A. Proof of Theorem 1

A useful strategy is to solve (10) for a set of solutions, then ask if any of these solutions satisfies an additional fairness constraint $\phi^{(K)}(\mathbf{z}) = 0$. This proof, as well as many of the ones below, illustrate this strategy in practice.

Proof. First, set $K = 1$ and $\mathbf{A}^{(0)} = \mathbf{A}_{\text{CG}}$ in (10). Since $v_0 \neq v_1$, the matrix $\mathbf{A}$ is full rank and therefore admits the solution (11). Considering $\mathbf{z}_0 \geq 0$ yields immediately the condition (12).

Next, set $K > 1$. Then either $\mathbf{z}_0$ is a solution (which is the case when all other fairness notions are linear and linearly dependent on $\begin{pmatrix} \mathbf{A}_{\text{CG}} \\ \mathbf{A}_{\mathsf{const}} \end{pmatrix}$), or otherwise no solution exists to both (10) and $\phi^{(1)}(\mathbf{z}) = \cdots = \phi^{(K-1)}(\mathbf{z}) = 0$ simultaneously. □

This theorem states that $\Phi = \{\text{CG}\}$ is incompatible when $v_0 \neq v_1$, since it is a singleton set of incompatible fairness.

The condition $v_0 \neq v_1$ is necessary in Theorem 1, which is reasonable to assume as we would expect the positive class to have a higher score than the negative class in the definition of CG. We can prove the necessity of this condition by contradiction. In the degenerate case $v_0 = v_1 = v$, $\Phi = \{\text{CG}\}$ is a set of compatible fairness notions. It turns out that (10) with $K = 1$ is only on rank 6. Denoting ⓘ as the ith row of the matrix, we have two linear dependencies, ⑤+⑥+v①=②and ⑦ + ⑧ + v③ = ④. There is no longer a unique solution to the (10); instead, we have a two-parameter family of solutions,

$$\mathbf{z}(\alpha, \beta) = \frac{1}{N(1-v)} \begin{pmatrix} v(N_1(1-v) - \alpha) \\ v\alpha \\ (1-v)(N_1(1-v) - \alpha) \\ (1-v)\alpha \\ v(N_0(1-v) - \beta) \\ v\beta \\ (1-v)(N_0(1-v) - \beta) \\ (1-v)\beta \end{pmatrix}, \tag{14}$$
$$0 \leq \alpha \leq (1-v)N_1, \quad 0 \leq \beta \leq (1-v)N_0.$$

Furthermore, this family of solutions satisfies $\mathbf{A}_{\mathsf{const}}\mathbf{z}_0 = \mathbf{b}_{\mathsf{const}}$ if and only if $v = M_0/N_0 = M_1/N_1$, i.e. the base rates are equal and furthermore the score for both bins is equal to the base rate.

B. Proof of Corollary 1

Proof. Consider the product

$$\begin{pmatrix} \mathbf{A}_{\text{PCB}} \\ \mathbf{A}_{\text{NCB}} \end{pmatrix} z_0 = \frac{M_1N_0 - M_0N_1}{N} \begin{pmatrix} \frac{v_0v_1}{M_0M_1} \\ \frac{(1-v_0)(1-v_1)}{(M_0-N_0)(M_1-N_1)} \end{pmatrix}. \tag{15}$$

This product equals the zero vector (and hence satisfies both PCB and NCB) if and only if either of the conditions of the Corollary hold. (The last solution, $v_0 = 1$ and $v_1 = 0$, is inadmissible since $v_0 < v_1$ by assumption.) □

C. Proof of Corollary 2

Proof. The result follows from solving

$$\mathbf{A}_{\text{DP}}\mathbf{z}_0 = \frac{M_1N_0 - M_0N_1}{N^2(v_1 - v_0)} = 0. \tag{16}$$

□

D. Proof of Corollary 3

Proof. The result follows from solving

$$\phi_{\text{PP}}(\mathbf{z}_0) = v_1(1-v_1)\left((M_1 - N_1v_0)^2 - (M_0 - N_0v_0)^2\right) = 0 \tag{17}$$

which is true if and only if either condition in the Corollary is true. (The last case, $v_1 = 0$, is inadmissible by assumption.) □

In addition, here is a situation of fairness "for free", in the sense that one notion of fairness automatically implies another.

Corollary 4. *Consider a classifier that satisfies CG fairness. Then, the classifier also satisfies EFOR fairness. In other words, {CG, EFOR} is incompatible.*

Proof. $\phi_{\text{EFOR}}(\mathbf{z}_0) = 0$ vanishes identically. □

E. Proof of Theorem 2

Proof. Finding the solution to $\phi_{\text{PP}}(\mathbf{z}) = \phi_{\text{EFPR}}(\mathbf{z}) = \phi_{\text{EFNR}}(\mathbf{z}) = 0$ and also the linear system $\mathbf{A}_{\mathsf{const}}\mathbf{z} = \mathbf{b}_{\mathsf{const}}$ yields the three conditions of the Theorem. □

F. CG–accuracy trade-offs

In the paper, we have only considered the case when $\lambda = \infty$ in the LAFOP: we only consider when the fairness criteria are satisfied exactly yielding several fairness–fairness trade-off results without heed to the accuracy of the classifiers. Nonetheless, recall that LAFOP allows us to express both fairness–accuracy and fairness–fairness trade-offs by introducing an accuracy objective along with a fairness regularizer. In this section, we show how the LAFOP can be used to theoretically analyze a simple fairness–accuracy trade-off. We present a small result that is relevant to the CG–accuracy trade-off considered in (Liu et al., 2019).

Theorem 3. *Let $\alpha = (M_0 + M_1)/N$ be the base rate. Consider a classifier that satisfies CG with $0 \leq v_0 < v_1 \leq 1$. Then, perfect accuracy is attained if and only if*

$$\frac{v_0(1 - 2v_1)}{1 - v_1 + v_0} = \alpha \leq \frac{1}{8}, \quad \left| v_0 - \frac{1}{4} \right| \leq \frac{\sqrt{1 - 8\alpha}}{4}. \tag{18}$$

Proof. The case of necessity ($\Rightarrow$) follows immediately from solving $\mathbf{c} \cdot \mathbf{z}_0 = 0$, where $\mathbf{z}_0$ is defined in Theorem 1. The inequality conditions follow immediately from the constraint $0 \leq v_0 < v_1 \leq 1$. The case of sufficiency ($\Leftarrow$) follows immediately from Theorem 1 and substituting the equality condition. □

The condition of this theorem relates the scores v_0 and v_1 to the base rate of the data, thus providing simple, explicit data dependencies that are necessary and sufficient.

G. Experiment Details

G.1. Optimization

For solving the optimization problems, we used solvers in the `scipy` package for Python (Jones et al., 2001). For linear fairness constraints, we used the simplex algorithm (Dantzig, 1963), and for other constrained optimization forms, we used sequential least-squares programming (SLSQP) solver (Kraft, 1988; 1994).

G.2. Model-agnostic multi-way fairness–accuracy trade-offs

We have only considered situations where zero or one parameter is sufficient to simultaneously specify the fairness strength for every fairness function, i.e. $\lambda = \lambda_0 = \cdots = \lambda_{K-1}$. In this section, we generalize this and allow each regularization parameter to vary freely. It is then natural to consider the multilinear least-squares accuracy–fairness optimality problem (MLAFOP): $\arg\min_{\mathbf{z}\in\mathcal{K}} (\mathbf{c} \cdot \mathbf{z})^2 + \sum_{i=0}^{K-1} \lambda_i \|\mathbf{A}^{(i)}\mathbf{z}\|_2^2$, where the regularization parameters λ_i now take different values across each of the K fairness constraints. This allows for a general inspection of the individual effect of fairness constraints in a group.

For instance, a three-way trade-off among EOd, DP, and accuracy can be visualized as a contour plot, similar to the one shown in Figure 4. And for general $(K+1)$-way trade-offs involving K fairness constraints and accuracy, we visualize two-dimensional slices along the $K+1$-dimensional surface. For example, consider a four-way trade-off between a group of three fairness definitions (DP, EOd, PCB) and accuracy. Figure 2 already showed that imposing PCB given (DP, EOd)

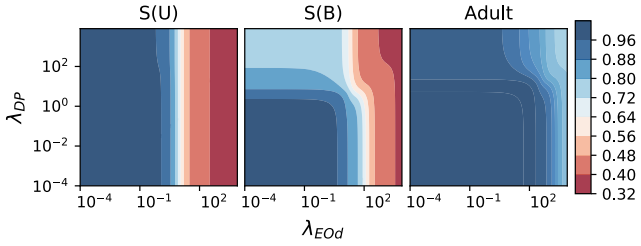

Figure 4. Fairness–fairness–accuracy trade-off analysis using contour plot of accuracy with varying regularization strengths of Demographic Parity (DP) and Equalized Odds (EOd) for the unbiased synthetic dataset (left), biased synthetic dataset (middle), and Adult dataset (right). The contours show how the regularization strength of each fairness individually influence the accuracy $(1-\delta)$ given the other (accuracy of 1.0 being the accuracy of the Bayes classifier). For the unbiased synthetic data, the accuracy change along the vertical axis (DP) is practically nonexistent given EOd, while along the horizontal axis (EOd) the change is drastic. Other datasets demonstrate more complex relationships.

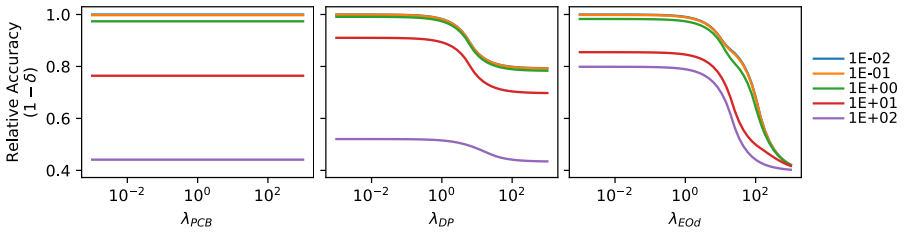

Figure 5. The four-way trade-off between accuracy, PCB, EOd, and DP in the biased synthetic dataset (Section 6.1). Shown here is the $(1-\delta)$ value as a function of some regularization strength λ_ϕ for some fairness function ϕ, while holding all other $\lambda_{\phi'}$s constant (accuracy of 1.0 being the accuracy of the Bayes classifier). The value next to each colored line in the legend represents constant values for the fixed $\lambda_{\phi'}$s. Sweeping through PCB while keeping DP and EOd fixed (left) does not change the accuracy, whereas the other plots show multiple levels of variations. For EOd (right), the accuracy levels converge quickly to the limiting value of 0.392 as shown in Figure 2, suggesting that the accuracy is more sensitive to changes in EOd constraint strength compared to the others.

does not affect δ, which implies that PCB is the weakest in terms of its influence on δ. To get more information, for the S(B) dataset, we show in Figure 5 three cases of varying one λ for one fairness constraint while keeping the other λ values fixed in MLAFOP. Sweeping through PCB condition (left) does not affect $1-\delta$ at fixed EOd and DP levels, confirming the observation from Figure 2. Sweeping through DP conditions while keeping PCB and EOd strengths fixed (middle) results in a slight drop, but not big enough to make all levels to converge to values reported in Figure 2 (0.392). Sweeping through EOd while keeping PCB and DP strengths fixed (right) on the other hand results in significant changes for all levels and convergence to the value 0.392, suggesting EOd is stronger than DP in terms of its influence on changing δ. This notion of relative influence of fairness deserves further investigation, to see if these preliminary results are robust across other slices and datasets. Nonetheless, such analysis demonstrates a clear picture of how different notions of fairness interact with one another when they are to be imposed together.

G.3. Connection to the post-processing methods for fair classification

We can explicitly rewrite the constraints in (8) using $\hat{z}$ and $\tilde{z}$, which respectively correspond to the fairness–confusion tensor of the given pre-trained classifier $\hat{Y}$ and the derived fair classifier $\tilde{Y}$:

$$\gamma_0(\tilde{Y}) = \gamma_1(\tilde{Y}) \iff \mathbf{A}_{\text{EOD}}\tilde{\mathbf{z}} = 0$$

$$\gamma_0(\tilde{Y}) \in P_0(\hat{Y}) \iff \left(\frac{\tilde{\mathbf{z}}_7}{\tilde{\mathbf{z}}_7 + \tilde{\mathbf{z}}_8}, \frac{\tilde{\mathbf{z}}_5}{\tilde{\mathbf{z}}_5 + \tilde{\mathbf{z}}_6}\right) \in \text{convhull}\left\{(0,0), \left(\frac{\hat{\mathbf{z}}_7}{\hat{\mathbf{z}}_7 + \hat{\mathbf{z}}_8}, \frac{\hat{\mathbf{z}}_5}{\hat{\mathbf{z}}_5 + \hat{\mathbf{z}}_6}\right), \left(\frac{\hat{\mathbf{z}}_8}{\hat{\mathbf{z}}_7 + \hat{\mathbf{z}}_8}, \frac{\hat{\mathbf{z}}_6}{\hat{\mathbf{z}}_5 + \hat{\mathbf{z}}_6}\right), (1,1)\right\} \tag{19}$$

$$\gamma_1(\tilde{Y}) \in P_1(\hat{Y}) \iff \left(\frac{\tilde{\mathbf{z}}_3}{\tilde{\mathbf{z}}_3 + \tilde{\mathbf{z}}_4}, \frac{\tilde{\mathbf{z}}_1}{\tilde{\mathbf{z}}_1 + \tilde{\mathbf{z}}_2}\right) \in \text{convhull}\left\{(0,0), \left(\frac{\hat{\mathbf{z}}_3}{\hat{\mathbf{z}}_3 + \hat{\mathbf{z}}_4}, \frac{\hat{\mathbf{z}}_1}{\hat{\mathbf{z}}_1 + \hat{\mathbf{z}}_2}\right), \left(\frac{\hat{\mathbf{z}}_4}{\hat{\mathbf{z}}_3 + \hat{\mathbf{z}}_4}, \frac{\hat{\mathbf{z}}_2}{\hat{\mathbf{z}}_1 + \hat{\mathbf{z}}_2}\right), (1,1)\right\} \tag{20}$$

where the subscript i of the fairness–confusion tensor corresponds to the i-th element in their vector representation as in Section 3. By setting the objective function to be the classification error, imposing EOd fairness constraint and the model-dependent feasibility constraints in (19) and (20), MS-LFAOP is the same optimization problem as the post-processing methods, now over the space of the fairness–confusion tensors. The FACT Pareto frontier obtained by solving MS-LAFOP therefore can assess the trade-off exhibited by any classifier post-processed in such ways.

In practice, the post-processing method solves (8) by parameterizing $\tilde{Y}$ with two variables for each group $a = 0, 1$: $\mathsf{Pr}(\tilde{Y} = 1|\hat{Y} = 1, A = a), \mathsf{Pr}(\tilde{Y} = 1|\hat{Y} = 0, A = a)$. (Hardt et al., 2016). These values are called the *mixing rates*, as they indicate the probability of labels that should be flipped or kept for each group when post-processing the given classifier $\hat{Y}$. The algorithm then randomly selects the instances for each group to flip according to these mixing rates. These mixing rates can also be written in terms of the fairness–confusion tensor $\tilde{z}$ and $\hat{z}$, by using the fact that

$$\begin{aligned}\mathsf{Pr}(\tilde{Y} = \tilde{y}|Y = y, A = a) = \mathsf{Pr}(\tilde{Y} = \tilde{y}|\hat{Y} = 1, A = a)\mathsf{Pr}(\hat{Y} = 1|Y = y, A = a) + \\ \mathsf{Pr}(\tilde{Y} = \tilde{y}|\hat{Y} = 0, A = a)\mathsf{Pr}(\hat{Y} = 0|Y = y, A = a),\end{aligned}$$

and that $\mathsf{Pr}(\tilde{Y} = \tilde{y}|Y = y, A = a)$, $\mathsf{Pr}(\hat{Y} = \hat{y}|Y = y, A = a)$ terms are essentially what $\tilde{z}$ and $\hat{z}$ encode. Therefore, by using $\tilde{z}$ obtained from the MS-LAFOP above, we can compute the mixing rates to post-process the given classifier.