A Additional experimental results

The main body of the paper discusses results on all five ACS datasets. However, due to space constraints, plots are only shown for two example datasets: ACSIncome and ACSPublicCoverage. Appendices A.1–A.3 show analogous versions of each previous plot for the remaining three ACS datasets: ACSTravelTime, ACSMobility, and ACSEmployment. Plots are shown in the same order as in the main paper body. Additionally, we run a similar experiment on ACS datasets using only the two largest sensitive groups (*White* and *Black*), shown in Appendix A.4. Appendix A.5 presents results on the MEPS dataset (Blewett et al., 2021), an entirely different data source corresponding to real-world surveys of healthcare usage across the United States. Appendix A.6 provides further evidence that model ranking is maintained throughout all levels of constraint violation, and Appendix A.7 compares the results of unconstrained model training to unprocessing constrained models.

We consciously refrain from evaluating on the popular COMPAS dataset (Angwin et al., 2016), as related work has surfaced severe data gathering issues, including measurement biases and label leakage (Bao et al., 2021; Barenstein, 2019; Fabris et al., 2022). The German Credit dataset (Dua & Graff, 2017) — another popular benchmark in the fairness literature — suffers from its small size (1 000 samples), the age of its data (dates back to 1973–1975), and encoding issues that make it impossible to retrieve accurate sensitive information such as the individual's sex (Grömping, 2019). Overall, a total of 11 different evaluation scenarios were studied, pertaining to 6 datasets, with sizes ranging from 49K to 2.3M samples. Confidence intervals and metric results are computed using bootstrapping on the respective evaluation dataset (Efron & Tibshirani, 1994). We hope the scale of our study suffices to convince the reader of the validity of our claims. Source code is made available to easily reproduce our setup on other datasets.[2] All appendix experiments are in accordance with the main findings presented in Section 3.

A.1 Comparison between fairness methods

Figure A1 shows Pareto frontiers for all studied GBM-based algorithms. We observe a similar trend to that seen in Figure 4: preprocessing fairness methods can increase fairness but at dramatic accuracy costs, while EG and FairGBM inprocessing fairness methods trade Pareto-dominance between each other. Postprocessing Pareto frontier is also shown for reference, but a more detailed comparison between postprocessing and all other contender models is shown in the following section.

A.2 Postprocessing vs other methods

Figures A2–A6 show complete views of the Pareto frontiers obtained by postprocessing the model with highest validation accuracy m^* on each dataset (potentially obtained by unprocessing a fairness-aware model), together with a scatter of all other competing preprocessing, inprocessing, or unconstrained models (1 000 in total per dataset). Figure A7 shows detailed postprocessing results on each dataset, zoomed on the region of interest (maximal accuracy and minimal constraint violation, i.e., bottom right portion of the plot). Figures A8–A12 show results using only a subset of models: only GBM-based models. The main paper hypothesis is confirmed on each and every plot: we can obtain optimally fair classifiers at any level of constraint violation by postprocessing the model with highest accuracy, m^*, irrespective of its constraint violation.

[2] Supplementary materials:
`https://github.com/socialfoundations/error-parity/tree/supp-materials`

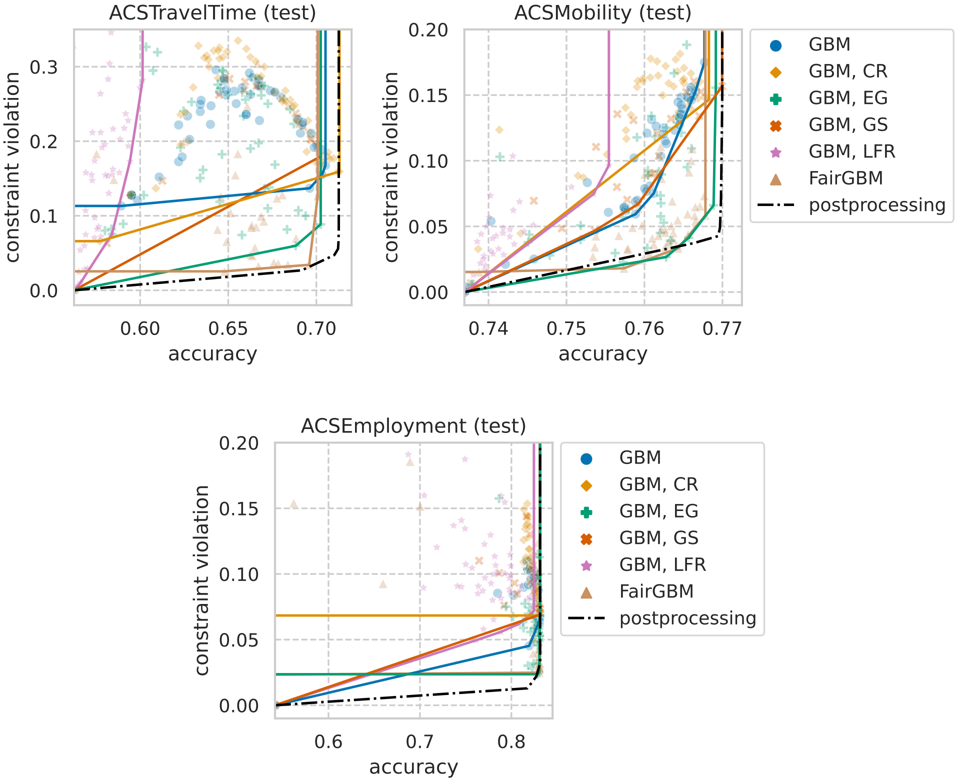

Figure A1: Pareto frontier attainable by each GBM-based ML algorithm, together with the Pareto frontier attained by postprocessing m^*, the GBM-based model with highest unprocessed validation accuracy. Plotted Pareto curves are linearly interpolated between Pareto-efficient models.

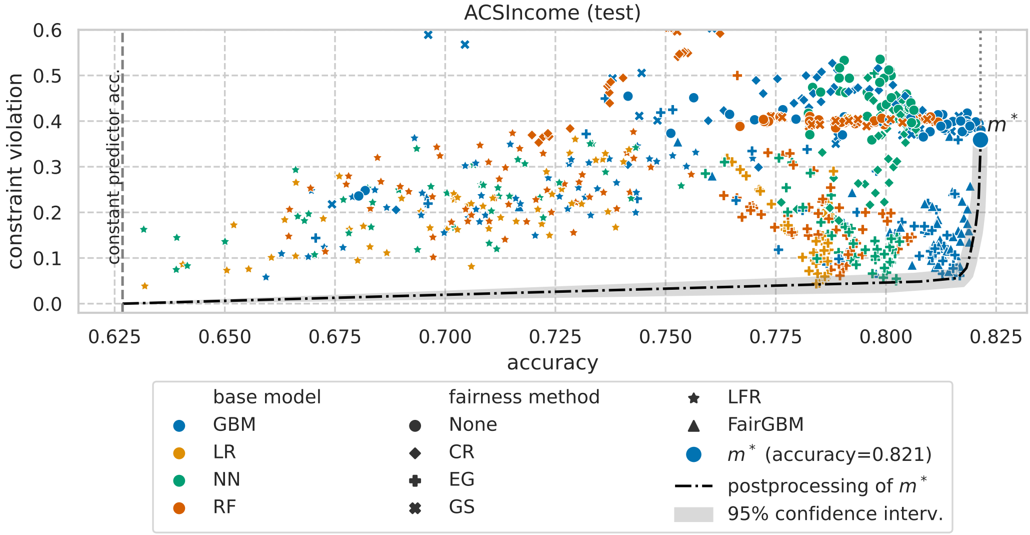

Figure A2: Fairness and accuracy test results for all 1 000 trained ML models (50 of each type) on the ACSIncome dataset. Colors portray different underlying unconstrained models and markers portray different fairness methods (or no fairness method for circle markers). The unconstrained model with highest validation accuracy, m^*, is shown with a larger marker, and the Pareto frontier attainable by postprocessing m^* is shown as a black dash-dot line, together with its 95% confidence intervals in shade. This is a colored and more granular version of Figure 1.

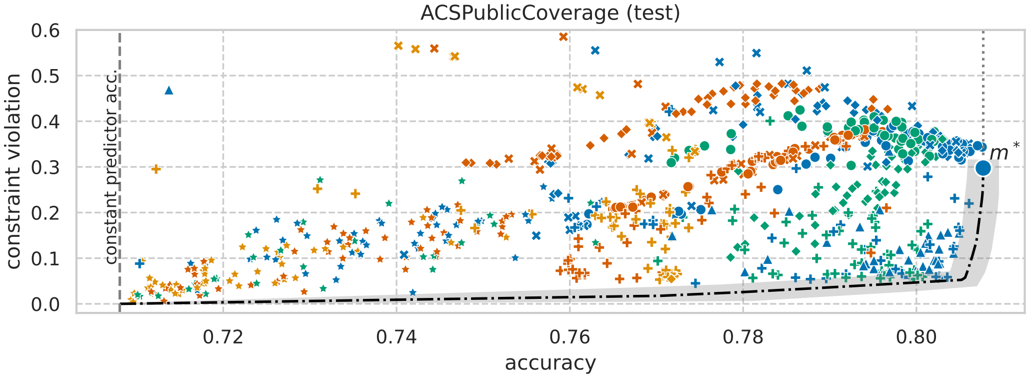

Figure A3: Fairness and accuracy test results on the ACSPublicCoverage dataset. Model m^* is of type ⟨GBM⟩ and achieves 0.808 accuracy. See legend and caption of Figure A2 for more details.

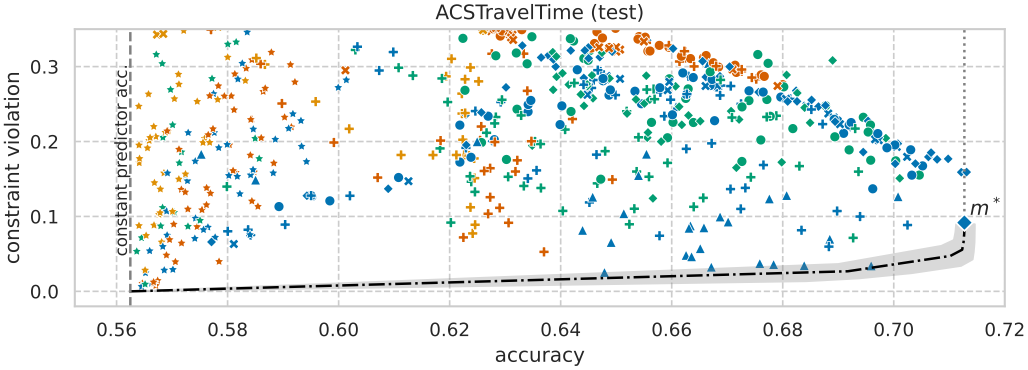

Figure A4: Fairness and accuracy test results on the ACSTravelTime dataset. Model m^* is of type ⟨GBM, CR⟩ and achieves 0.713 accuracy. See legend and caption of Figure A2 for more details.

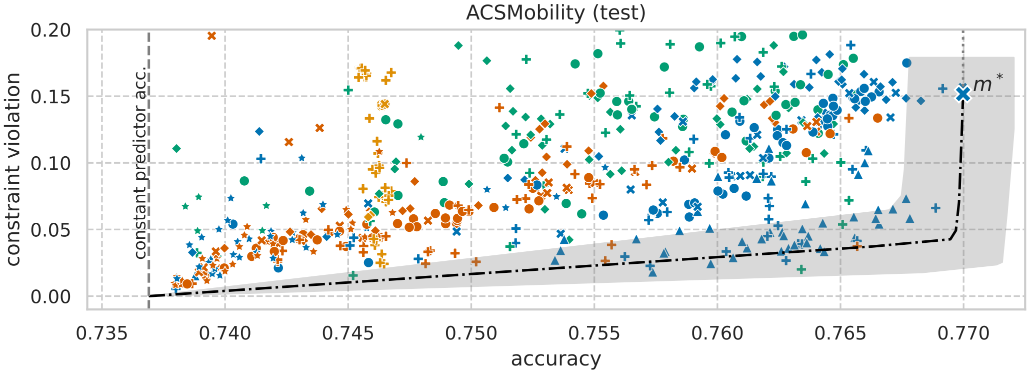

Figure A5: Fairness and accuracy test results on the ACSMobility dataset. Model m^* is of type ⟨GBM, GS⟩ and achieves 0.770 accuracy. See legend and caption of Figure A2 for more details.

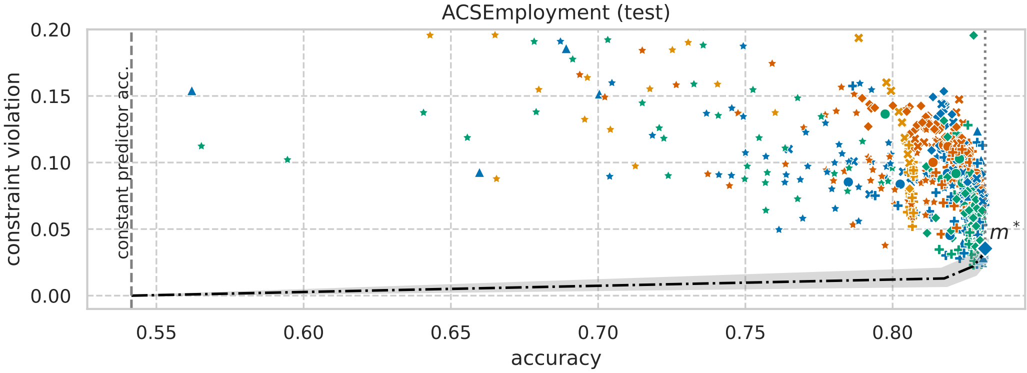

Figure A6: Fairness and accuracy test results on the ACSEmployment dataset. Model m^* is of type $\langle$GBM, CR$\rangle$ and achieves 0.831 accuracy. See legend and caption of Figure A2 for more details.

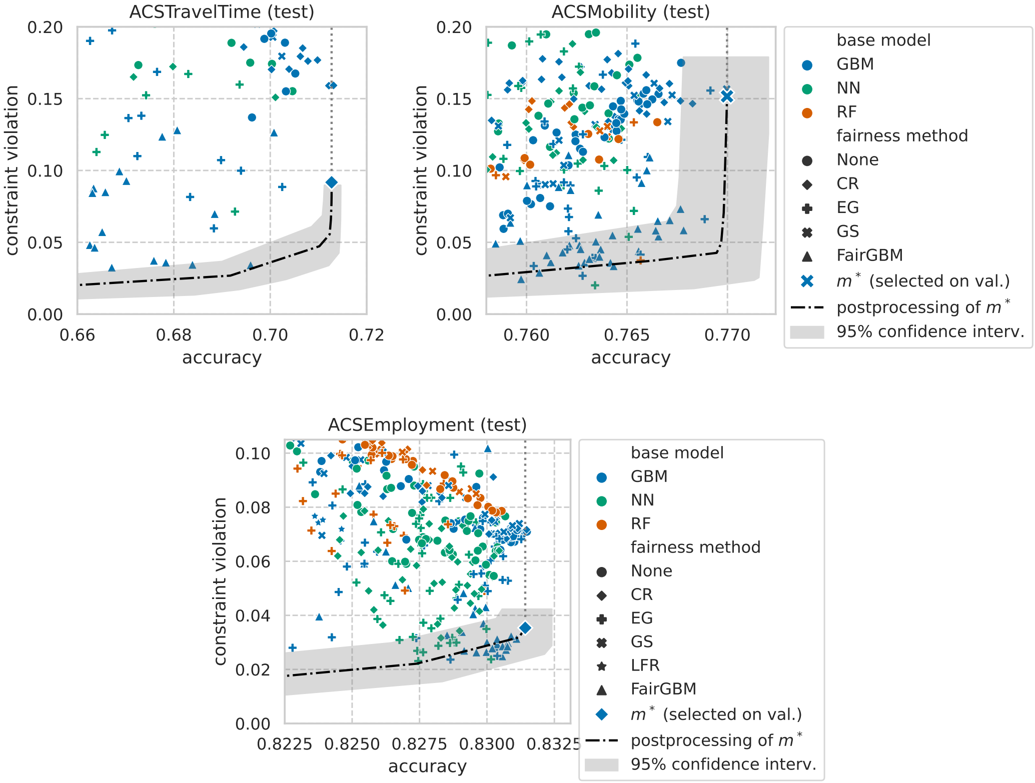

Figure A7: Detailed view of the postprocessing Pareto frontier on the ACSTravelTime (left), ACSMobility (right), and ACSEmployment (bottom) datasets. Respectively corresponds to zoomed-in versions of Figures A4 (left), A5 (right), and A6 (bottom).

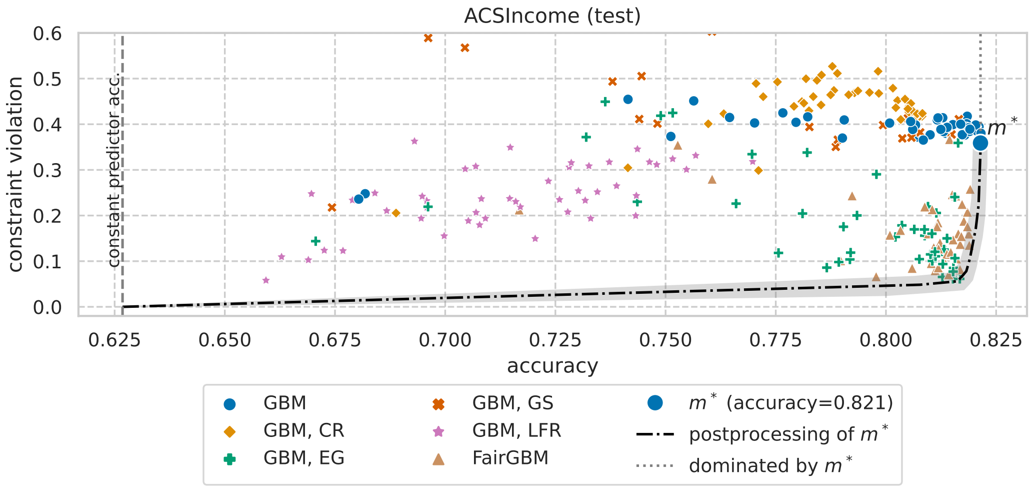

Figure A8: Fairness and accuracy test results for 300 GBM-based ML models (50 of each algorithm type) on the ACSIncome dataset. The unconstrained model with highest validation accuracy, m^*, is shown with a larger marker, and the Pareto frontier attainable by postprocessing m^* is shown as a black dash-dot line, together with its 95% confidence intervals in shade.

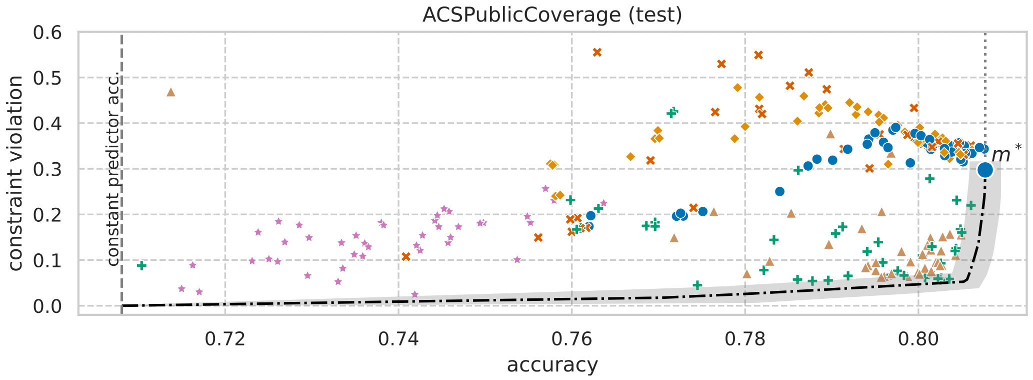

Figure A9: Fairness and accuracy test results for GBM-based ML models on the ACSPublicCoverage dataset. Model m^* is of type $\langle$GBM$\rangle$ and achieves 0.808 accuracy. See legend and caption of Figure A8 for more details.

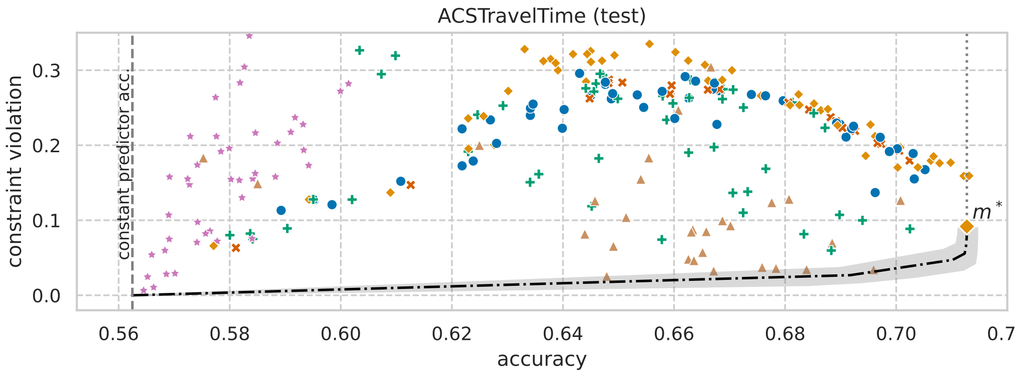

Figure A10: Fairness and accuracy test results for GBM-based ML models on the ACSTravelTime dataset. Model m^* is of type $\langle$GBM,CR$\rangle$ and achieves 0.713 accuracy. See legend and caption of Figure A8 for more details.

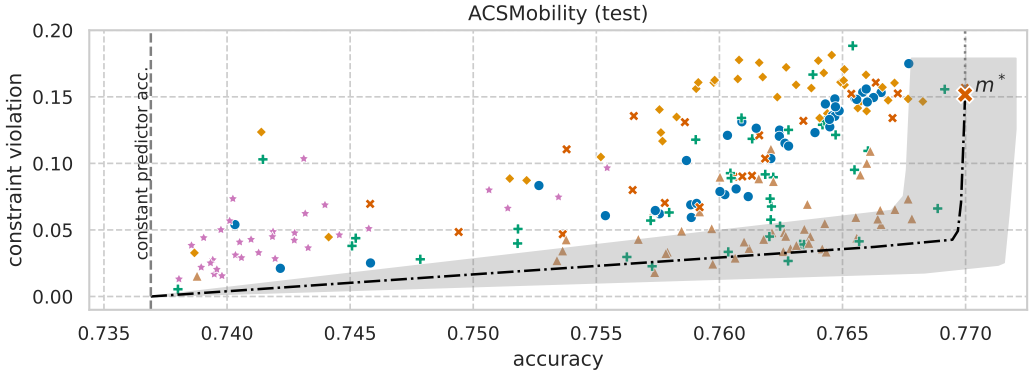

Figure A11: Fairness and accuracy test results for GBM-based ML models on the ACSMobility dataset. Model m^* is of type $\langle$GBM,GS$\rangle$ and achieves 0.770 accuracy.

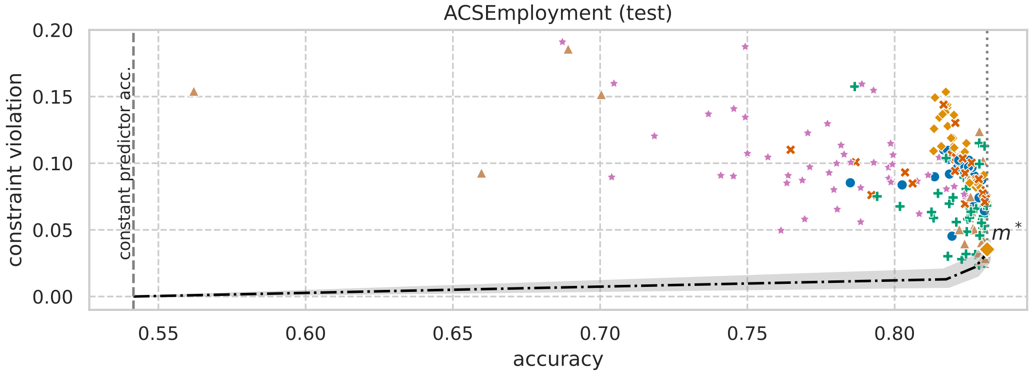

Figure A12: Fairness and accuracy test results for GBM-based ML models on the ACSEmployment dataset. Model m^* is of type $\langle$GBM,GS$\rangle$ and achieves 0.831 accuracy.

A.3 TIME TO FIT EACH METHOD

Figure A13 shows the mean time to fit each GBM-based model on three separate datasets. The trend is clear on all studied datasets: postprocessing is a small increment to the time taken to fit the base model, preprocessing methods take longer but are still within the same order of magnitude, the FairGBM inprocessing method also incurs a relatively small increment to the base model time, while EG and GS take one to two orders of magnitude longer to fit.

For clarification, all times listed are end-to-end process times for fitting and evaluating a given model. For example, postprocessing times include the time taken to fit the base GBM model plus the time taken to solve the LP. We note that most time consumed for postprocessing simply corresponds to computing the model scores for the respective dataset where postprocessing will be fitted, while solving the LP usually takes only a few seconds. Likewise, preprocessing fairness methods include the time taken to fit the preprocessing method, the time taken to transform the input data, and the time to fit the base model. Finally, inprocessing fairness methods include only the time taken to fit the inprocessing method, as no preprocessing or postprocessing steps are required. Nonetheless, the GS and EG inprocessing methods take significantly longer than any other competing method.

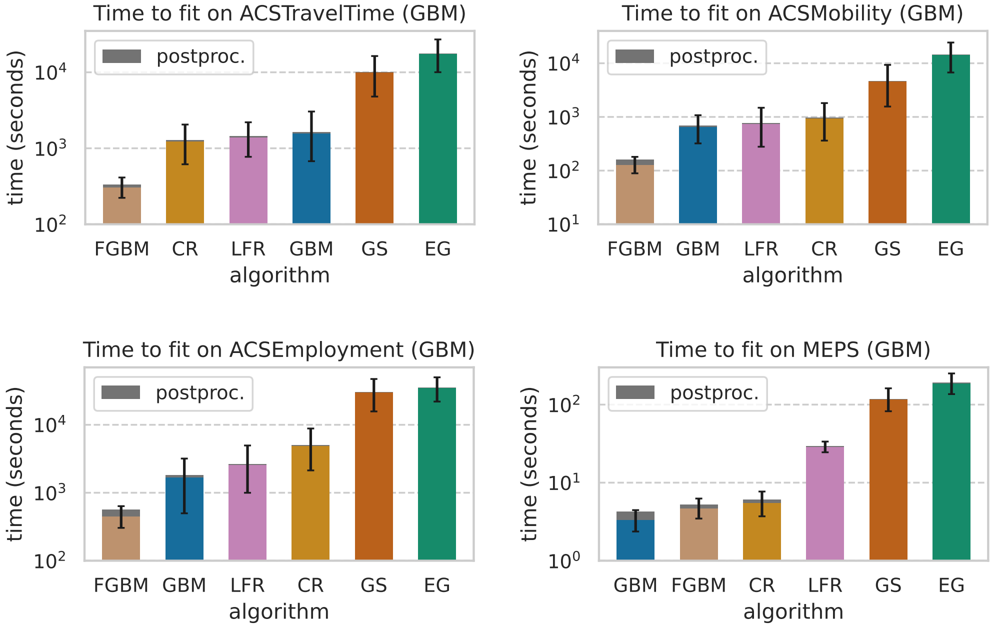

Figure A13: Mean time to fit the base GBM model and each studied fairness method on ACSTravelTime (top left), ACSMobility (top right), ACSEmployment (bottom left), and MEPS (bottom right), with 95% confidence intervals.

A.4 EXPERIMENTS WITH BINARY SENSITIVE GROUPS

While compatibility with more than two sensitive groups is arguably essential for real-world applicability of a fairness intervention, it is common among the fair ML literature to propose and evaluate methods considering only two groups (Zemel et al., 2013; Agarwal et al., 2018; Cruz et al., 2023).

In this binary-group setting, constrained optimization methods only have to consider two constraints:

$$\left|\mathbb{P}\left[\hat{Y}=1|S=0,Y=0\right]-\mathbb{P}\left[\hat{Y}=1|S=1,Y=0\right]\right|\leq\epsilon, \qquad \triangleright \text{ FPR constraint}$$

$$\left|\mathbb{P}\left[\hat{Y}=1|S=0,Y=1\right]-\mathbb{P}\left[\hat{Y}=1|S=1,Y=1\right]\right|\leq\epsilon, \qquad \triangleright \text{ TPR constraint}$$

respectively, a constraint on group-specific FPR, and another on group-specific TPR, with some small ϵ slack. By relaxing the equalized odds problem to only two constraints we expect to provide fairness-constrained methods with the best chance at disproving the paper hypothesis.

Figure A14 (as Figure 7) shows results of applying the experimental procedure detailed in Section 2.2 to a sub-sample of the ACS datasets: only samples from the two largest sensitive groups are used (*White* and *Black*). We observe substantially lower constraint violation across the board, both for unconstrained and fairness-aware models. In fact, even unconstrained unprocessed models (m^* on each plot) achieve below 0.1 constraint violation on 4 datasets when using binary groups (all but ACSIncome), and below 0.01 on 2 datasets (ACSMobility and ACSEmployment, see Figure A14). These results arguably discourage the use of binary sensitive groups on the ACSMobility and ACSEmployment datasets for fairness benchmarking, as very low disparities are effortlessly achieved.

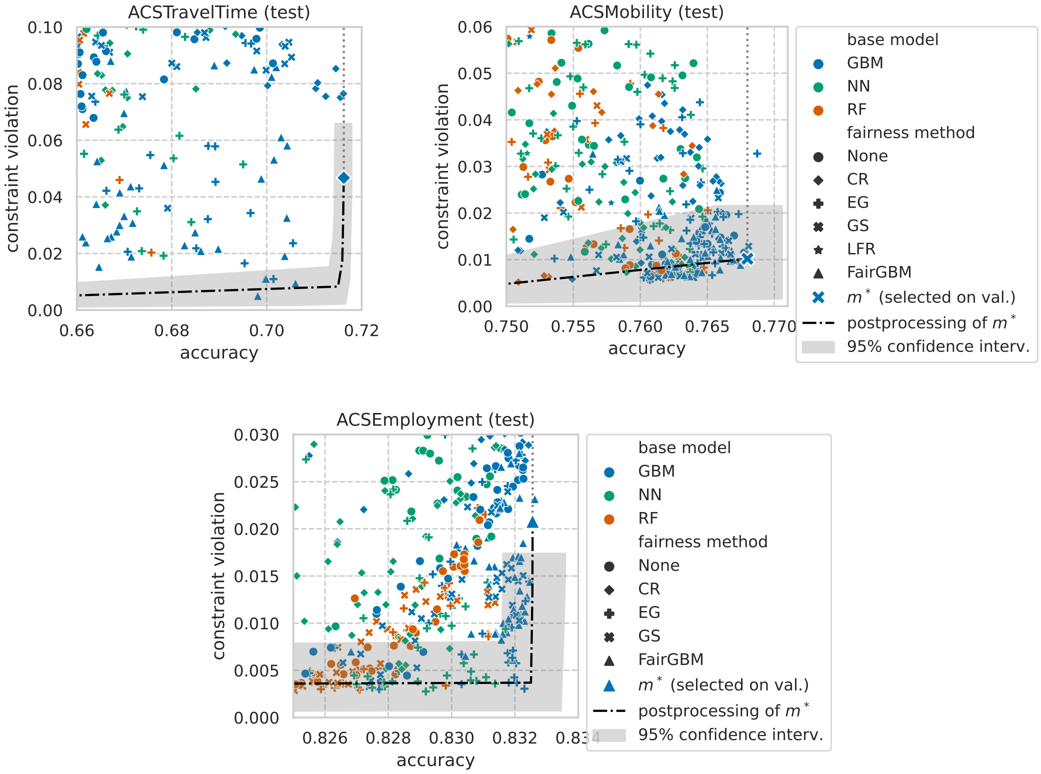

Figure A14: [**Binary protected groups**] Detailed view of the postprocessing Pareto frontier on the ACSTravelTime (left), ACSMobility (right), and ACSEmployment (bottom) datasets, when using only samples of the two largest groups (*White* and *Black*). Note the significantly reduced y axis range (constraint violation) when compared with results using four sensitive groups.

A.5 Results on the MEPS dataset

The Medical Expenditure Panel Survey (MEPS) (Blewett et al., 2021) dataset consists of large-scale surveys of families and individuals across the United States, together with their medical providers and employees. MEPS collects data on the health services used, costs and frequency of services, as well as demographic information of the respondents. The goal is to predict *low* (< 10) or *high* (≥ 10) medical services utilization. Utilization is defined as the yearly sum total of office-based visits, hospital outpatient visits, hospital emergency room visits, hospital inpatient stays, or home health care visits. Exact data pre-processing is made available in the supplementary materials.[2] We use survey panels 19 and 20 for training and validation (data is shuffled and split 70%/30%) — collected in 2015 and beginning of 2016 — and survey panel 21 for testing — collected in 2016. In total, the MEPS dataset consists of 49075 samples, 23380 of which are used for training, 10020 for validation, and 15675 for testing, making it over one order of magnitude smaller than the smallest ACS dataset in our study. We use race as the sensitive attribute, with 3 non-overlapping groups as determined by the panel data: *Hispanic*, *Non-Hispanic White*, and *Non-White*.

Figure A15 shows results of conducting the experiment detailed in Section 2.2 on the MEPS dataset. We note that the variance of results is the largest among all studied datasets, as evidenced by the wide confidence intervals. This is most likely due to the small dataset size. It is also possible that the m^* model on smaller datasets (such as MEPS) could produce scores that are farther from Bayes optimality than those of m^* on larger datasets (such as ACS). We hope that our study motivates additional empirical work on when exactly the optimality of postprocessing breaks in practice. We recall that, although no counter-example was observed among 11 000 trained models, there are known edge-cases where postprocessing is sub-optimal (Woodworth et al., 2017). Overall, empirical results on the MEPS dataset are in accordance with those observed on the ACS datasets: the most accurate unconstrained model can be postprocessed to match or dominate any other fairness-aware model.

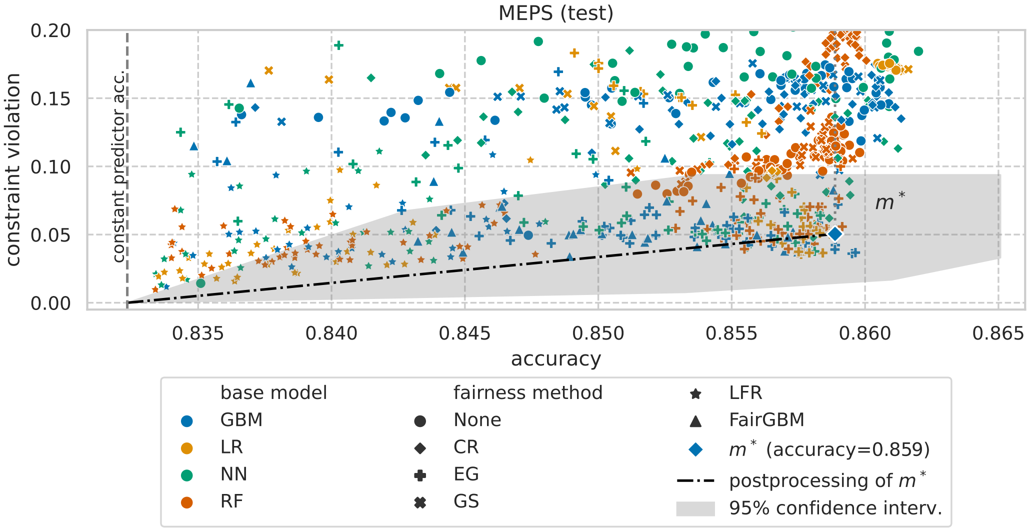

Figure A15: Detailed view of the postprocessing Pareto frontier of m^* on the MEPS dataset. Note the substantial variance in results, as shown by the wide postprocessing confidence intervals.

A.6 Ranking preservation between unprocessed and postprocessed versions

Figure A16 — akin to Figure 3 — shows real-data examples of the unprocessing-postprocessing experimental setup described in Section 2. The three plot panels show: (left) original results, (middle) results after unprocessing all models, and (right) original results with postprocessing curves overlaid.

We recall that the main experimental results (in Section 3) show that postprocessing the model with highest accuracy Pareto-dominates all other models (both fairness-aware and standard models). In this section, we present another perspective on the same empirical insight: given two specific incomparable models (A and B), the postprocessing curve of the model with highest unprocessed accuracy will Pareto-dominate the postprocessing curve of the model with lower unprocessed accuracy. That is, while Figure 6 compares postprocessing to all other fairness interventions, Figure A16 compares postprocessing to postprocessing. In this scenario, the same empirical insight is confirmed: taking the model with highest accuracy is superior at all levels of fairness constraint violation.

In summary, when near Bayes optimality,[3] model rankings are maintained across all postprocessing relaxations, i.e., if $A^* \succeq B^*$, then $\pi_r(A) \succeq \pi_r(B), \forall r \in [0, 1]$. We know this to be true on both extremes ($r = 0 \vee r = 1$) for a Bayes optimal model (Hardt et al., 2016): it achieves optimal accuracy, and its postprocessing achieves optimal fairness-constrained accuracy. At the same time, we know this to be false on some carefully constructed counter-examples (Woodworth et al., 2017). The focus of the present work is to study whether this ranking is generally maintained in practice, on real-world data. This hypothesis is confirmed on all experiments conducted throughout the paper.

[3] We only compare models that are Pareto-dominant among their algorithm cohort.

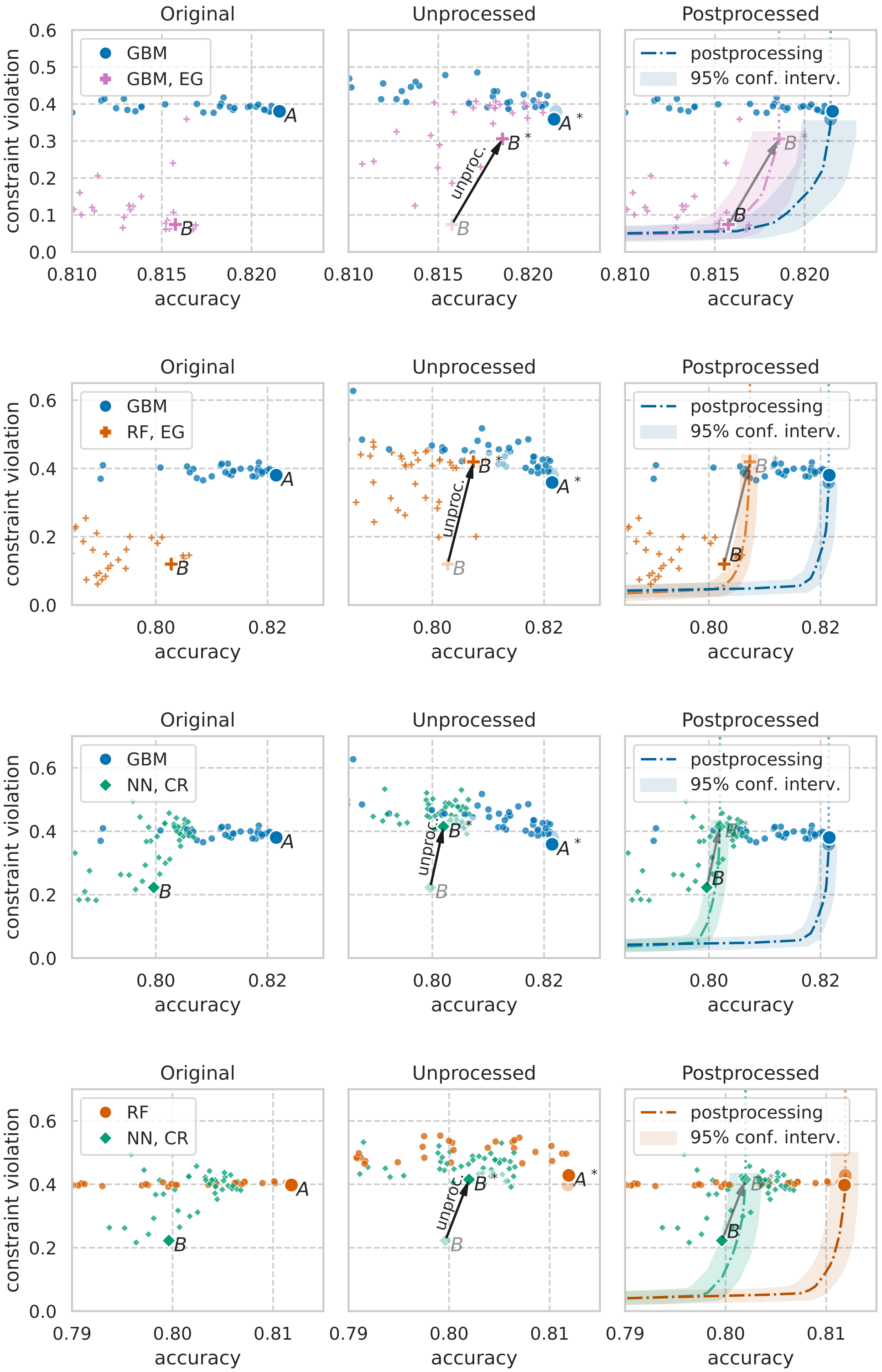

Figure A16: Comparison between postprocessing results between a variety of model pairs. Each model is selected as maximizing accuracy (model A) or maximizing a weighted average between accuracy and fairness (model B) among all models of the same algorithm cohort. Selection is performed on validation data, and results are shown on withheld test data; hence why some models may not be exactly at the Pareto frontier of their cohort. Results shown for the ACSIncome dataset.

One final noteworthy point is that unconstrained models are not significantly affected by unprocessing, occupying approximately the same fairness-accuracy region before and after optimization over group-specific thresholds (e.g., compare A with A^* in Figure A16). This is expected, as unconstrained learning optimizes for calibration by group (Liu et al., 2019), $P[Y = 1|R = r, S = s] = r, \forall s \in \mathcal{S}$, which leads to the same loss-minimizing threshold for all groups (further details in Appendix C).

A.7 Unprocessing vs unconstrained learning

As per Section 3, the best performing inprocessing fairness interventions are EG and FairGBM (i.e., highest Pareto-dominated area). In this section, we assess how unconstrained learning compares to unprocessing a model that was trained using either of these fairness interventions. Ideally, if enforcing the fairness constraint in-training did not hinder the learning process, we'd expect unprocessed models to approximately occupy the same fairness-accuracy region as unconstrained models.

Figure A17 shows results before and after unprocessing fairness-constrained models on the ACSIncome dataset. Unprocessing is done on validation data, and results are shown on withheld test data. The plots show that, after unprocessing, fairness-constrained models are naturally brought to similar levels of constraint violation as unconstrained models. While overlap between unconstrained and fairness-constrained models was previously minimal or non-existent (left plots), these models form clearly overlapping clusters after unprocessing (right plots). Figure A18 shows similar results before and after unprocessing fairness-constrained models, as well as results after postprocessing unconstrained models. As evident in the plots, unprocessing brings fairness-constrained models to the high-accuracy and high-disparity region that was previously occupied solely by unconstrained models; while postprocessing brings unconstrained models to the low-disparity region previously occupied solely by fairness-constrained models. This motivates the naming of *unprocessing*, as it can be seen as the inverse mapping of postprocessing. With these plots we aim to bring attention to the interchangeability of the underlying scores produced by both unconstrained and constrained models. Whether we want to deploy a fairness-constrained or an unconstrained classifier can be chosen after model training, by postprocessing a high-performing model to the appropriate value of fairness-constraint fulfillment. Finally, postprocessing has the added advantage of better-tuned fairness-constraint fulfillment, as models that were trained in a fairness-constrained manner suffer from a wide variability of constraint fulfillment (orange markers of left-most plots).

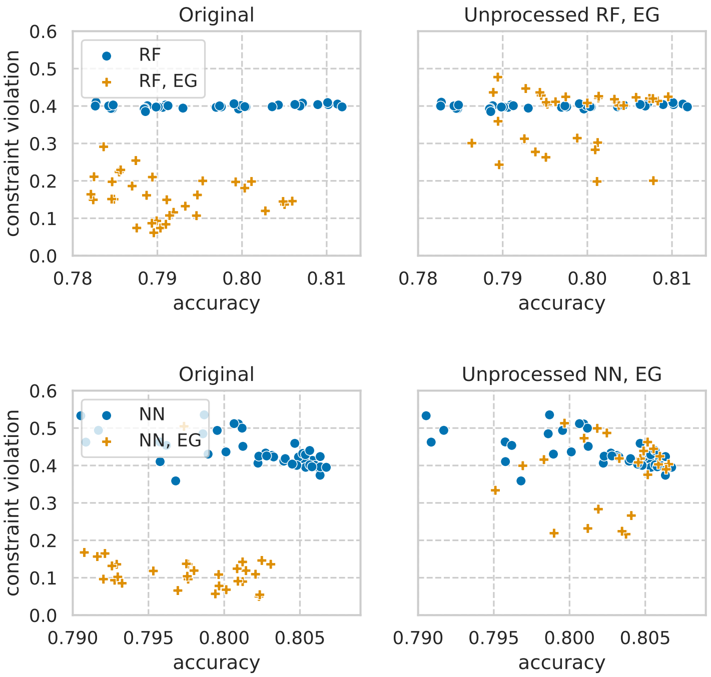

Figure A17: ACSIncome test results before (left) and after (right) unprocessing constrained models.

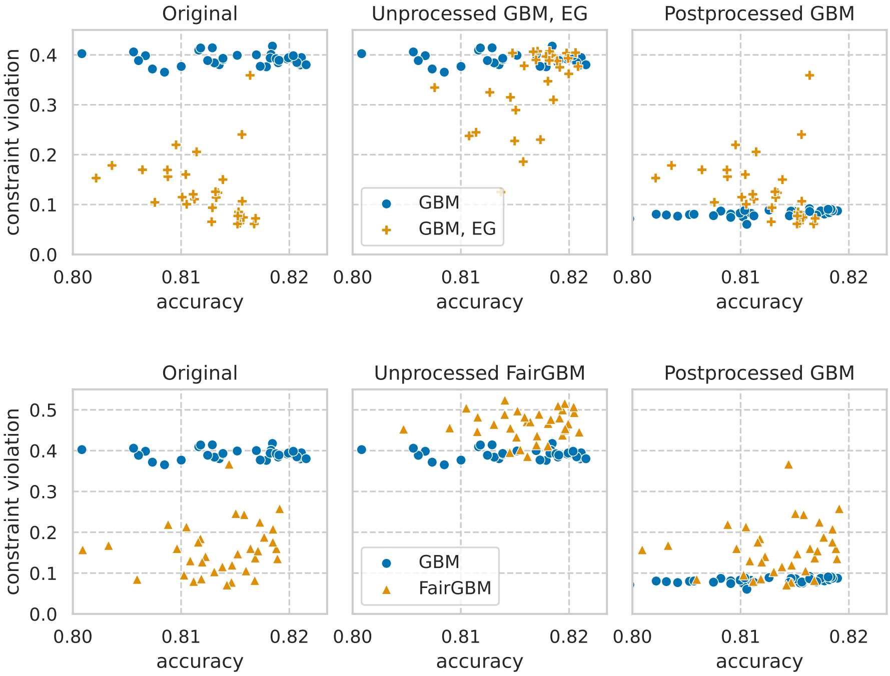

Figure A18: ACSIncome test results using GBM as the base model. *Left:* original results. *Middle:* after unprocessing fairness-constrained models. *Right:* after postprocessing unconstrained models.

B EXPERIMENT RUN DETAILS

All experiments were ran as jobs submitted to a centralized cluster, running the open-source `HTCondor` scheduler. Each job was given the same computing resources: 1 CPU. Compute nodes use `AMD EPYC 7662` 64-core CPUs. No GPUs were used. Memory was allocated as required for each algorithm: all jobs were allocated at least 16GB of RAM; GS and EG jobs were allocated 64GB of RAM as these ensembling algorithms have increased memory requirements.

An experiment job accounts for training and evaluating a single model on a given dataset. That is, 1 000 models were trained on each dataset (50 per algorithm type), totaling 11 000 models trained: 5 000 for the main ACS experiment using 4 sensitive groups, 5 000 for the ACS experiment using 2 sensitive groups, and 1 000 for the MEPS dataset experiment. Overall, the median job finished in 10.3 minutes, while the average job lasted for 112.0 minutes (most models are fast, but some fairness-aware models such as EG take a long time to fit, as seen in Figures 5 and A13). Compute usage was: 10 528 CPU hours for the main 4-group ACS experiment, 9 967 CPU hours for the binary group ACS experiment (Appendix A.4), and 31 CPU hours for the MEPS dataset experiment (Appendix A.5). Total compute usage was 20 526 CPU hours, which amounts to 14 days on a 64-core node. Detailed per-job CPU usage is available under folder `results` of the supplementary materials.[2]

Complete code base required to replicate experiments is provided as part of the supplementary materials, together with exact evaluation results for each trained model.[2]

C THRESHOLDING GROUP-CALIBRATED PREDICTORS

In this section we provide a proof for the following statement: for any classifier with group-calibrated scores (Equations 8–9), the group-specific decision thresholds that minimize the classification loss among each group all take the same value, $t_a = t_b, \forall a, b \in \mathcal{S}$, which is fully determined by the loss function, $t_s = \frac{\ell(1,0)}{\ell(1,0)+\ell(0,1)}, \forall s \in \mathcal{S}$.

Proof. Given a joint distribution over features, labels, and sensitive attributes (X, Y, S), a binary classification loss function $\ell : \{0,1\}^2 \to \mathbb{R}^+$, predictive scores $R = f(X)$, and binary predictions $\hat{Y} = \mathbb{1}\{R \geq t\}, t \in \mathcal{T} \subseteq \mathbb{R}$. Assume the scores R are *group-calibrated* (Barocas et al., 2019), i.e.:

$$\mathbb{P}[Y = 1 | R = r, S = s] = r, \qquad \forall r \in [0,1], \quad \forall s \in \mathcal{S}, \tag{8}$$

$$\mathbb{P}[Y = 0 | R = r, S = s] = 1 - r, \qquad \forall r \in [0,1], \quad \forall s \in \mathcal{S}. \tag{9}$$

We want to minimize the expected loss among samples of group s, $L_s(t) = \mathbb{E}\left[\ell(\hat{Y}, Y) | S = s\right]$:

$$L_s(t) = \ell(1,0) \cdot \mathbb{P}\left[\hat{Y} = 1, Y = 0 | S = s\right] + \ell(0,1) \cdot \mathbb{P}\left[\hat{Y} = 0, Y = 1 | S = s\right], \tag{10}$$

assuming w.l.o.g. no cost for correct predictions $\ell(0,0) = \ell(1,1) = 0$.

We have:

$$\mathbb{P}\left[\hat{Y} = 1, Y = 0 | S = s\right] = \mathbb{P}\left[\hat{Y} = 1 | Y = 0, S = s\right] \cdot \mathbb{P}[Y = 0 | S = s] = h_s^{\text{FP}}(t) \cdot \mathbb{P}[Y = 0 | S = s],$$

$$\mathbb{P}\left[\hat{Y} = 0, Y = 1 | S = s\right] = \mathbb{P}\left[\hat{Y} = 0 | Y = 1, S = s\right] \cdot \mathbb{P}[Y = 1 | S = s] = h_s^{\text{FN}}(t) \cdot \mathbb{P}[Y = 1 | S = s],$$

where $h_s^{\text{FP}}(t)$ and $h_s^{\text{FN}}(t)$ are, respectively, the False Positive Rate (FPR) and the False Negative Rate (FNR) among samples of group s, as functions of the chosen group-specific threshold t. We can trade-off FPR and FNR by varying the threshold, leading to a 2-dimensional curve known as the Receiver Operating Characteristic (ROC) curve.

Furthermore, given the conditional density function of R given $S = s$, $p_{R|s}(r)$, we have:

$$\begin{aligned}
h_s^{\text{FP}}(t) &= \mathbb{P}\left[\hat{Y} = 1 | Y = 0, S = s\right] \\
&= \mathbb{P}[R \geq t | Y = 0, S = s] \\
&= \frac{\mathbb{P}[Y = 0 | R \geq t, S = s] \cdot \mathbb{P}[R \geq t | S = s]}{\mathbb{P}[Y = 0 | S = s]} \\
&= \int_t^1 \frac{(1 - r) \cdot p_{R|s}(r)}{\mathbb{P}[Y = 0 | S = s]} \, dr, && \triangleright \text{using calibration (Eq. 9)} \\
\frac{\partial h_s^{\text{FP}}}{\partial t} &= \frac{(t - 1) \cdot p_{R|s}(t)}{\mathbb{P}[Y = 0 | S = s]},
\end{aligned}$$

and,

$$\begin{aligned}
h_s^{\text{FN}}(t) &= \mathbb{P}\left[\hat{Y} = 0 | Y = 1, S = s\right] \\
&= \mathbb{P}[R < t | Y = 1, S = s] \\
&= \frac{\mathbb{P}[Y = 1 | R < t, S = s] \cdot \mathbb{P}[R < t | S = s]}{\mathbb{P}[Y = 1 | S = s]} \\
&= \int_0^t \frac{r \cdot p_{R|s}(r)}{\mathbb{P}[Y = 1 | S = s]} \, dr, && \triangleright \text{using calibration (Eq. 8)} \\
\frac{\partial h_s^{\text{FN}}}{\partial t} &= \frac{t \cdot p_{R|s}(t)}{\mathbb{P}[Y = 1 | S = s]}.
\end{aligned}$$

The threshold t_s that minimizes the group-specific loss $L_s(t)$ is a solution to $\frac{\partial L_s}{\partial t} = 0$, where:

$$\begin{aligned}
L_s(t) &= \ell(1,0) \cdot h_s^{\text{FP}}(t) \cdot \mathbb{P}[Y = 0 | S = s] + \ell(0,1) \cdot h_s^{\text{FN}}(t) \cdot \mathbb{P}[Y = 1 | S = s], \\
\frac{\partial L_s}{\partial t} &= \ell(1,0) \cdot (t - 1) \cdot p_{R|s}(t) + \ell(0,1) \cdot t \cdot p_{R|s}(t).
\end{aligned}$$

Hence, for a group-calibrated predictor (fulfilling Equations 8–9), for any group $s \in \mathcal{S}$, the optimal group-specific decision threshold t_s does not depend on any group quantities, and is given by:

$$t_s = \frac{\ell(1,0)}{\ell(1,0) + \ell(0,1)}.$$

□