OxonFair: A Flexible Toolkit for Algorithmic Fairness

Eoin Delaney [1] **Zihao Fu** [1] **Brent Mittelstadt** [1] **Sandra Wachter** [1] **Chris Russell** [1]

Abstract

We present OxonFair, a new open source toolkit for enforcing fairness in binary classification. Compared to existing toolkits: (i) We support NLP and Computer Vision classification as well as standard tabular problems. (ii) We support enforcing fairness on validation data, making us robust to a wide range of overfitting challenges. (iii) Our approach can optimize any measure based on True Positives, False Positive, False Negatives, and True Negatives. This makes it easily extendable and much more expressive than existing toolkits. It supports 9/9 and 10/10 of the decision-based group metrics of two popular review papers. (iv) We jointly optimize a performance objective. This not only minimizes degradation while enforcing fairness, but can improve over the performance of inadequately tuned unfair baselines. OxonFair is compatible with standard ML toolkits including sklearn, Autogluon, and PyTorch and is available at https://github.com/oxfordinternetinstitute/oxonfair.

1. Introduction

The deployment of machine learning systems that make decisions about people offers an opportunity to create systems that work for everyone. However, such systems can lock in existing prejudices. Limited data for underrepresented groups can result in ML systems that do not work for them, while the use of training labels based on historical data can result in ML systems copying previous biases. As such, it is unsurprising that AI systems have repeatedly exhibited unwanted biases towards certain demographic groups in a wide range of domains including medicine (Wen et al., 2022; Obermeyer et al., 2019), finance (Hardt et al., 2016; Martinez & Kirchner, 2021), and policing (Angwin et al., 2022). Such groups are typically identified with respect to legally protected attributes, such as ethnicity or gender (Barocas et al., 2023; Berk et al., 2021; Hardt et al., 2016). The field of algorithmic fairness has sprung up in response to these biases.

Contributions to algorithmic fairness can broadly be split into methodological and policy-based approaches. While much methodological work focuses on measuring and enforcing (un)fairness, a common criticism from the policy side is that this work can occur *'in isolation from policy and civil societal contexts and lacks serious engagement with philosophical, political, legal and economic theories of equality and distributive justice'* (Mittelstadt et al., 2023).

In response to these criticisms, we have developed OxonFair, a more expressive toolkit for algorithmic fairness. We acknowledge that people designing algorithms are not always the right people to decide on policy, and as such we have chosen to create as flexible a toolkit as possible to allow policymakers and data scientists with domain knowledge to identify relevant harms and directly alter the system behaviour to address them. Unlike existing Fairness toolkits such as AIF360 (Bellamy et al., 2018), which take a method-driven approach, and provide access to a wide-range of methods but with limited control over their behaviour, we take a measure-based approach and provide one fairness method that is extremely customizable, and can optimize user-provided objectives and group fairness constraints.

To do this, we focus on one of the oldest and simplest approaches to group fairness: per-group thresholding (Kamiran et al., 2013; Feldman et al., 2015; Hardt et al., 2016), which is known to be optimal for certain metrics under a range of assumptions (Hardt et al., 2016; Corbett-Davies et al., 2017; Lipton et al., 2018). Our contribution is to make this as expressive as possible while retaining speed, for the relatively low number of groups common in algorithmic fairness. Inherently, any approach that allows a sufficiently wide set of objectives, and sets per-group thresholds will be exponential with respect to the number of groups, but we use a standard trick, widely used in the computation of measures such as MAP to make this search as efficient as possible. Accepting the exponential complexity allows us to solve a much wider-range of objectives than other toolkits, including maximizing F1 or balanced accuracy, minimizing difference in precision (Chouldechova, 2017), and guaranteeing that the recall is above k% for every group (Mittel-

[1]University of Oxford. Correspondence to: Eoin Delaney <eoin.delaney@oii.ox.ac.uk>.

International Conference on Machine Learning Workshop on the Next Generation of AI Safety, Copyright 2024 by the author(s).

stadt et al., 2023). Where groups are unavailable at test time, we simply use a secondary classifier to estimate group memberships (Menon & Williamson, 2018; Oneto et al., 2019) and set different thresholds per inferred group to enforce fairness with respect to the true groups.

Thresholding can be applied to most pretrained ML algorithms, and optimal thresholds can be selected using held-out validation data unused in training. This is vital for tasks involving deep networks such as NLP and computer vision, where the training error often goes to zero, and fairness methods that balance error rates between groups can not generalize from constraints enforced on overfitting training data to previously unseen test data (Zietlow et al., 2022). While overfitting is unavoidable in vision and NLP tasks, it is still a concern on tabular data. Section 7 Top-Right, shows examples of decision trees, random forests (Pedregosa et al., 2011) and XGBoost (Chen & Guestrin, 2016) trained with default parameters and obtaining 0 training error on standard datasets. This causes the Fairlearn reduction method (Weerts et al., 2023) to fail in enforcing fairness. NLP and vision are sufficiently challenging that two popular toolkits Fairlearn and AIF360 do not attempt to work with them. In contrast, we make use of a recent work (Lohaus et al., 2022) that showed how a fairness method based on inferred group thresholds can be compressed into a single-headed network.

2. Related Work

Bias mitigation strategies for classification have been broadly categorized into three categories (Barocas et al., 2023; Friedler et al., 2019; Balayn et al., 2021); pre-processing, in-processing and post-processing (See Appendix H for further details). Enforcing fairness on validation data avoids problems caused by the misestimation of error rates due to overfitting. It has shown particular promise in computer vision through Neural Architecture Search (Dutt et al., 2024), adjusting decision boundaries (Lohaus et al., 2022), reweighting (Wang et al., 2024) and data augmentation (Zietlow et al., 2022).

2.1. Fairness Toolkits

Most toolkits such as Fairness Measures (Zehlike et al., 2017), and TensorFlow Fairness Indicators (Xu et al., 2020) focus on measuring bias and do not support enforcing fairness through bias mitigation. FairML (Adebayo et al., 2016) and FairTest (Tramer et al., 2017) investigate the associations between application outcomes (e.g., insurance premiums) and sensitive attributes such as age to highlight and debug bias. Aequitas (Saleiro et al., 2018) provides examples of when different measures are (in)appropriate with support for some bias mitigation methods in binary classification. Themis-ML (Bantilan, 2018) supports the deployment of several simple bias mitigation methods such as relabelling (Kamiran & Calders, 2012), but focuses on linear models. Friedler et al. (2019) introduce the more complete Fairness Comparison toolkit where four bias mitigation strategies are compared across five tabular datasets and multiple models (Decision trees, Naïve Bayes, SVM, and Logistic Regression).

There are two fairness toolkits that support sklearn like OxonFair. These are the two most popular toolkits: Microsoft Fairlearn (Weerts et al., 2023) (1.7k GitHub Stargazers as of May 2024) and IBM AIF360 (Bellamy et al., 2018) (2.3k Stargazers). AIF360 offers a diverse selection of bias measures and pre-processing, in-processing and post-processing bias mitigation strategies on binary classification tabular datasets. For mitigation, Fairlearn primarily offers implementations of (Agarwal et al., 2018; Hardt et al., 2016) avoiding the use of the term *bias*, instead considering fairness through the lens of fairness-related harms (Crawford, 2017) where the goal is to *"help practitioners assess fairness-related harms, review the impacts of different mitigation strategies and make trade-offs appropriate to their scenario"*. Lee & Singh (2021) recognized Fairlearn as one of the most user-friendly fairness toolkits, and critiqued AIF360 as being the least user-friendly toolkit.

Both AIF360 and Fairlearn contain post-processing methods that select per-group thresholds. Unlike OxonFair, neither method uses the fast optimization we propose; both methods require group information at test time; AIF360 only supports two groups, but does use cross-validation to avoid overfitting; Fairlearn does not support the use of validation data, but does support more than two groups. According to their documentation, neither toolkit can be applied to NLP or computer vision.

3. Toolkit interface

The interface of OxonFair decomposes into three parts: (i) evaluation of fairness and performance for generic classifier outputs. (ii) evaluating and enforcing fairness for particular classifiers. (iii) specialist code for evaluating and enforcing fairness for deep networks.

Code for the evaluation of classifier outputs takes target labels, classifier outputs, groups, and an optional conditioning factor as input; while code for the evaluation and enforcement of fairness of a particular classifier, are initialized using the classifier, and from then on take datasets (in the form of a pandas dataframe (pandas development team, 2020), or a dictionary) as input, and automatically extracts these factors from them.

The evaluation code provide three functions: `evaluate` which reports overall performance of the classifier; `evaluate_per_group` which reports performance per group of the classifier; and `evaluate_fairness` which

reports standard fairness metrics. All methods allow the user to specify which metrics should be reported. We recommend data scientists focus on `evaluate_per_group` which shows direct harms such as poor accuracy, precision, or low selection rate for particular groups.

The interface `FairPredictor(classifier, validation_data, groups)` takes an existing classifier as an input, a validation dataset, and specification of the groups as an input and returns an object which we then enforce fairness on by calling `.fit(objective, constraint, value)`. Internally, the method explores a wide range of possible thresholds for each group, membership of which is assumed to be either known or inferred by an auxiliary classifier.

The resulting `FairPredictor` has evaluation methods as described above. When called without arguments, they report both the performance of the original and the updated fair classifier on the validation data. In addition, `FairPredictor` provides methods `predict` and `predict_proba` which make fair predictions and return scores corresponding to the left-hand side of Equation (1).

Calling `fit` optimizes the objective – typically a relevant performance criteria such as accuracy, subject to the requirement that the constraint is either greater or less than the value. If the objective should be minimized or maximized is inferred automatically, as is the requirement that the constraint is less than or greater than the value, but this default behavior can be user overridden.

This is a relatively minimal interface, but one that is surprisingly expressive. By explicitly optimizing an objective, we can not just minimize the degradation of the metric as we enforce fairness, but sometimes also improve performance over the unfair baseline that is not fully optimized with respect to this metric. Even when optimizing for accuracy, this can create situations where it looks like some improvements in fairness can be had for free, although generally this is an artifact of the gap between optimizing log-loss and true accuracy in training. By formulating the problem as a generic constrained optimization, and not requiring the constraint to be a typical fairness constraint, we leave it open for enforcing a much broader space of possible objectives. This can be seen in Section 4.3, where we show how to enforce minimax fairness (Martinez et al., 2020), maximize utility (Bakalar et al., 2021) combined with global recall constraints, and demonstrate levelling-up (Mittelstadt et al., 2023) by specifying minimum acceptable harm thresholds. Under the hood, a call to `fit` generates a Pareto frontier[1] and selects the solution that best optimizes the objective while satisfying the constraint. The frontier can be visualized with `plot_frontier`.

[1]A maximal set of solutions such that for every element in the set, any solution with a better score w.r.t the objective would have a worse score with respect to the constraint, and vice versa.

4. Inference

To make decisions, we assign thresholds to groups.We write $f(x)$ for the response of a classifier f, on datapoint x, t for the vector corresponding to the ordered set of thresholds, and $G(x)$ for the one-hot encoding of group membership. We make a positive decision if

$$f(x) - t \cdot G(x) \geq 0 \tag{1}$$

To optimize arbitrary measures we perform a grid search over the choices of threshold, t.

Efficient grid sampling: We make use of a common trick for efficiently computing measures such as precision and recall for a range of thresholds. This trick is widely used without discussion for efficient computation of the area under ROC curves, and we have had trouble tracking down an original reference for it. As one example, it is used by scikit-learn (Pedregosa et al., 2011). First the datapoints are sorted by classifier response, then a cumulative sum of the number of positive datapoints and the number of negatives, going from greatest response to least is generated. When picking a threshold between points i and $i+1$, TP is given by the cumulative sum of positives in the decreasing direction up to and including i; FN is the sum of negatives in the same direction; FP is the total sum of positives minus TP, and TN is the total sum of negatives minus TN.

We perform this trick per group, and efficiently extract the TP, FN, FP and TN for different thresholds. These are combinatorially combined across the groups and the measures computed. This two stage decoupling offers a substantial speed-up. If we write T for the number of thresholds, k for the number of groups, and n for the total number of datapoints, our procedure is upper-bounded by $O(T^k + n \log n)$, while the naïve approach is $O(nT^k)$. No other fairness method makes use of this, and in particular, all the threshold-based methods offered by AIF360 make use of a naïve grid search. From the grid sampling, we extract a Pareto frontier with respect to the two measures. The thresholds that best optimize the objective while satisfying the constraint are returned as a solution. If no such threshold exists, we return the thresholds closest to satisfying the constraint.

4.1. Inferred characteristics

When using inferred characteristics, we offer two pathways for handling estimated group membership. The first pathway we consider makes a hard assignment of individuals to groups, based on a classifier response. The second pathway explicitly uses the classifier confidence as part of a per-datapoint threshold. In practice, we find little difference between the two approaches, but the hard assignment to

groups is substantially more efficient and therefore allows for a finer grid search and generally better performance. However, the soft assignment remains useful for the integration of our method with neural networks, where we explicitly merge two heads (a classifier and a group predictor) of a neural network to arrive at a single fair model. For details of the two pathways see Appendix A.

4.2. Fairness for Deep Networks

We use the method proposed in (Lohaus et al., 2022). Consider a network with two heads f, and g, comprised of single linear layers, and trained to optimize two tasks on a common backbone B. Here f is a standard classifier trained to maximize some notion of performance such as log-loss and g is another classifier trained to minimize the squared loss[2] with respect to a one hot-encoding of group membership. Any decision $f(x) - t \cdot g(x) \geq 0$ can now be optimized for given criteria by tuning weights w using the process outlined in the slow pathway. As both f and g are linear layers on top of a common backbone, we can write them as:

$$f(x) = w_f \cdot B(x) + b_f, \quad g(x) = w_g \cdot B(x) + b_g \quad (2)$$

note that as $f(x)$ is a real number, and $g(x)$ is a vector w_f is a vector and b_f a real number, while w_g is a matrix and b_g a vector. This means that the decision function $f(x) - t \cdot g(x) \geq 0$ can be rewritten using the identity:

$$\begin{aligned} f(x) - t \cdot g(x) &= w_f \cdot B(x) + b_f - t \cdot w_g \cdot B(x) - t \cdot b_g \\ &= (w_f - t \cdot w_g) \cdot B(x) + (b_f - t \cdot b_g) \end{aligned} \quad (3)$$

This gives a 3 stage process for enforcing any of these decision/fairness criteria for deep networks.

1. Train a multitask neural network as described above.
2. Compute the optimal thresholds t on held-out validation data as described in Appendix A.
3. Replace the multitask head with a neuron with weights $(w_f - t \cdot w_g)$ and bias $(b_f - t \cdot b_g)$.

OxonFair has a distinct interface for deep learning. Training and evaluating NLP and vision frequently involves complex pipelines. To maximize applicability, we assume that the user has trained a two-headed network as described above, and evaluated on a validation set. Our constructor `DeepFairPredictor` requires: the output of the two-headed network over the validation set; the ground-truth labels; and the groups as inputs. `fit` and the evaluation functionality can then be called in the same way. Once a solution is found, the method `extract_coefficients` can be called to extract the weights from Equation 3.

[2]The squared loss is used rather than log-loss so that the output of $g(x)$ remains close to 0 and 1. With log-loss, the output pre-sigmoid is more likely to overwhelm confident decisions made by the original classifier.

4.3. Toolkit expressiveness

Out of the box, OxonFair supports all 9 of the decision-based group fairness measures defined by Verma & Rubin (2018) and all 10 of the fairness measures from Sagemaker Clarify (Das et al., 2021). OxonFair supports any fairness measure (including conditional fairness measures) that can be expressed per group as a weighted sum of True Positives, False Positives, True Negatives and False Negatives. OxonFair does not support notions of individual fairness such as fairness through awareness (Dwork et al., 2012).

See Appendix B for a discussion of how metrics are implemented and comparison with two review papers. Appendix C contains details of non-standard fairness metrics, including utility optimization (Bakalar et al., 2021); minimax fairness (Martinez et al., 2020; Diana et al., 2021; Abernethy et al., 2022); minimum rate constraints (Mittelstadt et al., 2023), Conditional Demographic Parity (Wachter et al., 2021) and Directional Bias Amplification (Zhao et al., 2017; Wang & Russakovsky, 2021).

Table 1: A comparison against standard approaches on CelebA attributes. ERM is the baseline architecture run without fairness. OxonFair (optimizing for accuracy and DEO), has better DEO scores than any other fair method. An extended table 13 shows results for minimax fairness.

	ERM	Uniconf. Adv	Domain Disc.	Domain Ind.	OxonFair DEO
Gender-Independent Attributes					
Acc.	**93.1**	92.7	93.0	92.6	92.8
DEO	16.5	19.6	14.6	7.78	**3.21**
Gender-Dependent Attributes					
Acc.	**86.7**	86.1	86.6	85.6	85.8
DEO	26.4	25.0	21.9	6.50	**3.92**
Inconsistently Labelled Attributes					
Acc.	83.0	82.5	**83.1**	82.3	82.1
DEO	21.9	29.1	25.3	17.2	**2.36**

5. Experimental Analysis

In this section we discuss experimental results from high dimensional domains. Tabular results including comparisons to other toolkits can be found in Appendix D.

Table 2: Multilingual Twitter dataset: Gender

	F1	Bal-Acc.	Acc.	DEO
Base	40.8	63.2	89.8	21.4
CDA	43.2	64.4	89.8	16.0
DP	37.2	61.7	89.5	17.9
EO	32.0	59.6	89.1	13.2
Dropout	32.2	59.8	88.9	13.8
Rebalance	38.2	62.1	89.5	19.1
OxonFair (Acc.)	34.1	60.7	88.5	8.45
OxonFair (F1)	44.6	69.1	84.7	2.10
OxonFair (Bal. Acc.)	47.1	71.2	84.8	7.33

Table 3: Jigsaw dataset: Religion

	F1	Bal. Acc.	Acc.	DEO
Base	42.1	74.8	75.0	7.33
CDA	40.4	73.8	73.0	8.98
DP	44.5	69.2	85.5	3.68
EO	41.1	68.8	82.2	4.60
Dropout	42.7	74.1	77.0	7.94
Rebalance	39.1	73.7	70.3	9.67
OxonFair (Acc.)	33.7	60.5	89.2	2.36
OxonFair (F1)	44.4	69.5	85.0	3.79
OxonFair (Bal. Acc.)	42.2	74.2	76.1	4.78

5.1. Computer Vision and CelebA

CelebA (Liu et al., 2015): We use the standard aligned & cropped partitions frequently used in fairness evaluation (Zietlow et al., 2022; Wang et al., 2020). Following Ramaswamy et al. (2021), we consider the 26 *gender-independent*, *gender-dependent* and *inconsistently labelled* attributes as the target attributes (see Table 10 for details). *Male* is treated as the protected attribute. We follow the setup of Wang et al. (2020) using a Resnet-50 backbone (He et al., 2016) trained on ImageNet (Deng et al., 2009). A multitask classification model is trained, replacing the final fully-connected layer of the backbone with a separate fully-connected head that performs binary prediction for all attributes (See Appendix E for methods and full details).

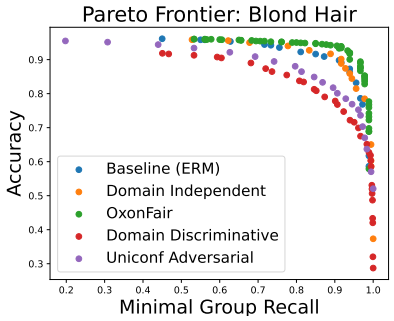


Figure 1: The Pareto frontier of min. group recall vs. accuracy on the `Blond Hair` attribute demonstrates OxonFair's superior performance.

CelebA	$\delta = 0.50$	$\delta = 0.75$	$\delta = 0.90$
Baseline	89.0	84.5	77.6
Adversarial	87.8	82.4	75.2
Domain-Dep	82.3	76.8	68.6
Domain-Ind	89.2	86.2	79.8
OxonFair	**89.9**	**87.3**	**81.8**

Table 4: Accuracy of fairness methods on 26 CelebA attributes while varying global decision thresholds to increase the minimum group recall level to δ.

Results: Tables 1 and 11 demonstrate that using OxonFair as described in section 4.2 generates fairer and more accurate solutions on unseen test data than other fair methods. Simple approaches such as Domain Independent training were more effective than adversarial training for enforcing fairness confirming (Han et al., 2024). Table 4 shows a novel fairness evaluation motivated by medical use cases (Mittelstadt et al., 2023; Zong et al., 2023) where practitioners might want to correctly identify at least $\delta\%$ of positive cases in each group. We evaluate how accuracy changes if we guarantee that the minimum recall is above $\delta\%$ for every group. For OxonFair, we call `.fit(accuracy, recall.min, `δ`)`. For other methods, we vary a global offset to ensure that the minimum recall is at least δ.

5.2. NLP and Toxic Content

We conducted experiments on hate speech detection and toxicity classification using two datasets: the Multilingual Twitter corpus (Huang et al., 2020) and Jigsaw (Jigsaw, 2018). Experiments were performed across five languages and five demographic factors were treated as the protected groups. We compare OxonFair with several popular approaches including the standard BERT Baseline, Counterfactual Data Augmentation **CDA** (Zmigrod et al., 2019), Demographic Parity **DP**, Equal Oportunity **EO** and **Dropout** regularization to enforce fairness, and **Rebalance** (Feldman et al., 2015; Li & Vasconcelos, 2019). For full experimental details, see Appendix F.1. We report scores optimized for different metrics: Accuracy; F1; and Balanced Accuracy.

Results are shown in Tables 2 and 3. Our observations indicate that: 1) all debiasing methods improve the equal opportunity score and help mitigate bias on Twitter, but not on Jigsaw. 2) our toolkit consistently reduces the difference in equal opportunity more than any other approach; 3) for 4/6 experiments we actually improve the objective over the baseline while enforcing fairness, showing the value in targeting a particular objective. For additional experiments on multilingual and multi-demographic data, and the Jigsaw race data, see Appendix F.3, and Appendix F.4.

6. Conclusion

The key contributions of our toolkit lie in being more expressive than other approaches, and supporting NLP and computer vision. Despite this, most of the experiments focus on the standard definitions of Demographic Parity and Equal Opportunity. This is not because we agree that they are the right measures, but because we believe that the best way to show that OxonFair works is to compete with other methods in what they do best. On low-dimensional tabular data, when optimizing accuracy and a standard fairness measure, it is largely comparable with Fairlearn, but if overfitting or non-standard performance criteria and fairness metrics are a concern, then OxonFair has obvious advantages. For NLP, and computer vision, our approach clearly improves on existing state-of-the-art. In no small part, this is due to the observation of (Zietlow et al., 2022), that methods for estimating or enforcing error-based fairness metrics on high-capacity models that do not use held-out validation data can not work.

We hope that OxonFair will free policy-makers and domain experts to directly specify fairness measures and objectives that are a better match for the harms that they face. In particular, we want to call out the measures in fig. 5 as relevant to medical ML. The question of how much accuracy can we retain, while guaranteeing that test sensitivity (AKA recall) is above k% for every group, captures notions of fairness and clinical relevance in a way that standard fairness notions do not (Mittelstadt et al., 2023). We join growing calls of Balayn et al. (2023) in encouraging practitioners to be reflective in their use of fairness toolkits and the associated harms.

Limitations: We have chosen to optimize as broad a set of formulations as possible. As a result, for certain metrics (particularly equalized odds (Hardt et al., 2016)) the solutions found are known to be suboptimal; and for others (Corbett-Davies et al., 2017) the exponential search is unneeded. Techniques targeting particular formulations may be needed to address this. A major driver of unfairness is a lack of data regarding particular groups. However, this very absence of data makes it hard for any toolkit to detect or rectify unfairness.

Broader Impact: OxonFair is a tool for altering the decisions made by ML systems that are frequently trained on biased data. Care must be taken that fair ML is used as a final step after correcting for bias and errors in data collation, and not as a sticking plaster to mask problems (Balayn et al., 2023). Indeed, inappropriate uses of fairness can lock in biases present in training (Wachter et al., 2020). Under the hood, OxonFair performs a form of positive discrimination, where we alter scores in response to (perceived) protected characteristics to rectify particular inequalities[3]. As such, there are many scenarios where its use may be inappropriate for legal or ethical reasons.

7. Acknowledgements

This work has been supported through research funding provided by the Wellcome Trust (grant nr 223765/Z/21/Z), Sloan Foundation (grant nr G-2021-16779), Department of Health and Social Care, EPSRC (grant nr EP/Y019393/1), and Luminate Group. Their funding supports the Trustworthiness Auditing for AI project and Governance of Emerging Technologies research programme at the Oxford Internet Institute, University of Oxford.

An early prototype version of the same toolkit, for tabular data, was developed while CR was working at AWS and is available online as autogluon.fair (https://github.com/autogluon/autogluon-fair/). CR is grateful to Nick Erickson and Weisu Yin for code reviews of the prototype. The authors thank Kaivalya Rawal for feedback on the manuscript and testing the codebase.

References

Abernethy, J. D., Awasthi, P., Kleindessner, M., Morgenstern, J., Russell, C., and Zhang, J. Active sampling for min-max fairness. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 53–65. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/abernethy22a.html.

Adebayo, J. A. et al. Fairml: Toolbox for diagnosing bias in predictive modeling, 2016.

Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., and Wallach, H. A reductions approach to fair classification. In *International conference on machine learning*, pp. 60–69. PMLR, 2018.

Alvi, M., Zisserman, A., and Nellåker, C. Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pp. 0–0, 2018.

Angwin, J., Larson, J., Mattu, S., and Kirchner, L. Machine bias. In *Ethics of data and analytics*, pp. 254–264. Auerbach Publications, 2022.

Bakalar, C., Barreto, R., Bergman, S., Bogen, M., Chern,

[3]However, see (Lohaus et al., 2022) for how other fairness methods may also be doing this.

B., Corbett-Davies, S., Hall, M., Kloumann, I., Lam, M., Candela, J. Q., et al. Fairness on the ground: Applying algorithmic fairness approaches to production systems. *arXiv preprint arXiv:2103.06172*, 2021.

Balayn, A., Lofi, C., and Houben, G.-J. Managing bias and unfairness in data for decision support: a survey of machine learning and data engineering approaches to identify and mitigate bias and unfairness within data management and analytics systems. *The VLDB Journal*, 30(5):739–768, 2021.

Balayn, A., Yurrita, M., Yang, J., and Gadiraju, U. "☑ fairness toolkits, a checkbox culture?" on the factors that fragment developer practices in handling algorithmic harms. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 482–495, 2023.

Bantilan, N. Themis-ml: A fairness-aware machine learning interface for end-to-end discrimination discovery and mitigation. *Journal of Technology in Human Services*, 36 (1):15–30, 2018.

Barikeri, S., Lauscher, A., Vulić, I., and Glavaš, G. Redditbias: A real-world resource for bias evaluation and debiasing of conversational language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1941–1955, 2021.

Barocas, S., Hardt, M., and Narayanan, A. *Fairness and machine learning: Limitations and opportunities*. MIT Press, 2023.

Bellamy, R. K., Dey, K., Hind, M., Hoffman, S. C., Houde, S., Kannan, K., Lohia, P., Martino, J., Mehta, S., Mojsilovic, A., et al. Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943*, 2018.

Berk, R., Heidari, H., Jabbari, S., Kearns, M., and Roth, A. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, 50(1):3–44, 2021.

Bickel, P. J., Hammel, E. A., and O'Connell, J. W. Sex bias in graduate admissions: Data from berkeley: Measuring bias is harder than is usually assumed, and the evidence is sometimes contrary to expectation. *Science*, 187(4175): 398–404, 1975.

Calmon, F., Wei, D., Vinzamuri, B., Natesan Ramamurthy, K., and Varshney, K. R. Optimized pre-processing for discrimination prevention. *Advances in neural information processing systems*, 30, 2017.

Cha, J., Chun, S., Lee, K., Cho, H.-C., Park, S., Lee, Y., and Park, S. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34:22405–22418, 2021.

Chakraborty, J., Majumder, S., and Menzies, T. Bias in machine learning software: Why? how? what to do? In *Proceedings of the 29th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering*, pp. 429–440, 2021.

Chen, T. and Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

Chiappa, S. Path-specific counterfactual fairness. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 7801–7808, 2019.

Chouldechova, A. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017.

Chuang, C.-Y. and Mroueh, Y. Fair mixup: Fairness via interpolation. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=DNl5s5BXeBn.

Corbett-Davies, S., Pierson, E., Feller, A., Goel, S., and Huq, A. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*, pp. 797–806, 2017.

Crawford, K. The trouble with bias - nips 2017 keynote. https://www.youtube.com/watch?v=fMym_BKWQzk, 2017.

Das, S., Donini, M., Gelman, J., Haas, K., Hardt, M., Katzman, J., Kenthapadi, K., Larroy, P., Yilmaz, P., and Zafar, B. Fairness measures for machine learning in finance. *The Journal of Financial Data Science*, 2021.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Diana, E., Gill, W., Kearns, M., Kenthapadi, K., and Roth, A. Minimax group fairness: Algorithms and experiments. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 66–76, 2021.

Dinan, E., Fan, A., Williams, A., Urbanek, J., Kiela, D., and Weston, J. Queens are powerful too: Mitigating gender bias in dialogue generation. *arXiv preprint arXiv:1911.03842*, 2019.

Dutt, R., Bohdal, O., Tsaftaris, S., and Hospedales, T. Fairtune: Optimizing parameter efficient fine tuning for fairness in medical image analysis. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ArpwmicoYW.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pp. 214–226, 2012.

Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., and Smola, A. Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505*, 2020.

Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., and Venkatasubramanian, S. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 259–268, 2015.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Freedman, D., Pisani, R., and Purves, R. *Statistics*. W. W. Norton & Company, New York, fourth edition, 2007. ISBN 978-0-393-92972-0. Hardcover, 700 pages.

Friedler, S. A., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E. P., and Roth, D. A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the conference on fairness, accountability, and transparency*, pp. 329–338, 2019.

Goethals, S., Calders, T., and Martens, D. Beyond accuracy-fairness: Stop evaluating bias mitigation methods solely on between-group metrics. *arXiv preprint arXiv:2401.13391*, 2024a.

Goethals, S., Delaney, E., Mittelstadt, B., and Russell, C. Resource-constrained fairness. *arXiv preprint arXiv:2406.01290*, 2024b.

Golovenkin, S. E., Bac, J., Chervov, A., Mirkes, E. M., Orlova, Y. V., Barillot, E., Gorban, A. N., and Zinovyev, A. Trajectories, bifurcations, and pseudo-time in large clinical datasets: applications to myocardial infarction and diabetes data. *GigaScience*, 9(11):giaa128, 2020.

Groh, M., Harris, C., Soenksen, L., Lau, F., Han, R., Kim, A., Koochek, A., and Badri, O. Evaluating deep neural networks trained on clinical images in dermatology with the fitzpatrick 17k dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1820–1828, 2021.

Han, X., Chi, J., Chen, Y., Wang, Q., Zhao, H., Zou, N., and Hu, X. FFB: A fair fairness benchmark for in-processing group fairness methods. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=TzAJbTClAz.

Hardt, M., Price, E., and Srebro, N. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

Hardt, M., Chen, X., Cheng, X., Donini, M., Gelman, J., Gollaprolu, S., He, J., Larroy, P., Liu, X., McCarthy, N., et al. Amazon sagemaker clarify: Machine learning bias detection and explainability in the cloud. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2974–2983, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Huang, X., Xing, L., Dernoncourt, F., and Paul, M. Multilingual twitter corpus and baselines for evaluating demographic bias in hate speech recognition. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 1440–1448, 2020.

Jigsaw. Jigsaw unintended bias in toxicity classification, 2018. URL https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification

Kamiran, F. and Calders, T. Data preprocessing techniques for classification without discrimination. *Knowledge and information systems*, 33(1):1–33, 2012.

Kamiran, F., Žliobaitė, I., and Calders, T. Quantifying explainable discrimination and removing illegal discrimination in automated decision making. *Knowledge and information systems*, 35:613–644, 2013.

Kamishima, T., Akaho, S., Asoh, H., and Sakuma, J. Fairness-aware classifier with prejudice remover regularizer. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II 23*, pp. 35–50. Springer, 2012.

Kearns, M., Neel, S., Roth, A., and Wu, Z. S. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International conference on machine learning*, pp. 2564–2572. PMLR, 2018.

Kim, B., Kim, H., Kim, K., Kim, S., and Kim, J. Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9012–9020, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kwegyir-Aggrey, K., Dai, J., Cooper, A. F., Dickerson, J., Hines, K., and Venkatasubramanian, S. Repairing regressors for fair binary classification at any decision threshold. In *NeurIPS 2023 Workshop Optimal Transport and Machine Learning*, 2023.

Lee, M. S. A. and Singh, J. The landscape and gaps in open source fairness toolkits. In *Proceedings of the 2021 CHI conference on human factors in computing systems*, pp. 1–13, 2021.

Li, Y. and Vasconcelos, N. Repair: Removing representation bias by dataset resampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9572–9581, 2019.

Lipton, Z., McAuley, J., and Chouldechova, A. Does mitigating ml's impact disparity require treatment disparity? *Advances in neural information processing systems*, 31, 2018.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Lohaus, M., Perrot, M., and Von Luxburg, U. Too relaxed to be fair. In *International Conference on Machine Learning*, pp. 6360–6369. PMLR, 2020.

Lohaus, M., Kleindessner, M., Kenthapadi, K., Locatello, F., and Russell, C. Are two heads the same as one? identifying disparate treatment in fair neural networks. *Advances in Neural Information Processing Systems*, 35: 16548–16562, 2022.

Madras, D., Creager, E., Pitassi, T., and Zemel, R. Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*, pp. 3384–3393. PMLR, 2018.

Martinez, E. and Kirchner, L. The secret bias hidden in mortgage-approval algorithms. *The Markup*, 2021.

Martinez, N., Bertran, M., and Sapiro, G. Minimax pareto fairness: A multi objective perspective. In *International Conference on Machine Learning*, pp. 6755–6764. PMLR, 2020.

Meade, N., Poole-Dayan, E., and Reddy, S. An empirical survey of the effectiveness of debiasing techniques for pre-trained language models. *arXiv preprint arXiv:2110.08527*, 2021.

Menon, A. K. and Williamson, R. C. The cost of fairness in binary classification. In *Conference on Fairness, accountability and transparency*, pp. 107–118. PMLR, 2018.

Mittelstadt, B., Wachter, S., and Russell, C. The unfairness of fair machine learning: Levelling down and strict egalitarianism by default. *arXiv preprint arXiv:2302.02404*, 2023.

Obermeyer, Z., Powers, B., Vogeli, C., and Mullainathan, S. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453, 2019.

Oneto, L., Doninini, M., Elders, A., and Pontil, M. Taking advantage of multitask learning for fair classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 227–237, 2019.

pandas development team, T. pandas-dev/pandas: Pandas, February 2020. URL https://doi.org/10.5281/zenodo.3509134.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Plečko, D. and Meinshausen, N. Fair data adaptation with quantile preservation. *Journal of Machine Learning Research*, 21(242):1–44, 2020.

Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J., and Weinberger, K. Q. On fairness and calibration. *Advances in neural information processing systems*, 30, 2017.

Ramaswamy, V. V., Kim, S. S., and Russakovsky, O. Fair attribute classification through latent space de-biasing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9301–9310, 2021.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Saleiro, P., Kuester, B., Hinkson, L., London, J., Stevens, A., Anisfeld, A., Rodolfa, K. T., and Ghani, R. Aequitas: A bias and fairness audit toolkit. *arXiv preprint arXiv:1811.05577*, 2018.

Sarhan, M. H., Navab, N., Eslami, A., and Albarqouni, S. Fairness by learning orthogonal disentangled representations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, pp. 746–761. Springer, 2020.

Singh, H., Kleindessner, M., Cevher, V., Chunara, R., and Russell, C. When do minimax-fair learning and empirical risk minimization coincide? In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31969–31989. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/singh23b.html.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Tartaglione, E., Barbano, C. A., and Grangetto, M. End: Entangling and disentangling deep representations for bias correction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13508–13517, 2021.

Tramer, F., Atlidakis, V., Geambasu, R., Hsu, D., Hubaux, J.-P., Humbert, M., Juels, A., and Lin, H. Fairtest: Discovering unwarranted associations in data-driven applications. In *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 401–416. IEEE, 2017.

Vapnik, V. N. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

Verma, S. and Rubin, J. Fairness definitions explained. In *Proceedings of the international workshop on software fairness*, pp. 1–7, 2018.

Wachter, S., Mittelstadt, B., and Russell, C. Bias preservation in machine learning: the legality of fairness metrics under eu non-discrimination law. *W. Va. L. Rev.*, 123:735, 2020.

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

Wang, A. and Russakovsky, O. Directional bias amplification. In *International Conference on Machine Learning*, pp. 10882–10893. PMLR, 2021.

Wang, H., Wu, Z., and He, J. Fairif: Boosting fairness in deep learning via influence functions with validation set sensitive attributes. *arXiv preprint arXiv:2201.05759v2*, 2024.

Wang, Z., Qinami, K., Karakozis, I. C., Genova, K., Nair, P., Hata, K., and Russakovsky, O. Towards fairness in visual recognition: Effective strategies for bias mitigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8919–8928, 2020.

Watkins, E. A., McKenna, M., and Chen, J. The four-fifths rule is not disparate impact: a woeful tale of epistemic trespassing in algorithmic fairness. *arXiv preprint arXiv:2202.09519*, 2022.

Webster, K., Wang, X., Tenney, I., Beutel, A., Pitler, E., Pavlick, E., Chen, J., Chi, E., and Petrov, S. Measuring and reducing gendered correlations in pre-trained models. *arXiv preprint arXiv:2010.06032*, 2020.

Weerts, H., Dudak, M., Edgar, R., Jalali, A., Lutz, R., and Madaio, M. Fairlearn: Assessing and improving fairness of ai systems. *Journal of Machine Learning Research*, 24(257):1–8, 2023. URL http://jmlr.org/papers/v24/23-0389.html.

Wen, D., Khan, S. M., Xu, A. J., Ibrahim, H., Smith, L., Caballero, J., Zepeda, L., de Blas Perez, C., Denniston, A. K., Liu, X., et al. Characteristics of publicly available skin cancer image datasets: a systematic review. *The Lancet Digital Health*, 4(1):e64–e74, 2022.

Xu, C., Greer, C., Joshi, M. N., and Doshi, T. Fairness indicators demo: Scalable infrastructure for fair ml systems, 2020.

Zafar, M. B., Valera, I., Rogriguez, M. G., and Gummadi, K. P. Fairness constraints: Mechanisms for fair classification. In *Artificial intelligence and statistics*, pp. 962–970. PMLR, 2017.

Zehlike, M., Castillo, C., Bonchi, F., Hajian, S., and Megahed, M. Fairness measures: Datasets and software for detecting algorithmic discrimination. *URL http://fairness-measures. org*, 2017.

Zemel, R., Wu, Y., Swersky, K., Pitassi, T., and Dwork, C. Learning fair representations. In *International conference on machine learning*, pp. 325–333. PMLR, 2013.

Zhang, B. H., Lemoine, B., and Mitchell, M. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 335–340, 2018.

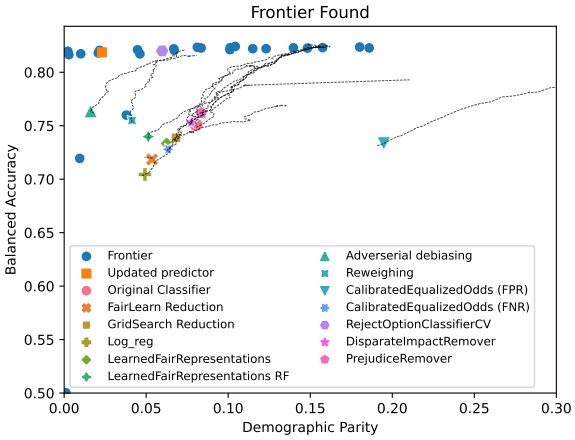
Zhao, H., Coston, A., Adel, T., and Gordon, G. J. Conditional learning of fair representations. *arXiv preprint arXiv:1910.07162*, 2019.

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., and Chang, K.-W. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *EMNLP*, 2017.

Zietlow, D., Lohaus, M., Balakrishnan, G., Kleindessner, M., Locatello, F., Schölkopf, B., and Russell, C. Leveling down in computer vision: Pareto inefficiencies in fair deep classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10410–10421, 2022.

Zmigrod, R., Mielke, S. J., Wallach, H., and Cotterell, R. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. *arXiv preprint arXiv:1906.04571*, 2019.

Zong, Y., Yang, Y., and Hospedales, T. Medfair: Benchmarking fairness for medical imaging. In *International Conference on Learning Representations (ICLR)*, 2023.



Classifier (dataset)	Partition	Fairlearn		OxonFair	
		Acc	EOp	Acc	EOp
Decision Tree	Train/Val	100%	0%	82%	2.0%
(Adult)	Test	81%	8.8%	81%	1.1%
Random Forest	Train/Val	100%	0%	86%	1.6%
(Adult)	Test	86%	7.5%	86%	3.3%
XGBoost	Train/Val	100%	0%	90%	0.6%
(myocardial infarction)	Test	89%	11.8%	87%	2.9%

Criterion	AIF360	Fairlearn	OxonFair
Number of methods	10+	5	1
Adjustible Fairness Criteria	×	✓	✓
Supports 3+ Groups	×	✓	✓
Fairness definitions enforced per method	<4	5	14+
Methods needing groups at eval	Some	1	No
Supports Utility Functions	×	×	✓
Supports Tabular Data	✓	✓	✓
Supports Computer Vision	×	×	✓
Supports NLP	×	×	✓

Figure 2: **Left:** The need for an objective when enforcing fairness. We evaluate a range of methods with respect to balanced accuracy and demographic parity (OxonFair generates a frontier of solutions). Only OxonFair and RejectOptimization optimize balanced accuracy. As we improve the balanced accuracy of fair methods by adjusting classification thresholds (gray lines) fairness deteriorates. To avoid this, we jointly optimize a fairness measure and an objective. For more examples, see Figure 4. **Right Top:** Using validation data in fairness. We compare against Fairlearn using standard algorithms with default parameters. These methods perfectly overfit and show no unfairness with respect to equal opportunity on the trainset, but substantial unfairness on test. OxonFair enforces fairness on held-out validation data and is less prone to overfitting. **Right Bottom:** A comparison of toolkits. AIF360 offers a large range of tabular methods most of which do not allow fairness metric selection, Fairlearn offers fewer but more customizable tabular methods. OxonFair offers one method that can be applied to text, image, and tabular data, while supporting more notions of fairness and objectives.

A. Inferred characteristics

In many situations, protected attributes are not available at test time. In this case, we simply use inferred characteristics to assign per-group thresholds and adjust these thresholds to guarantee fairness with respect to the true (i.e. uninferred) groups.

When using inferred characteristics, we offer two pathways for handling estimated group membership. The first pathway we consider makes a hard assignment of individuals to groups, based on a classifier response. The second pathway explicitly uses the classifier confidence as part of a per-datapoint threshold. In practice, we find little difference between the two approaches, but the hard assignment to groups is substantially more efficient and therefore allows for a finer grid search and generally better performance. However, the soft assignment remains useful for the integration of our method with neural networks, where we explicitly merge two heads of a neural network to arrive at a single fair model.

A.1. Fast pathway

The fast pathway closely follows the efficient grid search for known characteristics. We partition the dataset by inferred characteristics, and then repeat the trick. However, as the inferred characteristics do not need to perfectly align with the true characteristics, we also keep track of the true group datapoints belongs to, i.e., for all datapoints assigned to a particular inferred group, we compute the cumulative sum of positives and negatives that truly belong to each group. This allows us to vary the thresholds with respect to inferred groups while computing group measures with respect to the true groups. This can be understood as replacing the decision function (1) with $f(x) - t \cdot G'(x) \geq 0$ where G' is a binary vector valued function that sums to 1, but need not correspond to G exactly.

This explicit decoupling of inferred groups from the true group membership allows us to consider partitionings of the data that do not align with group membership. We found it particularly helpful to include an additional 'don't know' group. By default, any datapoint assigned a score[4] from the classifier below $2/3$ is assigned to this group, and receives a different threshold to those datapoints that the classifier is confident about. The improved frontiers are shown in the tabular

[4]User controllable threshold.

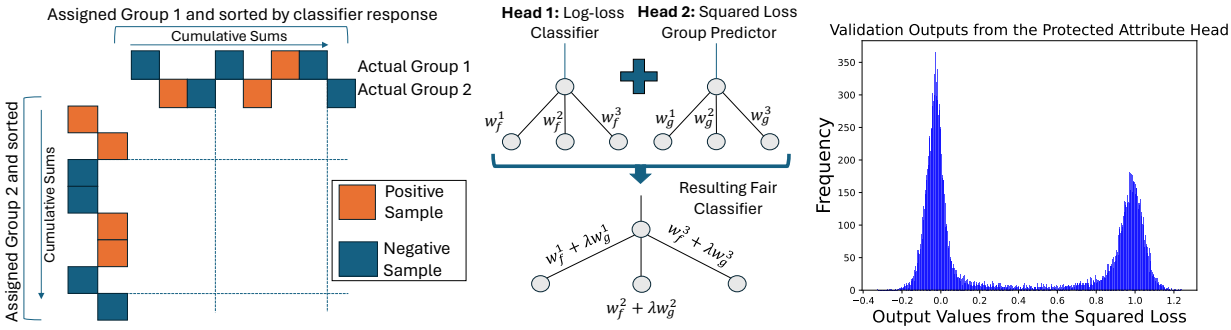


Figure 3: **Left:** Summary of the fast path algorithm for inferred attributes (Section 4.1). Groups are noisily estimated using a classifier. Within each estimated group, we cumulatively sum positive and negative samples that truly belong to each group. For each pair of thresholds, we select relevant sums from the inferred group and combine them. See Appendix A.1. **Center:** Combining two heads (original classifier and group predictor) to create a fair classifier. See Section 4.2. **Right:** The output of a second head predicting the protected attribute in CelebA. The pronounced bimodal distribution makes the weighted sum of the two heads a close replacement for per-group thresholds.

experimental section as OxonFair+, where they offer a clear advantage over our baseline OxonFair.

A.2. Slow pathway

The slow pathway tunes t to optimize the decision process $f(x) - t \cdot g(x) \geq 0$, where g is a real vector valued function. Given the lack of assumptions, no obvious speed-up was possible and we perform a two stage naïve grid-search, first coarsely to extract an approximate Pareto frontier, and then a finer search over the range of thresholds found in the first stage. This is then followed by a final interpolation that checks for candidates around pairs of adjacent candidates currently in the frontier.

In situations where $g(x)$ is the output of a classifier and $G'(x)$ its binarization, it is reasonable to suspect that loss of information from binarization might lead to a drop in performance when we compare the slow pathway with the fast. In practice, we never found a significant change, and in a like-with-like comparison over a similar number of thresholds the fast pathway was as likely to be fractionally better as it was to be worse. Moreover, for more than 3 groups the slow pathway becomes punitively slow, and to keep the runtime acceptable requires decreasing the grid size in a way that harms performance.

Despite this, we kept the slow pathway as it is directly applicable to deep networks as we describe in the next section. In practice, when working with deep networks we make use of a hybrid approach, and perform the fast and slow grid searches before fusing them into a single frontier and then performing interpolation. This allows us to benefit from the better solutions found by a fine grid search when the output of the second head is near binary (see Figure 3), and robustly carry over to the slower pathway where its binarization is a bad approximation of the network output.

B. Implementation of Performance and Fairness Measures

To make OxonFair as extensible as possible, we create a custom class to implement all performance and fairness measures. This means when OxonFair doesn't support a particular measure, both the objectives and constraints can be readily extended by the end user.

Measures used by OxonFair are defined as instances of a python class `GroupMetrics`. Each group measure is specified by a function that takes the number of True Positives, False Positives, False Negatives, and True Negatives and returns a score; A string specifying the name of the measure; and optionally a bool indicating if greater values are better than smaller ones.

For example, accuracy is defined as:

```
accuracy = gm.GroupMetric(lambda TP, FP, FN, TN: (TP + TN) / (TP + FP + FN + TN),
'Accuracy')
```

For efficiency, our approach relies on broadcast semantics and all operations in the function must be applicable to numpy

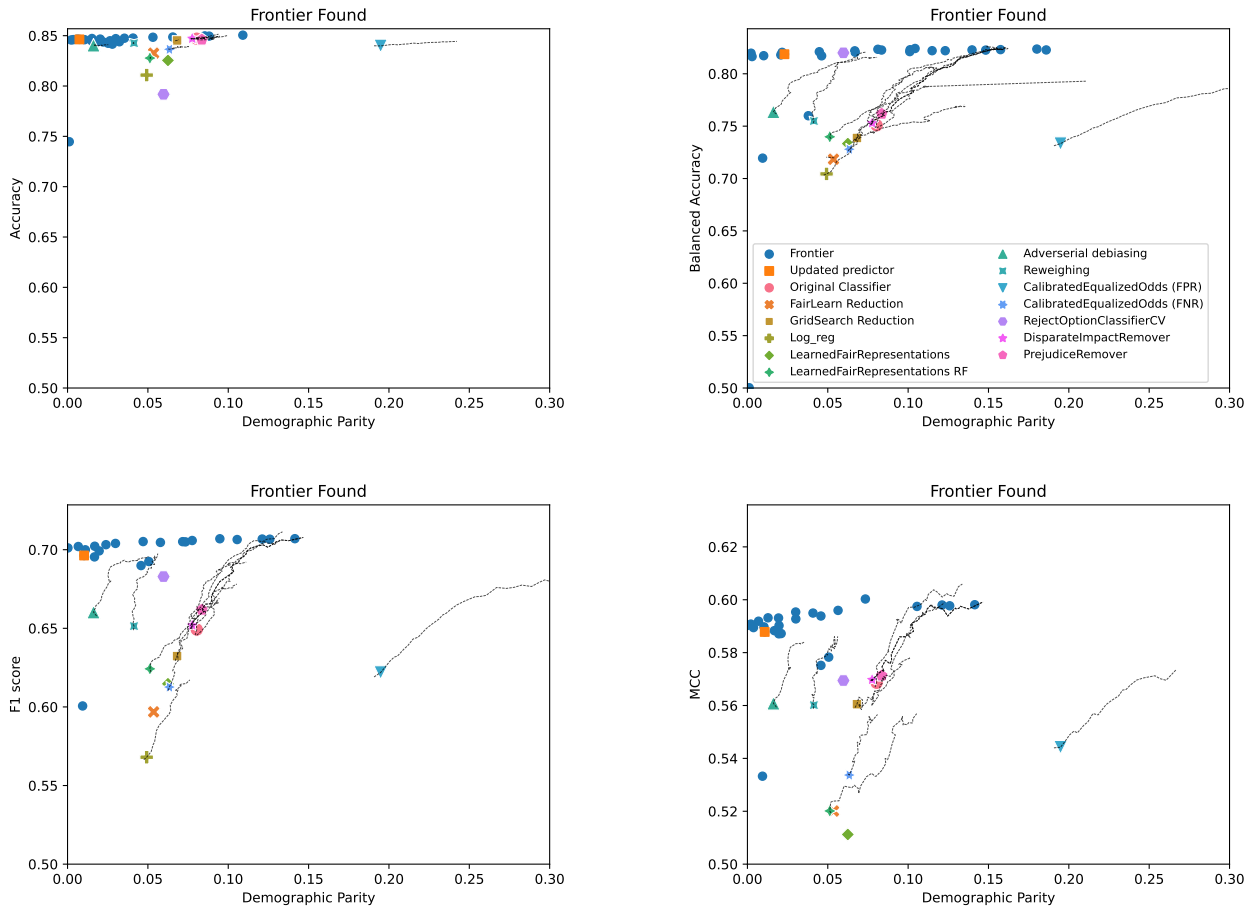


Figure 4: We show a full comparison of the methods provided by AIF360 and Fairlearn on the adult dataset with 4 different choices of metric (accuracy, balanced accuracy, F1, and MCC), while enforcing demographic parity. We follow the design decisions of (Bellamy et al., 2018) and use a random forest with 100 trees and a minimum leaf size of 20. Only OxonFair allows the specification of an objective, and for all other methods we try to alter the decision threshold to better optimize the objective. However, as we improve the objective, we see fairness deteriorates. To avoid this, OxonFair jointly optimize both a fairness measure and an objective.

arrays. Having defined a GroupMetric it can be called in two ways. Either:

```
accuracy(target_labels, predictions, groups)
```

Here `target_labels` and `predictions` are binary vectors corresponding to either the target ground-truth values, or the predictions made by a classifier, with 1 representing the positive label and 0 otherwise. `groups` is simply a vector of values where each unique value is assumed to correspond to a distinct group.

The other way it can be called is by passing it a single 3D array of dimension 4 by number of groups by k, where k is the number of candidate classifiers that the measure should be computed over.

As a convenience, GroupMetrics automatically implements a range of functionality as sub-objects.

Having defined a metric as above, we have a range of different objects:

- `metric.diff` reports the average absolute difference of the method between groups.
- `metric.average` reports the average of the method taken over all groups.
- `metric.max_diff` reports the maximum difference of the method between any pair of groups.
- `metric.max` reports the maximum value for any group.
- `metric.min` reports the minimum value for any group.
- `metric.overall` reports the overall value for all groups combined, and is the same as calling metric directly
- `metric.ratio` reports the average over distinct pairs of groups of the smallest value divided by the largest
- `metric.per_group` reports the value for every group.

All of these can be passed directly to fit, or to the evaluation functions we provide.

The vast majority of fairness metrics are implemented as a `.diff` of a standard performance measure, and by placing a `.min` after any measure such as recall or precision it is possible to add constraints that enforce that the precision or recall is above a particular value for every group.

These classes make it easy to extend OxonFair. To demonstrate the OxonFair's versatility, Tables 5 and 6 show the metrics of two reviews and how many can are implemented out of the box by our approach. An example showing how all clarify metrics can be enforced using inferred groups, and three group labels on compas can be seen in Table 7.

C. Additional Metrics

C.1. Minimax Fairness

Minimax fairness(Martinez et al., 2020; Diana et al., 2021; Abernethy et al., 2022) refers to the family of methods which minimize the loss of the group where the algorithm performs worst, i.e., they minimize the maximal loss. (Singh et al., 2023) observed that sufficiently expressive classifiers, such as those considered by this paper, including boosting, random forests, or deep networks on image and NLP tended to be per group optimal, when the groups do not correspond to the predicted label. As such they are already minimax optimal and the solutions found by minimax fairness methods are indistinguishable from those found by empiric risk minimization. This still leaves the case where groups include the label (for example, the groups may correspond to the product of gender and the variable we are trying to predict, such as sick or not sick). In this case, as convincingly shown by (Martinez et al., 2020), the solutions found do not correspond to ERM.

Here, we compare OxonFair against minimax fairness. To do this, we define a new performance measure corresponding to the lowest accuracy over the positive or negative labelled datapoints.

$$\text{min accuracy} = \min\left(\frac{TP}{TP+FP}, \frac{TN}{FN+TN}\right) \tag{4}$$

Martinez et al (Martinez et al., 2020) argued that we should seek a Pareto optimal solution that has the highest possible overall accuracy, subject to the requirement it maximizes the lowest per group accuracy. We can do this in OxonFair by calling

Table 5: The fairness measures in the review of (Verma & Rubin, 2018). All 9 group metrics that concern the decisions made by a classifier are supported by OxonFair.

Vema and Rudin(Verma & Rubin, 2018) Metrics	OxonFair name	Fairlearn
Group fairness or statistical parity	`demographic_parity`	Yes
Conditional statistical parity	`conditional_group_metrics.pos_pred_rate.diff`	No
Predictive parity	`predictive_parity`	No
False positive error rate balance	`false_pos_rate.diff`	Yes
False negative error rate balance	`false_neg_rate.diff`	Yes
Equalized odds	`equalized_odds`	Yes
Conditional use accuracy equality	`cond_use_accuracy`	No
Overall accuracy equality	`accuracy.diff`	No
Treatment equality	`treatment.diff`	No
Test-fairness or calibration	Not decision based	
Well calibration	Not decision based	
Balance for positive class	Not decision based	
Balance for negative class	Not decision based	
Causal discrimination	Individual fairness	
Fairness through unawareness	Individual fairness	
Fairness through awareness	Individual fairness	
No unresolved discrimination	Individual fairness	
No proxy discrimination	Individual fairness	
Fair inference	Individual fairness	

Table 6: The post-training fairness measures in the review of (Hardt et al., 2021). All measures are supported by OxonFair.

Post-training Metrics (Hardt et al., 2021)	OxonFair name	Fairlearn
Diff. in pos. proportions in predicted labels	`demographic_parity`	Yes
Disparate Impact	`disparate_impact`	No
Difference in Conditional Acceptance	`cond_accept.diff`	No
Difference in Conditional Rejection	`cond_reject.diff`	No
Accuracy Difference	`accuracy.diff`	No
Recall Difference	`recall.diff`	Yes
Difference In Acceptance Rates	`acceptance_rate.diff`	No
Difference in Rejection Rates	`rejection_rate.diff`	No
Treatment Equality	`treatment_equality`	No
Conditional Demographic Disparity	`conditional_group_metrics.pos_pred_rate.diff`	No

Table 7: Enforcing fairness for all definitions in (Hardt et al., 2021) on COMPAS with inferred attributes. We enforce the fairness definitions with respect to three racial groups, African American, Caucasian, and Other – consisting of all other labelled ethnicities. There are a total 350 individuals labelled 'Other' in the test set, making most metrics of fairness unstable and difficult to enforce. Nonetheless, we improve on all metrics. For all metrics except disparate impact, we enforce that the score on train is below 2.5% and for disparate impact we enforce that the score on train is above 97.5%. XGBoost is used as the base classifier, and the dataset is split into 70% train and 30% test.

	Measure (original)	Measure (updated)	Accuracy (original)	Accuracy (updated)
Demographic Parity	0.148706	0.097142	0.661345	0.620588
Disparate Impact	0.668305	0.740940	0.661345	0.605042
Difference in Conditional Acceptance Rate	0.231862	0.151159	0.661345	0.642857
Difference in Conditional Rejectance Rate	0.048625	0.025138	0.661345	0.655882
Difference in Accuracy	0.013172	0.006351	0.661345	0.665546
Difference in Recall	0.151210	0.105154	0.661345	0.612185
Difference in Acceptance Rate	0.070072	0.066591	0.661345	0.662605
Difference in Specificity	0.097490	0.064139	0.661345	0.660504
Difference in Rejection Rate	0.050085	0.050215	0.661345	0.661345
Treatment Equality	0.201717	0.105115	0.661345	0.660924
Conditional Demographic Parity	0.673950	0.626471	0.150927	0.073203

`fpredictor.fit(gm.min_accuracy.min, gm.accuracy, 0)` Here `min_accuracy.min` corresponds to the lowest min accuracy of any group. We use accuracy > 0 as the constraint, as we do not want an active constraint from preventing us from finding the element of the Pareto frontier (see (Martinez et al., 2020) for frontier details) with the highest minimum accuracy. Note that the groups used by OxonFair with this loss correspond to the true groups, such as ethnicity or gender, while the groups used by minimax fairness are the product of these groups with the target labels. Existing methods for minimax fairness optimize the same loss and have indistinguishable accuracy, only differing in the speed of convergence(Abernethy et al., 2022). As such, in table 8 we only report results for a variant of (Abernethy et al., 2022). Similarly, in computer vision (Zietlow et al., 2022), optimized the same objective by iteratively generating synthetic data for the worst performing group, where groups were defined as the product of ground-truth labels, and sex. We compare against them in table 14.

XGBoost: adult (sex)	Min Accuracy	Overall Accuracy
ERM Training	70.3%	90.9%
Minimax Training	85.2%	88.9%
ERM Validation	58.8%	86.9%
Minimax Validation	76.2%	83.9%
OxonFair Validation	79.1%	84.4%
ERM Test	59.6%	86.6%
Minimax Test	77.9%	84.1%
OxonFair Test	80.5%	84.6%

Table 8: Results for XGBoost: Adult (sex)

C.2. Utility Optimization

OxonFair supports the utility-based approach of Bakalar et al. (2021), whereby different thresholds can be selected per group to optimize a utility based objective. Utility functions can be defined in one line: `my_utility=gm.Utility([1, 1, 5, 0]`. Here, we consider a scenario where an ML system identifies issues that may require interventions. Every intervention has a cost of 1, regardless of if it was needed, but a missed intervention that was needed has a cost of 5. Not making an intervention when one was not needed has a cost of 0. The code `fpredictor.fit(my_utility, gm.recall, 0.5)` minimizes the utility subject to the requirement that the overall recall can not drop below 0.5.

C.3. Levelling up

One criticism of many methods of algorithmic fairness is that enforcing equality of recall rates (as in equal opportunity) or selection rates (as in demographic parity) will decrease the recall/selection rate for some groups while increasing it for others. This behavior is an artifact of trying to maximize accuracy (Mittelstadt et al., 2023) and occurs despite fairness methods altering the overall selection rate (Goethals et al., 2024a). As an alternative, OxonFair supports **levelling up** where harms are reduced to, at most, a given level per group (Mittelstadt et al., 2023). For example, if we believe that black patients are being disproportionately harmed by a high number of false negatives in cancer detection (i.e., low recall), instead of enforcing that these properties be equalized across groups, we can instead require that every group of patients has, at least, a minimum recall score. Depending on the use case, similar constraints can be imposed in with respect to per-group minimal selection rates, or minimal precision. These constraints can be enforced by a single call, for example, enforcing that the precision is above 70% while otherwise maximizing accuracy can be enforced by calling: `.fit(gm.accuracy, gm.precision.min, 0.7)`.

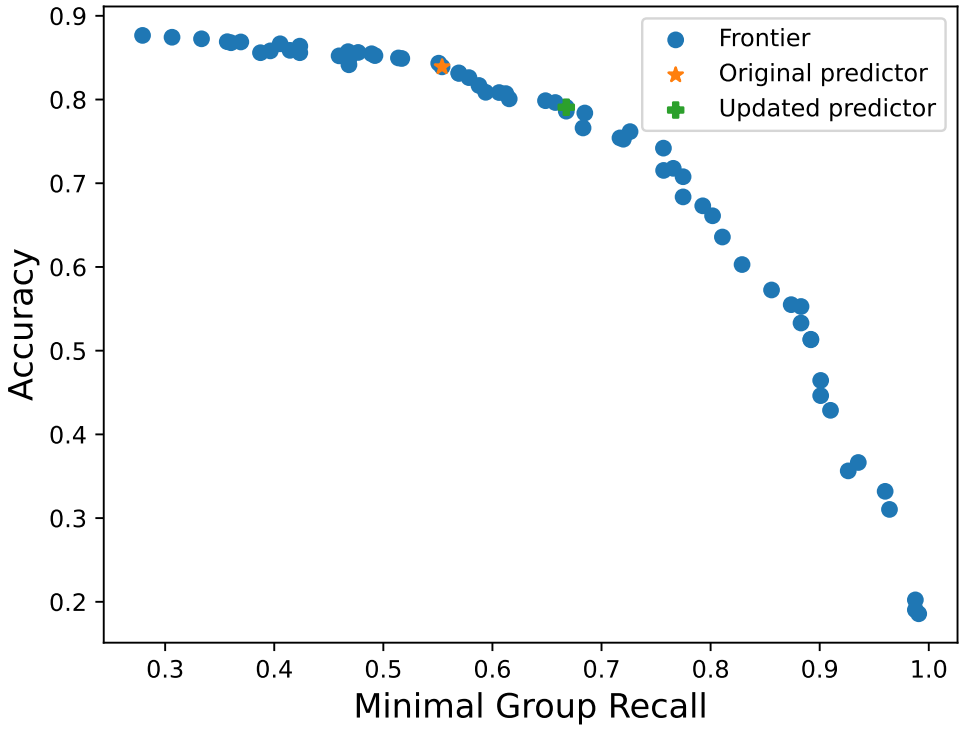


Figure 5: Levelling-up with OxonFair by imposing a minimum group recall of 0.7 on the Fitzpatrick-17k (Groh et al., 2021) validation set - `fpredictor.fit(gm.accuracy, gm.recall.min, 0.7)`.

C.4. Conditional Metrics

A key challenge of using fairness in practice is that often some sources of bias are known, and the practitioner is expected to determine if additional biases exist and to correct for them. For example, someone's salary affects which loans they are eligible for, but salary has a distinctly different distribution for different ethnicities and genders. (Chiappa, 2019). Identifying and correcting fairness here rapidly becomes challenging, when considering the intersection of attributes, many small groups arise and purely by chance some form of unfairness may be observed (Kearns et al., 2018; Wachter et al., 2021)suggested the use of a technique from descriptive statistics that (Freedman et al., 2007) had previously applied to the problem of schools admissions at Berkley (Bickel et al., 1975). In this famous example, every school in Berkley showed little gender bias, but due to different genders applying at different rates to different schools, and the schools themselves having substantially different acceptance rate, a strong overall gender bias was apparent.

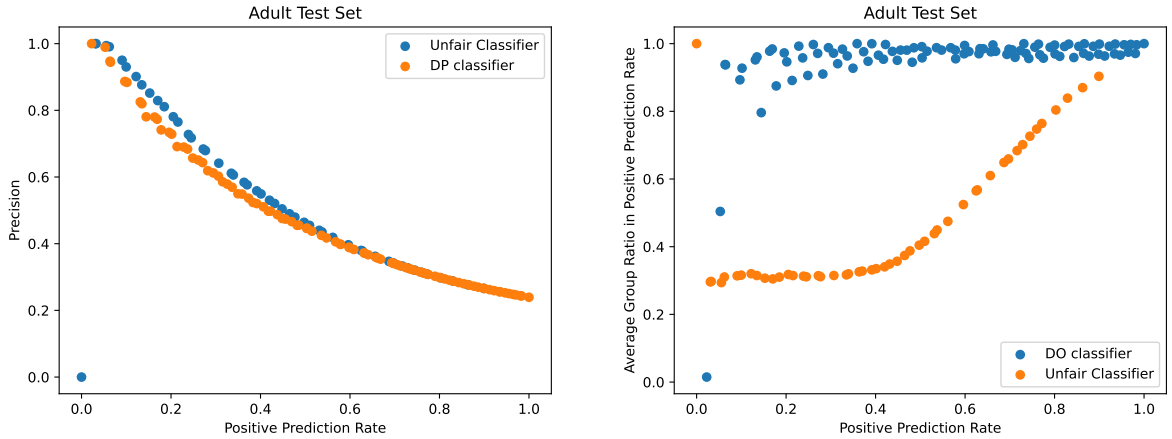


Figure 6: Solutions found when enforcing demographic parity with varying rate constraints. See appendix C.5. **Left:** the change in precision as we enforce demographic parity. Note that we report precision as it is more informative than accuracy for low selection rates. **Right:** The ratio between selection rates (i.e. disparate impact) for different groups. We report the ratio rather than the difference, as the difference tends to zero as the selection rate also tends to zero. However, as the right figure shows, this ratio becomes unstable as the rate tends to zero.

(Freedman et al., 2007) observed that you could correct for this bias by computing the per school selection-rate, and then taking a weighted average, where the weights are given by the total number of people applying to the school. The resulting selection rates are equivalent to a weighted selection-rate over the whole population, where the weight w_i for an individual i in a particular group and applying to a particular school is $w_i = \frac{\#\text{individuals in school}}{\#\text{individuals in group and school}}$. To enforce this form of conditional demographic parity in OxonFair, we simply replace the sum of true positives etc. in Section 3, with the weighted sum. We support a range of related fairness metrics, including conditional difference in accuracy; and conditional equal opportunity (note that for equal opportunity we replace the numbers used to compute w_i with the same counts but only taking into account those that have positive ground-truth labels). As such metrics can level down (Appendix C.3), we also support conditional minimum selection rates, and conditional minimum recall.

C.5. Fairness under constrained capacity

When deploying fairness in practice, we may be capacity limited. For example, as in fig. 5 we may use the output of a classifier for detecting cancer to schedule follow-up appointments. In such a case, you might wish that the recall is high for each demographic group, but be constrained by the number of available appointments. Calling `.fit(gm.recall.min, gm.pos_pred_rate, 0.4, greater_is_better_const=False)` will maximize the recall on the worst-off group subject to a requirement that no more than 40% of cases are scheduled follow-up appointments.

In general, maximizing the group minimum of any measure that is monotone with respect to the selection rate, while enforcing a hard limit on the selection rate will enforce equality with respect to that measure (e.g. optimizing `gm.recall.min` will result in equal recall a.k.a. equal opportunity, while maximizing `gm.pos_pred_rate.min` will result in demographic parity), while also enforcing the selection rate constraints. See (Goethals et al., 2024b) for proof and a discussion of the issues arising, and (Kwegyir-Aggrey et al., 2023) for an alternate approach.

As such, calling `.fit(gm.recall.min, gm.pos_pred_rate, k, greater_is_better_const = False)` will enforce equal opportunity at $k\%$ selection rate, and `.fit(gm.pos_pred_rate.min, gm.pos_pred_rate, 0.4, greater_is_better_const = False)` will enforce demographic parity at $k\%$ selection rate.

C.6. Directional Bias Amplification Metric Derivation for OxonFair

We also support a variant of Bias Amplification, as defined by Wang et al. (Wang & Russakovsky, 2021).

Closely following the notation of Wang et al. (Wang & Russakovsky, 2021), let $\mathcal{A}$ be the set of protected demographic groups: for example, $\mathcal{A} = \{\text{male, female}\}$. A_a for $a \in \mathcal{A}$ is the binary random variable corresponding to the presence of the group a; thus $P(A_{\texttt{woman}} = 1)$ can be empirically estimated as the fraction of images in the dataset containing women. Let T_t with $t \in \mathcal{T}$ similarly correspond to binary target tasks. Let $\hat{A}_a$ and $\hat{T}_t$ denote model predictions for the protected group a and the target task t, respectively.

$$
\begin{aligned}
\text{BiasAmp}_{\rightarrow} &= \frac{1}{|\mathcal{A}||\mathcal{T}|} \sum_{a\in\mathcal{A},t\in\mathcal{T}} y_{at}\Delta_{at} + (1-y_{at})(-\Delta_{at}) \\
y_{at} &= \mathbb{1}\left[P(A_a = 1, T_t = 1) > P(A_a = 1)P(T_t = 1)\right] \\
\Delta_{at} &= \begin{cases} P(\hat{T}_t = 1|A_a = 1) - P(T_t = 1|A_a = 1) \\ \text{if measuring } Attribute \rightarrow Task\ Bias \\ P(\hat{A}_a = 1|T_t = 1) - P(A_a = 1|T_t = 1) \\ \text{if measuring } Task \rightarrow Attribute\ Bias \end{cases}
\end{aligned} \tag{5}
$$

Of which, the Attribute $\rightarrow$ Task Bias is relevant here.

Each component can be written as a function of the global True Positives, False Positives etc., and the per group True Positives, and as such it can be optimized by our framework, albeit, not by using a standard group metrics. However, this metric is gamable, and consistently underestimating labels in groups where they're over-represented and vice versa would be optimal, but undesirable behavior that leads to a negative score.

Instead, we consider the absolute BiasAmp:

$$
\begin{aligned}
|\text{BiasAmp}|_{\rightarrow} &= \frac{1}{|\mathcal{A}||\mathcal{T}|} \sum_{a\in\mathcal{A},t\in\mathcal{T}} |y_{at}\Delta_{at} + (1-y_{at})(-\Delta_{at})| \\
&= \frac{1}{|\mathcal{A}||\mathcal{T}|} \sum_{a\in\mathcal{A},t\in\mathcal{T}} |\Delta_{at}|
\end{aligned} \tag{6}
$$

We can decompose $|\Delta_{at}|$ into the appropriate form for a `GroupMetric` (see appendix B) as follows:

$$\Delta_{at} = P(\hat{T}_t = 1|A_a = 1) - P(T_t = 1|A_a = 1) \tag{7}$$

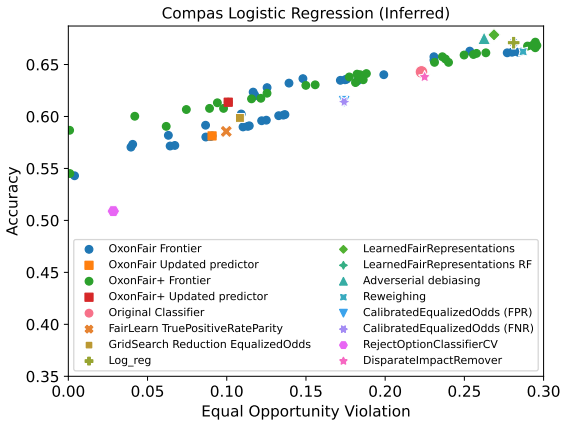
$$\Delta_{at} = \frac{TP + FN}{TP + TN + FP + FN} - \frac{TP + FP}{TP + TN + FP + FN} \tag{8}$$

$$|\Delta_{at}| = \left|\frac{FN - FP}{TP + TN + FP + FN}\right| \tag{9}$$

This will give a per group estimate of the absolute bias amplification, and calling its `.average` method will give the absolute bias amplification over all groups.

D. Tablular Data Experiments

For tabular data, we compare with all group fairness methods offered by AIF360, and the reductions approach of Fairlearn. OxonFair is compatible with any learner with an implementation of the method `predict_proba` consistent with scikit-learn (Pedregosa et al., 2011) including AutoGluon (Erickson et al., 2020) and XGBoost (Chen & Guestrin, 2016). A comparison with Fairlearn and the group methods from AIF360 on the adult dataset can be seen in figures 7 and 4 using random forests. This follows the setup of (Bellamy et al., 2018): we enforce fairness with respect to race and binarize the attribute to white vs everyone else (this is required to compare with AIF360), 50% train data, 20% validation, and 30% test, and a minimum leaf size of 20. With this large leaf size, all errors on train, validation, and test are broadly comparable, but our approach of directly optimizing an objective and a fairness measure leads us to outperform others.



#Groups	Accuracy	Dem. Par.	Time
OxonFair 5	86.6	2.5%	0.7+55.7s
Fairlearn 5	86.5	3.8%	33.2s
OxonFair 4	86.5	1.2%	0.7+0.93s
Fairlearn 4	86.3	2.0%	33.3s
OxonFair 3	86.6	1.5%	0.7+0.078s
Fairlearn 3	86.6	1.0%	25.4s
OxonFair 2	86.7	1.1%	0.7+0.05s
Fairlearn 2	87.0	2.0%	19.4s

Figure 7: **Left:** Results on Compas. **Right:** Runtime Comparison for Fairlearn Reductions and OxonFair on Adult using a Macbook M2. To alter the groups, we iteratively merge the smallest racial group with 'Other', reducing the search space. For both methods, we enforced demographic parity over a train set consisting of 70% of the data. Despite the exponential complexity of our approach, we remain faster until we reach 5 groups. The 0.7+ indicates the seconds to train XGBoost.

Figure 7 *top right* shows the importance of being able to use a validation set to balance errors. Using sklearn's default parameters we overfit to adult, and as the classifier is perfect on the training set, all fairness metrics that match error rates are trivially satisfied (Wachter et al., 2020; Zietlow et al., 2022). The same behavior can be observed using XGBoost on the medical dataset (Golovenkin et al., 2020) when enforcing equal opportunity with respect to sex[5]. In general, tabular methods need not overfit, and tuning parameters carefully can allow users to get relatively good performance while maintaining error rates between training and test.

Figure 7 left shows Equal Opportunity on the COMPAS dataset. To show that OxonFair can also work in low-data regimes where we have insufficient data for validation, we enforce fairness on the training set. As before, we binarize race to allow the use of AIF360. We drop race from the training data, and use inferred protected attributes to enforce fairness. Here OxonFair generates a frontier that is comparable or better than results from existing toolkits, and OxonFair+ (see section A), further improves on these results. See Figure 7 right for a comparison with Fairlearn varying the groups.

E. Computer Vision Experiments

Table 9: Hyperparameter details for the CelebA experiment.

Hyperparameter	**Value/Range**
Learning Rate	0.0001
Batch Size	32
Dropout Rate	0.5
Backbone	Resnet-50
Weight Decay	0
Optimizer	Adam (Kingma & Ba, 2014)
Epochs	20

E.1. Methods

We extensively used the codebase of Wang et. al (Wang et al., 2020) to conduct comparative experiments[6].

[5]This dataset is carefully curated and balanced. To induce unfairness we altered the sampling and dropped half the people recorded as male and that did not have medical complications across the entire dataset.

[6]https://github.com/princetonvisualai/DomainBiasMitigation

Table 10: CelebA Attribute-level information from Ranaswamy et al. (Ramaswamy et al., 2021). The columns are target attribute name, percentage of positive samples, skew. For example, `Earrings` has a skew of 0.97 towards $g=-1$, that is, 97% of positive `Earrings` samples have gender expression label $g=-1$ (`Female`)

Attribute type	Attribute statistics	
Inconsistently labeled	Positive	Skew
BigLips	24.1%	0.73 $g=-1$
BigNose	23.6%	0.75 $g=1$
OvalFace	28.3%	0.68 $g=-1$
PaleSkin	4.3%	0.76 $g=-1$
StraightHair	20.9%	0.52 $g=-1$
WavyHair	31.9%	0.81 $g=-1$
Gender-dependent	Positive	Skew
ArchedBrows	26.6%	0.92 $g=-1$
Attractive	51.4%	0.77 $g=-1$
BushyBrows	14.4%	0.71 $g=1$
PointyNose	27.6%	0.75 $g=-1$
RecedingHair	8.0%	0.62 $g=1$
Young	77.9%	0.66 $g=-1$
Gender-independent	Positive	Skew
Bangs	15.2%	0.77 $g=-1$
BlackHair	23.9%	0.52 $g=1$
BlondHair	14.9%	0.94 $g=-1$
BrownHair	20.3%	0.69 $g=-1$
Chubby	5.8%	0.88 $g=1$
EyeBags	20.4%	0.71 $g=1$
Glasses	6.5%	0.80 $g=1$
GrayHair	4.2%	0.86 $g=1$
HighCheeks	45.2%	0.72 $g=-1$
MouthOpen	48.2%	0.63 $g=-1$
NarrowEyes	11.6%	0.56 $g=-1$
Smiling	48.0%	0.65 $g=-1$
Earrings	18.7%	0.97 $g=-1$
WearingHat	4.9%	0.70 $g=1$
Average	24.1%	0.73

- **Empirical Risk Minimization (ERM) (Vapnik, 1999):** Acts as a baseline in our experiments where the goal is to minimize the average error across the dataset without explicitly considering the sensitive attributes.
- **Adversarial Training with Uniform Confusion (Alvi et al., 2018)**: The goal is to learn an embedding that maximizes accuracy whilst minimizing any classifier's ability to recognize the protected class. The uniform confusion loss from Alvi et al. (Alvi et al., 2018) is used following the implementation of (Wang et al., 2020).
- **Domain-Discriminative Training (Wang et al., 2020)**: Domain information is explicitly encoded and then the correlation between domains and class labels is removed during inference.
- **Domain-Independent Training (Wang et al., 2020)**: Trains a different classifier for each attribute where the classifiers do not see examples from other domains.
- **OxonFair + Multi-Head (Lohaus et al., 2022)**: Described in Section 4.2. $N-1$ heads are trained to minimize the logistic loss over the target variable, where N is the total number of attributes. A separate head minimizes the squared loss over the protected attribute. Fairness is enforced on validation data with two separate optimization criteria. **OxonFair-DEO** calls `fpredictor.fit(gm.accuracy, gm.equal_opportunity, 0.01)` to enforce Equal Opportunity. **OxonFair-MGA** calls `fpredictor.fit(gm.accuracy, gm.min.accuracy.min, 0)`.

Table 11: We report mean scores over the 14 gender independent labels (Ramaswamy et al., 2021) of CelebA. Single task methods and FairMixup scores in the second and third blocks are from Zietlow et al. (Zietlow et al., 2022). ERM indicates the baseline architecture run without fairness. OxonFair (optimizing for accuracy and DEO), has better accuracy and DEO scores than any other fair method.

	ERM multitask	Domain Disc.	Domain Indep.	Uniconf. Adv.	OxonFair DEO	ERM single task	Debiasing GAN	Regularized	g-SMOTE Adaptive	g-SMOTE	ERM	FairMixup
Acc.	**93.07**	92.96	92.63	92.71	92.75	92.47	92.12	91.05	92.56	**92.64**	**92.74**	88.46
DEO	16.47	14.61	7.78	19.63	**3.21**	12.54	9.11	**3.77**	14.28	15.11	7.97	**3.58**

Table 12: Comparing accuracy of fairness methods while varying minimum recall level thresholds, δ.

CelebA - 26 Attributes	$\delta = 0.50$	$\delta = 0.75$	$\delta = 0.85$	$\delta = 0.90$	$\delta = 0.95$
Baseline (ERM)	89.0	84.5	80.6	77.6	72.7
Adversarial	87.8	82.4	78.2	75.2	69.3
Domain-Dependent	82.3	76.8	72.4	68.6	62.2
Domain-Independent	89.2	86.2	82.9	79.8	74.4
OxonFair	**89.9**	**87.3**	**84.4**	**81.8**	**76.9**

Table 13: Extended Version of Table 1. Performance Comparison of Different Algorithmic Fairness Methods on the CelebA Test Set. Results monitor the mean Accuracy, Difference in Equal Opportunity (DEO) and the Minimum Group Minimum Label Accuracy across the attributes.

	ERM	Uniconf. Adv (Alvi et al., 2018)	Domain Disc. (Wang et al., 2020)	Domain Ind. (Wang et al., 2020)	OxonFair DEO	OxonFair MGA
Gender-Independent Attributes						
Acc.	**93.1**	92.7	93.0	92.6	92.8	90.9
Min grp. min acc.	64.1	72.3	76.5	71.2	72.3	**85.8**
DEO	16.5	19.6	14.6	7.78	**3.21**	3.52
Gender-Dependent Attributes						
Acc.	**86.7**	86.1	86.6	85.6	85.8	82.3
Min grp. min acc.	43.4	53.7	59.6	53.8	52.5	**78.5**
DEO	26.4	25.0	21.9	6.50	**3.92**	3.96
Inconsistently Labelled Attributes						
Acc.	83.0	82.5	**83.1**	82.3	82.1	79.2
Min grp. min acc.	36.1	43.0	50.2	42.7	44.3	**69.5**
DEO	21.9	29.1	25.3	17.2	**2.36**	4.86

Table 14: Performance comparison of Baseline, Adaptive g-SMOTE, g-SMOTE, OxonFair-DEO, and OxonFair-MGA on the training set. Reported are the means over the 32 labels from Zietlow et al. (2022).

4 Protected Groups		**ERM**	**Adaptive g-SMOTE**	**g-SMOTE**	**OxonFair-DEO**	**OxonFair-MGA**
Full Training Set	**Acc.**	**90.49**	85.77	87.27	89.21	86.18
	Min. grp. acc.	61.74	68.06	61.84	54.20	**78.48**
	DEO	24.70	12.27	21.91	**3.93**	5.58

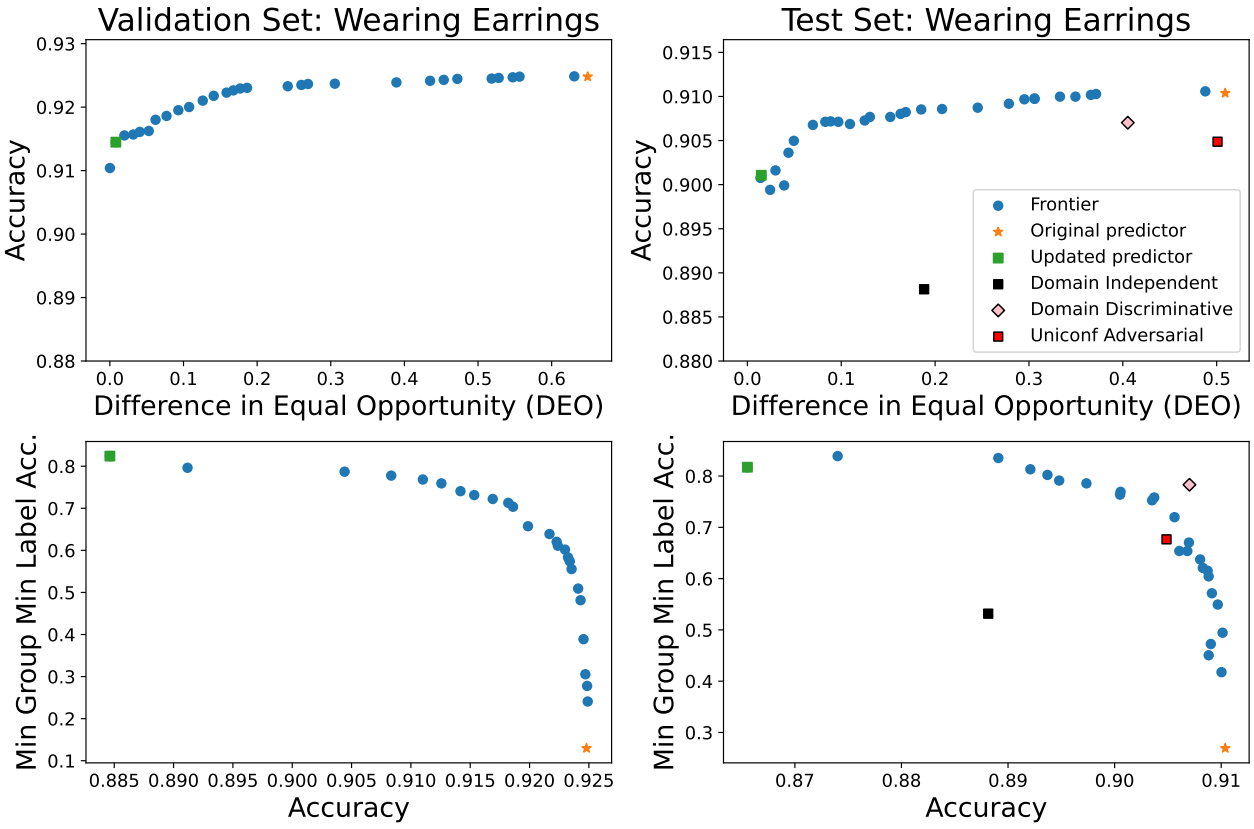


Figure 8: A comparison of the Pareto frontier on validation and test data when enforcing two fairness measures (DEO and Min Group Min Label Acc) for the Wearing Earrings attribute in CelebA whilst monitoring model accuracy.

	Gender	Country	Ethnicity	Age
English	41200/7008/6927	44487/7744/7639	40731/6954/6845	39003/6628/6608
Polish	11782/1461/1446	2218/489/471	8567/1199/1235	8610/1199/1235
Spanish	2240/407/410	2299/436/439	2244/407/410	2249/407/410
Portuguese	1408/150/163	1105/198/197	1377/150/163	1389/150/163
Italian	2730/350/369	3769/514/516	2706/348/368	2676/349/368

Table 15: Multilingual Twitter corpus train/val/test statistics.

	original DEO	updated DEO	original Acc.	updated Acc.
English	5.13	3.19	84.0	84.2
Polish	21.4	10.1	89.6	85.8
Spanish	9.39	1.64	69.8	67.3
Portuguese	17.3	1.29	60.7	52.1
Italian	7.77	0.42	75.6	77.5

Figure 9: Multilingual Experiment.

	original DEO	updated DEO	original Acc.	updated Acc.
Gender	21.4	8.45	89.6	88.5
Country	10.2	8.32	81.4	82.2
Ethnicity	8.56	4.92	83.1	82.7
Age	12.5	6.02	82.1	80.5

Figure 10: Demographic Experiments.

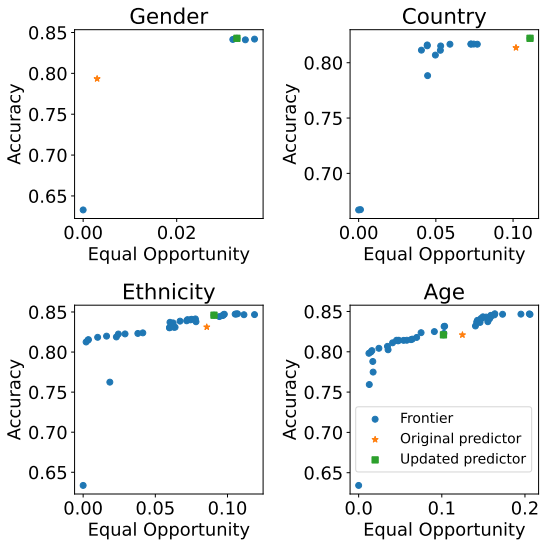


Figure 11: Demographics frontier plot.

F. NLP Experiments

F.1. Experimental Details

We employ a BERT-based model architecture (Devlin et al., 2018), augmented with an additional head to simultaneously predict demographic factors (see Section 4.2. During training, we utilize the standard cross-entropy loss for the primary prediction task and a mean squared error loss for the demographic predictions, aggregating these to compute the overall loss. We ensure data consistency by excluding entries with missing demographic information. To facilitate easy comparison with different models, we select the Polish language for the multilingual Twitter corpus, noted for its high DEO score, to demonstrate how various models can reduce this score. We also conducted our experiment on the Jigsaw data. Unlike the multilingual Twitter corpus, the Jigsaw religion dataset contains three groups: Christian, Muslim, and others. The entire model, including the BERT backbone, is fine-tuned for 10 epochs using an initial learning rate of 2×10^{-5}, following the original BERT training setup. All experiments are conducted on an NVIDIA A100 80GB GPU.

F.2. Methods

We compare OxonFair with the following approaches. **Base** reports results of the standard BERT model(Devlin et al., 2018). **CDA** (Counterfactual Data Augmentation) (Zmigrod et al., 2019; Dinan et al., 2019; Webster et al., 2020; Barikeri et al., 2021; Meade et al., 2021) rebalances a corpus by swapping bias attribute words (e.g., he/she) in a dataset based on a given dictionary. **DP** (Demographic Parity) uses regularization(Zafar et al., 2017; Han et al., 2024) to enforce DP. **EO** (Equal Opportunity(Hardt et al., 2016)) uses the regularization of (Zafar et al., 2017; Han et al., 2024) to enforce EO. **Dropout** (Webster et al., 2020; Meade et al., 2021) is used as a regularization technique (Srivastava et al., 2014) for bias mitigation and improving small group generalization. **Rebalance** (Feldman et al., 2015; Li & Vasconcelos, 2019) method resamples the minor groups to the same sample size as other groups to mitigate bias.

F.3. Hate Speech Detection Task

We follow the methodology outlined in (Huang et al., 2020) to conduct the hate speech detection task using our tool. Variables such as age and country in the multilingual Twitter corpus are binarized using the same method as described in (Huang et al., 2020). The data splits for training, development, and testing are shown in Table 15.

Multilingual Experiment. To demonstrate the capability of our proposed tool in handling multilingual scenarios, we conduct experiments across five languages: English, Polish, Spanish, Portuguese, and Italian and the results are shown in Figure 9. Observations from the results indicate that: 1) Our model improves equal opportunity performance with minimal sacrifice to the main task performance. 2) The datasets in Polish and Portuguese show higher equal opportunity, indicating more severe bias compared to other languages, yet our proposed method effectively enhances performance in these conditions.

Demographic Experiments. To demonstrate our tool's ability to address various demographic factors in text, we conducted experiments focusing on age, country, gender, and ethnicity, with results detailed in Figure 10 and Figure 11. The outcomes reveal that our tool effectively improves equal opportunities across all demographic factors, underscoring its capability to handle general debiasing scenarios.

	Christian	Other	Muslim
Train	22845/1892	3783/554	9527/2390
Valid	5681/470	946/148	2425/578
Test	2944/251	604/78	1119/319

Table 16: Jigsaw religion data.

	Black	Asian
Train	6718/2811	2187/246
Valid	1684/698	547/61
Test	841/364	284/25

Table 17: Jigsaw race data.

F.4. Toxicity Classification Task

We also evaluate toxicity classification using the Jigsaw toxic comment dataset (Jigsaw, 2018), which has been transformed into a Kaggle challenge. To demonstrate the ability of OxonFair to handle multiple protected groups, we consider religion as the protected attribute and evaluate performance across three groups: Christian, Muslim, and Other. Owing to the limitted dataset size, all samples labelled as a religion that was neither Christian nor Muslim were merged into Other and unlabeled samples were discarded. The statistics for this dataset are shown in Table 16, where each cell displays the count of negative and positive samples, respectively. The experimental results are discussed in the main paper.

For the Jigsaw dataset, we follow the setup of (Chuang & Mroueh, 2021), selecting race as the protected attribute. We focus on the subset of comments identified as Black or Asian, as these two groups exhibit the largest gap in the probability of being associated with toxic comments. The data statistics are shown in Table 17 where each cell displays the count of negative and positive samples, respectively. The experimental results, presented in Table 18, demonstrate that our proposed tool outperforms all other models.

	F1 score	Balanced Accuracy	Accuracy	Equal Opportunity
Base	53.4	68.9	72.1	23.7
CDA (Zmigrod et al., 2019)	52.7	68.2	76.4	7.65
DP (Zafar et al., 2017)	47.4	64.6	72.6	4.35
EO (Hardt et al., 2016)	47.1	64.5	73.2	5.85
Dropout (Webster et al., 2020)	52.4	68.0	72.0	12.7
Rebalance (Feldman et al., 2015)	51.7	67.5	74.4	5.57
OxonFair (Accuracy)	37.5	60.8	77.7	2.10
OxonFair (F1)	52.8	68.5	69.2	11.9
OxonFair (Balanced Accuracy)	52.7	68.5	68.5	0.41

Table 18: Jigsaw dataset: Race (w groups: Black, Asian).

G. Comparison Table Information

In this section, we provide further details on the information from Figure 7. While all approaches have many fairness definitions that can be computed, very few can be enforced via bias mitigation. As a minimum, OxonFair supports enforcing the methods from tables 5 and 6 (eliminating duplicates give the number 14 in the table). In addition to this, it supports a wide range of metrics that aren't used in the literature, for example minimizing the difference in balanced accuracy, F1 or MCC between groups, e.g., by using `balanced_accuracy.diff` as a constraint. It also supports the definitions set out in Appendix C, including minimax notions; absolute bias amplification; and enforcing for minimum rates per group in recall, or precision, or sensitivity actively promoting *levelling-up* (Mittelstadt et al., 2023).

G.1. FairLearn Methods Support

Fairlearn provides an overview of the supported bias mitigation algorithms and supported fairness constraints in their documentation[7]. The number of performance and fairness objectives supported are dependent on the method.

Methods supported include ExponentiatedGradient and GridSearch that provide a wrapper around the reductions approach to fair classification of Agarwal et al. (Agarwal et al., 2018). Supported fairness definitions for classification are Demographic-Parity, Equalized Odds, True Positive Rate Parity, False Positive Rate Parity and Error Rate Parity. For postprocessing the ThresholdOptimizer approach of Hardt et al. (Hardt et al., 2016) is supported. The adversarial approach of (Zhang et al., 2018) is also supported and can enforce fairness based on Demographic Parity and Equalized Odds. The CorrelationRemover method provides preprocessing functionality to remove correlation between sensitive features and non-sensitive features through linear transformations. It should be emphasized that Fairlearn also provides an interface for defining custom Moments for fairness and objective optimization, however, as of the current version 0.10 no documentation or examples are provided for doing so.

G.2. AIF360 Methods Support

AIF360 provides support for a wide variety of methods[8][9] that enforce fairness, many of which overlap with Fairlearn. We consider group fairness approaches.

Preprocessing algorithms include DispirateImpactRemover (Feldman et al., 2015), LFR (Zemel et al., 2013), Optimized Pre-processing (Calmon et al., 2017), Reweighting (Kamiran & Calders, 2012) and FairAdapt (Plečko & Meinshausen, 2020). Inprocessing algorithms include AdversarialDebiasing (Zhang et al., 2018), PrejudiceRemover (Kamishima et al., 2012), Exponentiated GradientReduction and GridSearchReduction (Agarwal et al., 2018). Postprocessing approaches include CalibratedEqOddsPostprocessing (Pleiss et al., 2017), EqOddsPostprocessing (Hardt et al., 2016), RejectOptionClassification (Kamiran & Calders, 2012).

H. Bias Mitegation Stratagies

Pre-processing algorithms improve fairness by altering the dataset in an attempt to remove biases such as disparate impact (Feldman et al., 2015) before learning a model itself. Popular pre-processing approaches include simply re-weighting samples in the training data to enhance fairness (Kamiran & Calders, 2012), optimizing this process by learning probabilistic transformations (Calmon et al., 2017), or by generating synthetic data (Chakraborty et al., 2021; Ramaswamy et al., 2021; Zmigrod et al., 2019).

In-processing / In-training methods mitigate bias by adjusting the training procedure. Augmenting the loss with fair regularizers (Zafar et al., 2017; Lohaus et al., 2020) is common for logistic regression and neural networks. (Agarwal et al., 2018) iteratively alters the cost for different datapoints to enforce fairness on the train set. Approaches based on adversarial training typically learn an embedding that reduces an adversary's ability to recover protected groups whilst maximizing predictive performance (Zhang et al., 2018; Madras et al., 2018; Zhao et al., 2019; Kim et al., 2019). Other popular approaches include Disentanglement (Tartaglione et al., 2021; Sarhan et al., 2020), Domain Generalization (Sagawa et al., 2019; Cha et al., 2021; Foret et al., 2020), Domain-independence (Wang et al., 2020) and simple approaches such

[7] `https://FairLearn.org/main/user_guide/mitigation/index.html`
[8] `https://aif360.readthedocs.io/en/stable/modules/algorithms.html`
[9] `https://aif360.readthedocs.io/en/stable/modules/sklearn.html`

as up-sampling or reweighing minority groups during training. Notably, in the case of high-capacity models in medical computer-vision tasks, a recent benchmark paper by Zong et al. (Zong et al., 2023) discovered that state-of-the-art in-processing methods do not significantly improve outcomes over training without considering fairness at all. A comprehensive benchmark study of in-processing methods in domains outside of healthcare is provided by Han et al. (Han et al., 2024).

Post-processing methods aim to enforce fairness by using thresholds or randomization to adjust the predictions of a trained model based on the protected attributes (Hardt et al., 2016; Pleiss et al., 2017). Post-processing methods are typically *model-agnostic* and can be applied to any model that returns confidence scores.

I. Limitations and Broader Impact

Limitations: We have chosen to optimize as broad a set of formulations as possible. As a result, for certain metrics (particularly equalized odds (Hardt et al., 2016)) the solutions found are known to be suboptimal; and for others (Corbett-Davies et al., 2017) the exponential search is unneeded. Techniques targeting particular formulations may be needed to address this. A major driver of unfairness is a lack of data regarding particular groups. However, this very absence of data makes it hard for any toolkit to detect or rectify unfairness.

Broader Impact: We reiterate the findings of Balayn et al., who note that fairness toolkits can act as a double-edged sword (Balayn et al., 2023). Open source toolkits can enable wider adoption of the assessment and mitigation of bias and fairness related harms. However, if misused, these toolkits can create a flawed certification of algorithmic fairness, endangering false confidence in flawed methodologies (Lee & Singh, 2021; Watkins et al., 2022). We join growing calls in encouraging practitioners to be reflective in their use of fairness toolkits (Bakalar et al., 2021). Specifically, we urge practitioners to adopt a harms first approach to fairness and be reflective in their measurement and enforcement of fairness.

OxonFair is a tool for altering the decisions made by ML systems that are frequently trained on biased data. Care must be taken that fair ML is used as a final step after correcting for bias and errors in data collation, and not as a sticking plaster to mask problems (Balayn et al., 2023). Indeed, inappropriate uses of fairness can lock in biases present in training (Wachter et al., 2020). Under the hood, OxonFair performs a form of positive discrimination, where we alter scores in response to (perceived) protected characteristics to rectify particular inequalities. As such, there are many scenarios where its use may be inappropriate for legal or ethical reasons.