] F. Kamiran and T. Calders. Classifying without discriminating. In *Computer, Control and Communication, 2009. IC4 2009. 2nd International Conference on*, pp. 1–6. IEEE, 2009.

[26] F. Kamiran and T. Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33, 2012.

[27] J. Kleinberg, S. Mullainathan, and M. Raghavan. Inherent trade-offs in the fair determination of risk scores. *arXiv*, 2016.

[28] M. J. Kusner, J. Loftus, C. Russell, and R. Silva. Counterfactual fairness. In *Advances in Neural Information Processing Systems*.

[29] P. K. Lohia, K. N. Ramamurthy, M. Bhide, D. Saha, K. R. Varshney, and R. Puri. Bias mitigation post-processing for individual and group fairness. *arXiv preprint arXiv:1812.06135*, 2018.

[30] M. Madsen and S. Gregor. Measuring human-computer trust. In *11th australasian conference on information systems*, 2000.

[31] D. H. McKnight, V. Choudhury, and C. Kacmar. Developing and validating trust measures for e-commerce: An integrative typology. *Information Systems Research*, 2002.

[32] D. H. McKnight, L. L. Cummings, and N. L. Chervany. Initial trust formation in new organizational relationships. *Academy of Management Review*, 23(3):473–490, 1998.

[33] A. Narayanan. 21 fairness definitions and their politics. Conference on Fairness, Accountability, and Transparency, NYC, 2018.

[34] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

[35] J. Pearl. Causal inference in statistics: An overview. *Statistics Surveys*, 3(0):96–146, 2009. doi: 10.1214/09-SS057

[36] A. Rajabi and O. O. Garibay. Tabfairgan: Fair tabular data generation with generative adversarial networks, 2021.

[37] X. Shen, S. Ma, P. Vemuri, and G. Simon. Challenges and opportunities with causal discovery algorithms: Application to alzheimer's pathophysiology. *Scientific Reports*, 2020.

[38] O. Sofrygin, M. J. van der Laan, and R. Neugebauer. Simcausal r package: conducting transparent and reproducible simulation studies of causal effect estimation with complex longitudinal data. *J. of Statistical Software*, 2017.

[39] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT Press, Cambridge, MA, 2000.

[40] S. Wachter, B. Mittelstadt, and C. Russell. Why fairness cannot be automated: Bridging the gap between eu non-discrimination law and ai. *Computer Law & Security Review*, 41:105567, 2021.

[41] J. Wang and K. Mueller. Visual causality analysis made practical. In *Proc. IEEE VAST*, pp. 151–161, 2017.

[42] Q. Wang, Z. Xu, Z. Chen, Y. Wang, S. Liu, and H. Qu. Visual analysis of discrimination in machine learning. *arXiv*, 2020.

[43] Y. Wu, L. Zhang, and X. Wu. On discrimination discovery and removal in ranked data using causal graph. In *ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2018.

[44] T. Xie, Y. Ma, J. Kang, H. Tong, and R. Maciejewski. Fairrankvis: A visual analytics framework for exploring algorithmic fairness in graph mining models. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):368–377, 2021.

[45] X. Xie, F. Du, and Y. Wu. A visual analytics approach for exploratory causal analysis: Exploration, validation, and applications. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1448–1458, 2020.

[46] J. N. Yan, Z. Gu, H. Lin, and J. M. Rzeszotarski. Silva: Interactively assessing machine learning fairness using causality. In *ACM CHI Conference on Human Factors in Computing Systems*, 2020.

[47] B. Zarebavani, F. Jafarinejad, M. Hashemi, and S. Salehkaleybar. cupc: Cuda-based parallel pc algorithm for causal structure learning on gpu. *IEEE Trans. on Parallel and Distributed Systems*, 31(3):530–542, 2019.

[48] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork. Learning fair representations. In *Intern. Conf. on Machine Learning*, pp. 325–333, 2013.

[49] L. Zhang and X. Wu. Anti-discrimination learning: a causal modeling-based framework. *Intern. J. Data Science and Analytics*, 4(1):1–16, 2017.

[50] L. Zhang, Y. Wu, and X. Wu. A causal framework for discovering and removing direct and indirect discrimination. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017.