Towards Standardizing AI Bias Exploration

Emmanouil Krasanakis*, Symeon Papadopoulos

Centre for Research & Technology Hellas, 6th km Charilaou-Thermi, Thessaloniki, Greece, 57001

Abstract

Creating fair AI systems is a complex problem that involves the assessment of context-dependent bias concerns. Existing research and programming libraries express specific concerns as measures of bias that they aim to constrain or mitigate. In practice, one should explore a wide variety of (sometimes incompatible) measures before deciding which ones warrant corrective action, but their narrow scope means that most new situations can only be examined after devising new measures. In this work, we present a mathematical framework that distils literature measures of bias into building blocks, hereby facilitating new combinations to cover a wide range of fairness concerns, such as classification or recommendation differences across multiple multi-value sensitive attributes (e.g., many genders and races, and their intersections). We show how this framework generalizes existing concepts and present frequently used blocks. We provide an open-source implementation of our framework as a Python library, called *FairBench*, that facilitates systematic and extensible exploration of potential bias concerns.

Keywords

Measuring bias, Auditing tools, Algorithmic frameworks, Multidimensional bias

1. Introduction

Artificial Intelligence (AI) systems see widespread adoption across many applications that affect people's lives. Since they tend to pick up and exacerbate real-world biases or discrimination, as well as spurious correlations between predicted values and sensitive attributes (e.g., gender, race, age, financial status), making them fair is a subject of intensive research and regulatory efforts. These include quantification of bias concerns through appropriate measures so that unfair behavior can be detected and corrected. To this end, several measures of bias have been proposed in research papers and implemented as reusable components within programming libraries or toolkits (Section 2). Recognizing that bias and, more generally, fairness depends on the social context and the particular settings in which AI systems are deployed, each created measure is limited to assessing a different type of concern. In the end, research and development focuses on mitigating or constraining measures when those reveal fairness issues, but not on how a systematic exploration of many measures could be carried out to perform fairness audits.

Therefore, and despite the obvious value of presenting reusable algorithmic solutions to specific problems, there is a need for methods that critically examine real-world systems across a wide range of concerns and not just a few of them. The main barrier in pursuing such methods is that measures of bias are designed in a monolithic manner and rarely consider how they

AIMMES 2024 Workshop on AI bias: Measurements, Mitigation, Explanation Strategies | co-located with EU Fairness Cluster Conference 2024, Amsterdam, Netherlands

*Corresponding author.

maniospas@iti.gr (E. Krasanakis); papadop@iti.gr (S. Papadopoulos)

0000-0002-3947-222X (E. Krasanakis); 0000-0002-5441-7341 (S. Papadopoulos)

could be generalized or ported to different settings. For example, differential fairness [1] was only recently proposed as a means of generalizing disparate impact assessment to intersectional fairness, which recognizes the cumulative effect of sensitive attributes (e.g., multiple genders and races), despite disparate impact measures being around for decades [2].

In this work, we assist systematic exploration of fairness concerns by decomposing measures of bias into simple building blocks. These can be recombined to create new measures covering a wide range of contexts and concerns. For example, a block that aggregates classification bias in multidimensional settings (e.g., with multiple races and genders) can be combined with a block that another measure uses to assess recommendation bias between only two groups of people (e.g., only whites vs non-whites) to create a new measure that assesses multidimensional recommendation bias. We implement the proposed framework in a Python library, called *FairBench*, that standardizes how measures of bias are defined by combining interoperable blocks of each kind. The library's functional interface sets up a fixed representation of existing and new measures of bias, and can create bias reports while tracing prospective fairness issues to the raw quantities computed over predictive outcomes. Our contribution is threefold:

a) We present a mathematical framework that systematically combines building blocks to construct many existing and new measures of bias.
b) We express several building blocks of literature measures of bias within this framework.
c) We introduce the FairBench Python library that implements the above blocks and combination mechanisms to compute measures of bias in a wide range of computing environments.

This paper is structured as follows. After this section's introduction, Section 2 presents theoretical background and related work. Section 3 describes our proposed mathematical framework. Section 4 extracts several bias building blocks from the AI fairness literature. Section 5 introduces the programming interface provided by our FairBench Python library to explore bias. Finally, Section 6 summarizes our work and points to future directions.

2. Background and Related Work

2.1. Fair AI

The problem of creating fair AI systems is the subject has attracted attention as part of the broader theme of Trustworthy AI in worldwide regulation and ethical guidelines, like the EU's Assessment List for Trustworthy AI [3], and the NIST's AI Risk Management Framework [4]. Evaluating system fairness is a complex problem that depends on the context being examined and the systems being created. A part of answering this problem consists of mathematical or algorithmic exploration that enables automated system assessment and oversight with practices like TrustAIOps [5], which monitor the evolution of system bias in the deployment context.

Mathematical definitions of fairness are often categorized into the following types [6, 7, 8, 9]:

i) *Group fairness* focuses on equal treatment between population groups or subgroups. This is the subject of most research and covered extensively in the next subsection.
ii) *Individual fairness* [10] focuses on the fair treatment of individuals, for example by obtaining similar predictions for those with similar features. This can be modelled as group fairness too, by considering every person to belong to their own group.

iii) *Counterfactual fairness* [11, 12] learns causal models of predictive mechanisms that account for discrimination and then makes predictions in a would-be fair reality. Although originally coined as a variation of individual fairness, recent understanding [13] also suggests that counterfactual fairness is a kind of group fairness.

Making (e.g., training) AI to be fair typically involves measures of bias that quantify numerical deviation from exact definitions of fairness. Measures are either minimized or subjected to constraints [14], but identifying which are important is not only a matter of applying scientific principles. In particular, it is mathematically impossible to simultaneously satisfy all conceivable definitions of fairness [15, 16], which means that systems should only address the fairness concerns that matter to humans (e.g., stakeholders or policymakers) in the particular situation. There are also trade-offs between satisfying fairness and maintaining predictive performance. In this work we propose that many types of bias should be monitored simultaneously to reveal concerning trends that require further context-dependent human evaluation.

In practice, AI systems employ fairness libraries or toolkits to compute popular measures of bias and either constrain predictive tasks to achieve the accepted measure values or carry out trade-offs of the latter with predictive performance. Some popular software projects for fair AI are AIF360 [17], FairLearn [18], What if [19], and Aequitas [20]. Each of these focuses on supporting different computational backends and data structures. AIF360 and FairLearn are programming libraries and provide bias mitigation algorithms. What If and Aequitas focus on assessment of fairness, and specialize in assessing counterfactual and group fairness respectively. All libraries and toolkits implement ad-hoc measures of bias that capture fairness concerns well-studied in the literature, where the latter typically account for very restricted types of evaluation that do not model complex concerns.

2.2. Measures of bias

We now present common measures of bias, starting from ones that quantify group fairness concerns for classifiers [21]. We organize people with the same sensitive attribute values into population groups. All presented measures assume values in the range $[0, 1]$ and, when necessary, we show the complements of measures of fairness with respect to 1 to let them assess bias instead, i.e., a value of 0 indicates perfect fairness. For example, below we use $1 - \text{prule}$ as a measure of bias, instead of prule, which is a measure of fairness.

Measures of classification bias. The measures of $1 - \text{prule}$ [2] and Calders-Verwer disparity cv [22] quantify disparate impact concerns, i.e., prediction rate inequality between two groups of samples $\mathcal{S}$ and $\mathcal{S}' = \mathcal{S}_{all} \setminus \mathcal{S}$, where the latter complements the former within the population $\mathcal{S}_{all}$. This models one binary sensitive attribute indicating membership to the group. Mathematically:

$$\text{prule} = \min\left\{\frac{\text{P}(c(x)=1|x\in\mathcal{S})}{\text{P}(c(x)=1|x\in\mathcal{S}')}, \frac{\text{P}(c(x)=1|x\in\mathcal{S}')}{\text{P}(c(x)=1|x\in\mathcal{S})}\right\}, \quad \text{cv} = |\,\text{P}(c(x)=1|x\in\mathcal{S}) - \text{P}(c(x)=1|x\in\mathcal{S}')|$$

where $\text{P}(\cdot|\cdot)$ denotes conditional probability and $c(x) \in \{0, 1\}$ is the binary outcome of classifying data sample x. Disparate impact is eliminated when $1 - \text{prule} = \text{cv} = 0$. Another concern is disparate mistreatment [23], which captures misclassification differences between groups per:

$$|\Delta m| = |m(\mathcal{S}) - m(\mathcal{S}')|$$

where $m(\cdot)$ denotes a misclassification measure over groups of samples, such as their false positive rate (fpr) or their false negative rate (fnr). The same formula can express cv or accommodate error measures for other predictive tasks, and therefore constitutes a building block of generalized fairness evaluation frameworks [24] (also see Subsection 4.3). An example of difference-based bias for probabilistic measures of predictive performance is equalized opportunity difference [25, 26]:

$$|\Delta eo| = |\,\mathrm{P}(c(x) = 1 | x \in \mathcal{S}, Y(x) = y) - \mathrm{P}(c(x) = 1 | x \in \mathcal{S}', Y(x) = y)|$$

where $y \in \{0, 1\}$ and $Y(x)$ is the true test/validation data label corresponding to sample x.

Measures of scoring and recommendation bias. The same underlying principles can be ported to other predictive tasks [5, 14] such as recommendation and scoring. In recommendation, the base measure $m(\cdot)$ in $|\Delta m|$ may be replaced by some form of exposure of items to group members or the fraction of top-k recommendations that are members of the group. In scoring tasks, $m(\cdot)$ may represent the fraction of score mass concentrated on groups, which can be compared to a desired fraction or between groups to satisfy concerns similar to disparate impact [27]. Aggregate statistics, like the area under the roc (auc) or score means, may also be compared between groups [28]. A definition that we use later for receiver operating characteristic (roc) curves of groups $\mathcal{S}_i$ accounts for all pairs of false positive rate (fpr) and true positive rate (tpr) values obtained for the groups over different thresholds θ of whether scores are interpreted as positive predictions or not (we use the maplet arrow to express rocs as maps from fpr to tpr):

$$\mathrm{roc}(\mathcal{S}_i) = \{\mathrm{fpr}(\mathcal{S}_i, \theta) \mapsto \mathrm{tpr}(\mathcal{S}_i, \theta) \,:\, \theta \in (-\infty, \infty)\} \tag{1}$$

More fine-grained approaches summarize the differences between curves or distributions (a mathematical expression for this appears in Subsection 4.3). For example, viable measures of bias are the absolute betweenness area between roc curves (abroca) [29] and the Kullback-Leibler divergence between distributions of system outcomes for each group [30].

2.3. Frameworks to assess multidimensional bias

Fairness concerns may span several sensitive groups and their intersections [31]. This setting is known as multidimensional fairness. For example, there may be multiple protected demographics (e.g., genders, races, and their intersections). Two frameworks for expressing several measures of bias in the multidimensional setting are a) what we later call *groups vs all* comparison [24] and b) *worst-case bias* [30]. In this work, we generalize these frameworks under a more expressive one. Our analysis also accounts for c) individual fairness by treating it under a multidimensional framework where each individual is a separate group of one element.

Groups vs all. This sets up the following generic framework for measures of bias:

$$F_{bias}(\$) = \odot_{\mathcal{S}_i \in \$}\, F_{base}(\mathcal{S}_i) \tag{2}$$

where \$ are all groups, $F_{base}(\mathcal{S}_i)$ is a measure of bias for one group $\mathcal{S}_i \in \$$ (this corresponds to treating that group as having only one binary sensitive attribute), and $\odot$ is a reduction

mechanism, such as the minimum or maximum. Groups may be overlapping when examining each sensitive attribute value independently without considering their intersections [32, 33, 34, 35]. It has been proposed that group intersections may also replace the groups within this analysis, therefore creating *subgroups* in their place [1, 31, 30].

In Equation 2, the base measure of bias compares one group against the total population. Example measures that employ this scheme are statistical parity subgroup fairness (spsf) and false positive subgroup fairness (fpsf) [31]. For these, each subgroup is compared to the total population to create multidimensional variations of Δeo and Δfpr respectively. Both employ a reduction mechanism that weighs comparisons by group size.

Worst-case bias. This framework starts from pairwise group or subgroup comparisons and keeps the worst case. We compute worst-case bias (wcb) for any probabilistic measure $F(\mathcal{S}_i)$, such as of accurate or erroneous predictions over groups or subgroups $\mathcal{S}_i \in \mathbb{S}$, per:

$$\text{wcb}(\mathbb{S}) = 1 - \min_{(\mathcal{S}_i, \mathcal{S}_j) \in \mathbb{S}^2} \frac{F(\mathcal{S}_i)}{F(\mathcal{S}_j)} \tag{3}$$

For example, differential fairness [1] states that prule should reside in the range $[e^{-\epsilon}, e^{\epsilon}]$ for some $\epsilon \geq 0$ when it compares *all* subgroups pairwise. A viable measure of bias for this definition, which we call differential bias (db), is given for subgroups $\mathbb{S}$ when $F(\mathcal{S}_i) = \text{P}(c(x) = 1 | x \in \mathcal{S}_i)$, for which differential fairness is equivalent to satisfying $\text{wcb}(\mathbb{S}) \leq 1 - e^{-\epsilon}$.

Individual fairness Multidimensional discrimination can also model individual fairness [10, 36] by setting each individual as a separate group. In this case, and given that individual fairness is formally defined to satisfy $d_y(c(x_i), c(x_j)) \leq d_x(x_i, x_j)$ across all pairs of individuals (x_i, x_j), where c is a predictive mechanism and d_y, d_x are some distance metrics, a viable measure of individual bias (ib) that satisfies individual fairness when $\text{ib} \leq 1$ is:

$$\text{ib} = \max_{x_i, x_j} \frac{d_y(c(x_i), c(x_j))}{d_x(x_i, x_j)} \tag{4}$$

3. A General Bias Measurement Framework

In this section we present a framework for defining measures of bias from fundamental building blocks. This lets us decompose existing measures into blocks, introduce variations, and create novel combinations. We recognize four types of blocks, which correspond to successive computational steps: a) selecting which pairs of population (sub)groups to compare, b) base measures that assess some system property on each group, c) comparisons between group assessments, and d) reductions that convert multiple pairwise comparisons to one value.

Mathematically, we annotate base measures as $F(\mathcal{S}_i)$ and compute them over groups of data samples $\mathcal{S}_i \subseteq \mathcal{S}_{all}$ of some population $\mathcal{S}_{all}$. We consider any information necessary for this computation (e.g., predicted and ground truth labels) directly retrievable from the respective samples. We also annotate the set of all these groups as $\mathbb{S} = \{\mathcal{S}_i : i = 0, 1, \dots\}$. The base measures could be error rates of predictions or unsupervised statistics, like positive rates. Then, we consider a set of subgroup pairs to compare $C(\mathbb{S}, \mathcal{S}_{all}) \in \left(2^{\mathcal{S}_{all}}\right)^2$, where $2^{\mathcal{S}_{all}}$ denotes the

powerset. The created set of subgroup pairs comprises any necessary detected comparisons between groups or subgroups within $\mathcal{S}_{all}$, even those that do not directly reside in $\mathbb{S}$ but may be derived from the latter, such as subgroups. We will make pairwise comparisons $f_{ij} = F(\mathcal{S}_i) \oslash F(\mathcal{S}_j)$ between (sub)groups $(\mathcal{S}_i, \mathcal{S}_j) \in C(\mathbb{S}, \mathcal{S}_{all})$. Finally, the reduction mechanism will be expressed as $\odot_{(S_i,S_j)} f_{ij}$ across all pairs (S_i, S_j) and outcomes of comparing them f_{ij}.

Putting the above in one formula yields our framework for expressing measures of bias:

$$F_{bias} = \odot_{(\mathcal{S}_i,\mathcal{S}_j)\in C(\mathbb{S},\mathcal{S}_{all})} \left(F(\mathcal{S}_i) \oslash F(\mathcal{S}_j)\right) \tag{5}$$

This is a direct generalization of Equation 2 in that it supports more varied strategies for expressing group comparisons. At the same time, it systematizes the comparison mechanisms between groups of people through the operation $\oslash$. Our framework is also a generalization of Equation 3 in that, in addition to the base measure (which we allow to also be non-probabilistic, if so desired), it provides flexibility in the reduction and comparison mechanisms beyond the choices of minimum and fractional comparison that characterize the worst case.

Each tuple of choices $(F, C, \oslash, \odot)$ creates a different measure of bias. In the next section, we delve into how each building block type among the members of this tuple could vary between contexts that comprise different fairness concerns and predictive tasks. For every building block, a systematic exploration would consider all possible combinations with others of different kinds so that eventually we not only reconstruct the original measures of bias, but also create variations that address different settings. Table 1 exemplifies how the measures discussed in Subsection 2.2 arise from specific choices. Creating a full taxonomy of existing measures under our framework is not this work's objective (we only aim to demonstrate its wide expressive breadth) and is left to future work.

Measure		$F(\mathcal{S}_i)$	$C(\mathbb{S}, \mathcal{S}_{all})$	$f_i \oslash f_j$	$\odot$
Example measures					
|Δfpr|	[23]	$\mathrm{P}(c(x) = 1 \mid x \in \mathcal{S}_i, Y(x) = 0)$	$\{\mathcal{S}, \mathcal{S}_{all} \setminus \mathcal{S}\}^2$ for $\mathbb{S} = \{\mathcal{S}\}$	$f_i - f_j$	max
|Δfnr|	[23]	$\mathrm{P}(c(x) = 0 \mid x \in \mathcal{S}_i, Y(x) = 1)$	$\{\mathcal{S}, \mathcal{S}_{all} \setminus \mathcal{S}\}^2$ for $\mathbb{S} = \{\mathcal{S}\}$	$f_i - f_j$	max
|Δ*eo*|	[25]	$\mathrm{P}(c(x) = 1 \mid x \in \mathcal{S}_i)$	$\{\mathcal{S}, \mathcal{S}_{all} \setminus \mathcal{S}\}^2$ for $\mathbb{S} = \{\mathcal{S}\}$	$f_i - f_j$	max
spsf	[31]	$\mathrm{P}(c(x) = 1 \mid x \in \mathcal{S}_i)$	$\mathbb{S} \times \{\mathcal{S}_{all}\} \cup \{\mathcal{S}_{all}\} \times \mathbb{S}$	$\lvert f_i - f_j \rvert$	wmean
fpsf	[31]	$\mathrm{P}(c(x) = 1 \mid x \in \mathcal{S}_i, Y(x) = 0)$	$\mathbb{S} \times \{\mathcal{S}_{all}\} \cup \{\mathcal{S}_{all}\} \times \mathbb{S}$	$\lvert f_i - f_j \rvert$	wmean
cv	[22]	$\mathrm{P}(c(x) = 1 \mid x \in \mathcal{S}_i)$	$\{\mathcal{S}, \mathcal{S}_{all} \setminus \mathcal{S}\}^2$	$f_i - f_j$	max
$1 - \mathrm{prule}$	[2]	$\mathrm{P}(c(x) = 1 \mid x \in \mathcal{S}_i)$	$\{\mathcal{S}, \mathcal{S}_{all} \setminus \mathcal{S}\}^2$	$1 - f_i/f_j$	max
db	[1]	$\mathrm{P}(c(x) = 1 \mid x \in \mathcal{S}_i)$	$\mathbb{S}^2$	$1 - f_i/f_j$	max
Multidimentional bias frameworks					
F_{bias}	[24]	any*	$\mathbb{S} \times \{\mathcal{S}_{all}\} \cup \{\mathcal{S}_{all}\} \times \mathbb{S}$	any	any
wcb	[30]	any	$\mathbb{S}^2$	$1 - f_i/f_j$	max
ib	[10]	$c(x)$	$\{\{x\} : x \in \mathcal{S}_{all}\}$	$d_y(f_i, f_j)/d_x(\mathcal{S}_i, \mathcal{S}_j)$	max

Table 1
Expressing the measures of classification bias of Subsection 2.2 and multidimensional bias frameworks of Subsection 2.3 under our bias measure definition framework. Interpretation of reductions can be found in Subsection 4.4. Frameworks allow for any blocks of certain kinds. *F_{bias} does not explicitly acknowledge base measure building blocks.

4. Bias building blocks

Here we present building blocks that occur from decomposing popular measures of bias of Sections 2.2 and 2.3. Combinations under our framework can create new measures.

4.1. Selecting groups or subgroups to compare

The group selection mechanisms we study include a) base selection, b) accounting for individual fairness [10], and c) accounting for intersectionality. More blocks can be created in the future.

Base selection. Several fairness measures are defined by comparing each population group or subgroup with the total population $\mathcal{S}_{all}$. Given that the pairwise comparison mechanism is not symmetric, i.e., it may hold that $f_i \oslash f_j \neq f_j \oslash f_i$, we account for both $(\mathcal{S}_i, \mathcal{S}_{all})$ and $(\mathcal{S}_{all}, \mathcal{S}_i)$ for subgroups $\mathcal{S}_i \in \mathbb{S}$. Mathematically, we define a *group vs any sample* selection per:

$$\text{vsany}(\mathbb{S}, \mathcal{S}_{all}) = \mathbb{S} \times \{\mathcal{S}_{all}\} \cup \{\mathcal{S}_{all}\} \times \mathbb{S} \tag{6}$$

Alternatively, one may consider all group pairs (we let reduction remove any self-comparisons):

$$\text{pairs}(\mathbb{S}, \mathcal{S}_{all}) = \mathbb{S}^2 \tag{7}$$

For one-dimensional fairness, i.e., only one sensitive attribute with only two potential values, such as indicating which samples belong to a protected group of people and which do not, each group $\mathcal{S}$ is typically compared to its *complement* within the whole population $\mathcal{S}_{all} \setminus \mathcal{S}_i$. In this case, $\mathbb{S} = \{\mathcal{S}\}$ and we write the pairwise selection between it and its complement per $\text{compl}(\mathbb{S}, \mathcal{S}_{all}) = \{\mathcal{S}, \mathcal{S}_{all} \setminus \mathcal{S}\}^2$. One generalization to multidimensional settings would be to consider $\mathcal{S}_{all} \setminus \mathcal{S}$ as a group and recreate Equation 7. However, an equally valid generalization it to compare groups to their complements only:

$$\text{compl}(\mathbb{S}, \mathcal{S}_{all}) = \bigcup_{\mathcal{S}_i \in \mathbb{S}} \{\mathcal{S}_i, \mathcal{S}_{all} \setminus \mathcal{S}_i\}^2 \tag{8}$$

Accounting for individual fairness. Individual fairness disregards the notion of population groups or subgroups and instead compares all individuals pairwise. This is an instance of the pairs mechanism, where each individual is assigned to their own group. Mathematically, this would occur if we specified subgroups $\mathbb{S} = \{\{x\} : x \in \mathcal{S}_{all}\}$. We hereby extend all comparison mechanisms C to follow this schema when an empty set of groups $\emptyset$ is provided[1] per:

$$C(\emptyset, \mathcal{S}_{all}) = C(\{\{x\} : x \in \mathcal{S}_{all}\}, \mathcal{S}_{all}) \tag{9}$$

Accounting for intersectionality. Measures of bias like db, spsf, and spsf account for subgroups that form intersections of groups too. To model this scenario, we define intersectional comparisons between groups while ignoring empty intersections (this conveniently ignores intersections between mutually exclusive sensitive attribute values) per:

$$C_{intersect}(\mathbb{S}, \mathcal{S}_{all}) = C(\{\mathcal{S} = \mathcal{S}_{i_1} \cap \mathcal{S}_{i_2} \cap \cdots : \mathcal{S}_{i_k} \in \mathbb{S}, \mathcal{S} \neq \emptyset\}, \mathcal{S}_{all}) \tag{10}$$

[1]The empty set indicates no knowledge of a group, in which case our fallback becomes individual fairness.

4.2. Base measures

Our framework accepts any measure of predictive performance. In this subsection, we demonstrate two concepts: a) computing aggregate assessments to feed as-are in comparison mechanisms, and b) tracking underlying metadata that could be used by other building blocks without altering Equation 5. Base measures can also be ported from fairness definitions that do not specify measures of bias, such as equal group benefit [37]. Our framework can supply the rest of building blocks to try all available options at defining bias, such as all comparison mechanisms.

Aggregate assessment. Here, we list base measures of aggregate system assessment often used to define bias. Probabilistic definitions for them can be found in domain literature. When working with classifiers, popular base measures are positive rates (pr), fpr, and fnr. Other misclassification measures are also used in expressions of disparate mistreatment [23]. In multiclass settings, each class can be treated as a different group of data samples with its own target labels. Bias has been assessed for the top-k individual/item recommendations by comparing the base measures of hit rate and skew [38, 39] across groups when checking for proportional representation. Recommendation correctness measures may be similarly accommodated, such as the click-through rate [39]. Recommendation is assessed across several values of k, which leads to measures that keep track of metadata (see below).

Keeping track of metadata. An emerging concern is that measures of bias affirm the presence of certain biases but not their absence. To verify the latter, one needs to look at nuances of underlying distributions [30]. For example, it has been argued that statistical tests should replace the prule to show non-discrimination in certain legal settings [40]. Thus, our framework retains building block metadata to be used in subsequent computations. To understand this, think of the abroca measure [29] that compares the area between the roc curves of Equation 1 instead of comparing aucs through differences or fractions.

We model curve comparisons by tracking computational metadata and retrieving them through an appropriate predicate curve(·) defined alongside base measures. For example, auc's definition as a building block should include the statement:

$$\text{curve}(\text{auc}(\mathcal{S}_i)) = \text{roc}(\mathcal{S}_i)$$

We do not directly return the metadata (e.g., the roc curves) in order to compare aggregate assessments with as many mechanisms as possible. For instance, also computing $|\text{auc}(\mathcal{S}_i) - \text{auc}(\mathcal{S}_j)|$ and checking differences with abroca may create high-level insights (e.g., if these two quantities are equal between two groups, the corresponding roc curves do not intersect).

We now present a second base measure used in recommendation systems to assess top predictions and contains a curve, namely average representation (ar) within the top predictions:

$$\text{ar}_K = \frac{1}{K} \sum_{k=1}^{K} \text{P}(x \in \mathcal{S}_i | x \in \text{top}_k)$$

To keep working with integratable curves with continuous horizontal axes, we use Dirac's delta function $\delta(k) = 0$ for $k \neq 0$ and $\int_{-\infty}^{\infty} \delta(k)\text{d}k = 1$ to define the injection:

$$\text{curve}(\text{ar}_K) = \left\{k \mapsto \text{P}(x \in \mathcal{S}_i | x \in \text{top}_k)\delta(0) \text{ if } k \in \{1, 2, \dots, K\}, 0 \text{ otherwise} : k \in (-\infty, \infty)\right\}$$

4.3. Comparison mechanisms

We examine three types of comparison to serve as building blocks in our framework: a) no comparison, b) numeric errors (differences and fractions), and c) curve comparisons. The last type encompasses mechanisms that compute weighed curve differences. Other mechanisms can be readily created, and we demonstrate thresholded variations.

No comparison. This is a valid option when aiming to remove confounding bias, i.e., bias that directly leads to erroneous system predictions. For example, the worst measure assessment across groups or subgroups is maximized if all groups exhibit high predictive performance [41]. In this case, our framework just assesses AI performance for all groups. Mathematically:

$$f_i \oslash_{none} f_j = f_i$$

Numeric errors. These compute some deviation between f_i and f_j, such as absolute error (abs), relative error (rel), and their signed values (sabs and srel respectively):

$$f_i \oslash_{abs} f_j = |f_i - f_j|\,,\ f_i \oslash_{rel} f_j = |1 - f_i/f_j|\,,\ f_i \oslash_{sabs} f_j = f_i - f_j\,,\ f_i \oslash_{srel} f_j = 1 - f_i/f_j$$

Curve comparisons. Let us consider that the curve predicate can extract from base measure assessment curves $crv_i = \text{curve}(f_i)$, $crv_j = \text{curve}(f_j)$ with a known common domain $\mathscr{D}$. We can compare these given a weighting mechanism $I(k)$ for the domain and comparison $\oslash_{crv}$ as:

$$f_i \oslash f_j = \frac{1}{\int_{\mathscr{D}} I(k)\mathrm{d}\theta} \int_{\mathscr{D}} I(k) \cdot \left(crv_i(k) \oslash_{crv} crv_j(k)\right) \mathrm{d}k$$

Essentially, we are building curve comparisons from the simpler components $(I, \oslash_{crv})$. As an example, we reconstruct the abroca pairwise group comparison [29] based on the auc base measure of Subsection 4.2 given the curve comparison defined by the pair of operations $\oslash_{abroca} = (I_{const}(k) = 1, \oslash_{abs})$. Measures that retain curves of top-k recommendations may also consider NDCG-like weighting $I_{DCG}(k) = \left(\frac{1}{\log(1+k)} \text{ if } k > 0, 0 \text{ otherwise}\right)$ [38, 39].

Thresholded variations. Oftentimes, small deviations from perfect fairness are accepted. Settling for thresholds under which measures of bias output 0 can take place either at the end of the assessment, or during intermediate steps. If the last option is chosen, comparisons $f_i \oslash f_j$ need to be replaced with the following variations of maximum acceptable threshold $\epsilon \geq 0$:

$$f_i \oslash_\epsilon f_j = \max\{0, f_i \oslash f_j - \epsilon\}$$

4.4. Reduction mechanisms

The final step of our framework consists of reducing to one quantity all pairwise comparisons f_{ij} between groups or subgroups $(\mathcal{S}_i, \mathcal{S}_j) \in C(\mathbb{S}, \mathcal{S}_{all})$. Common reductions are the maximum, minimum, or arithmetic mean of all values. Especially the maximum is a cornerstone of the worst-case bias framework, although it does not differentiate between systems that have the

same worst case. Alternatively, reduction could adopt some formula that becomes the weighted mean in the group vs all case [31]. Here, we demonstrate one feasible option, which obtains weights $1-|\mathrm{P}(x \in S_i)-\mathrm{P}(x \in S_{all})=1-|\mathrm{P}(x \in S_i)-1|=\mathrm{P}(x \in S_i)$ under vsany but also ignores self-comparisons among groups under pairs$(\cdot,\cdot)$ or compl$(\cdot,\cdot)$:

$$\mathrm{wmean}_{i,j} f_{ij} = \sum_{f_{ij}:\mathcal{S}_i \neq \mathcal{S}_j} (1-|\mathrm{P}(x \in S_i)-\mathrm{P}(x \in S_j)|) f_{ij}$$

5. The FairBench Library

We implement our bias measure definition framework within an open-source Python library called FairBench.[2] This focuses on ease-of-use, comprehensibility, and compatibility with popular working environments. To achieve these, we adopted a forward-oriented paradigm [42] to design the programming interfaces of code blocks as callable methods that can be combined in reporting mechanisms. Block parameters are transferred through keyword arguments.

5.1. Forks of sensitive attribute values

Our design is centered around uniformly representing many groups of data samples. Previous fairness libraries tend to parse lists of sensitive attribute names and match these with columns of clearly understood programming datatypes, such as Pandas dataframes [43]. However, coupling data loading and fairness assessment creates inflexible usage patterns and source code that is harder to extend. For example, it needs new methods and classes to support graph or image AI.

To bypass this issue, FairBench parses vectors of predictions (e.g., scores), ground truth (e.g., classification labels), and binary membership to protected groups. Vectors can come from any computational backend with a duck-type extension of Python's `Iterable` interface; they could be NumPy arrays [44], PyTorch tensors [45], TensorFlow tensors [46], JAX tensors [47], Pandas columns [43], or Python lists. We support end-to-end integration with automatic differentiation frameworks by extending their common functional interface provided by the EagerPy library [48] and setting the internal computational backend per `fairbench.backnend(name)`.

We simplify handling of multiple protected groups by organizing them into one data structure, which we call a `Fork` of the sensitive attribute. This is a programming equivalent of $. All interfaces accept a sensitive attribute fork, and base measures run for every group. Thus, every group being analysed becomes a computational branch of the fork. We provide dynamic constructor patterns to easily declare forks in common setups. One pattern is demonstrated below, where a fork `s` lets us compute the accuracy fork `acc` that holds assessment outcomes for both whites and blacks:

```
import fairbench as fb
race, predictions, labels = ...  # arrays, tensors, etc
s = fb.Fork(white=..., black=...) # branches hold binary iterables
acc = fb.accuracy(predictions=..., labels=..., sensitive=s)
fb.visualize(acc)  # visualize accuracy, which is also a fork
```

[2]The documentation of FairBench is available at: https://fairbench.readthedocs.io

Forks may also be constructed from categorical iterables (e.g., Pandas dataframe columns or native lists like `cats=['man', 'woman', 'man', 'nonbinary']`) with the constructor pattern `fb.Fork(fb.categories@cats)`. Forks of multidimensional sensitive attributes can be created by adding all categorical parsing to the constructor as positional arguments. Intersections between fork branches per Equation 10 can be achieved by calling a corresponding method:

```
races, genders = ...  # categorical iterables
s = fb.Fork(fb.categories@races, fb.categories@genders)
s = s.intersectional()
print(s.sum())  # visualization is not very instructive for many branches
```

5.2. Fairness reports

Multidimensional fairness reports for several default popular measures, comparisons, and reductions can be computed and displayed with the following snippet:

```
vsany = fb.unireport(predictions=..., labels=..., sensitive=s)
pairs = fb.multireport(predictions=..., labels=..., sensitive=s)
report = fb.combine(pairs, vsany)
fb.describe(report)  # or print or fb.visualize
```

This combines the comparisons of Equation 6 (unireport) and Equation 7 (multireport). For more report types or how to declare one specific measure of bias, refer to the library's documentation. The base measures to analyse are determined from keyword arguments. For example, the above code snippet computes classification base measures, but if we added a `scores=...` arguments (in addition to or instead of `predictions`) recommendation measures would be obtained too. Reports can also parse a custom selection of base measures, including externally defined ones.

The outcome of running the above code is presented in Figure 1. Column names correspond to the combination of reduction and comparison mechanisms, whereas rows to the base measures. Familiarization with both the library and our mathematical framework is required to understand FairBench reports, but these present -to our knowledge- the first systematic way of looking at many bias assessments at once, regardless of the exact setting. For example, in the report below it is easy to detect the small minimum ratio (minratio) of pr, which corresponds to prule, even if there are generally balanced tpr and tnr (equivalent to balanced fnr and fpr).

Metric	min	wmean	minratio	maxdiff	min[vsAny]	wmean[vsAny]	minratio[vsAny]	maxdiff[vsAny]
accuracy	0.798	0.837	0.872	0.117	0.798	0.837	0.953	0.078
pr	0.049	0.189	0.191	0.209	0.049	0.189	0.261	0.139
tpr	0.690	0.708	0.925	0.056	0.690	0.708	0.993	0.051
tnr	0.836	0.865	0.905	0.088	0.836	0.865	0.960	0.054

Figure 1: Example of a combined FairBench report.

Finally, reports can be interactively explored through `fb.interactive(report)`. This uses the Bokeh library [49] to run in notebook outputs or new browser tabs and create visualization

similar to Figure 2. Through the visual interface, users can not only see report values but also focus on columns or rows, and delve into explanations of which raw values contributed to computations, as indicated via the `.explain` part of the exploration path on top. Explanations correspond to tracking metadata values through predicates per Subsection 4.2. The same exploration can be performed purely programmatically too.

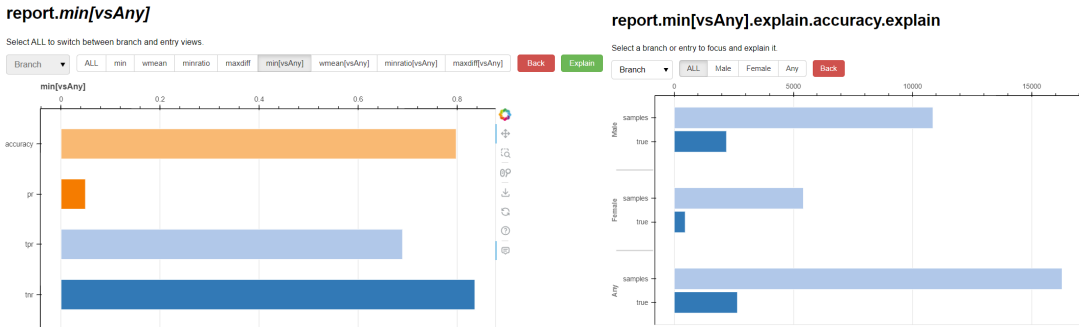

Figure 2: Example steps during the interactive exploration of a FairBench report.

6. Conclusions

In this work we provided a mathematical framework that standardizes the definition of many existing and new measures of bias by splitting them in base building blocks and compiling all possible combinations. This framework systematically explores AI fairness through measures that account for a wide range of settings and bias concerns, beyond the confines of ad-hoc exploration. Implementation of popular building blocks are provided in the FairBench library via interoperable Python interfaces. The library provides additional features, like visualization and computation backtracking, that enable in-depth exploration of a wide range of bias concerns. It can also be used alongside popular computational backends.

Future research and development could work towards identifying more bias building blocks by decomposing more measures from the literature, as well as additional block types that may arise in the future. FairBench is open source and we also encourage implementations of such analysis from the community. We further plan to extend currently experimental features, like creating fairness model cards that include caveats and recommendations obtained from interdisciplinary collaborations with social scientists, and creating higher-level assessments.

Acknowledgement

This research work was funded by the European Union under the Horizon Europe MAMMOth project, Grant Agreement ID: 101070285. UK participant in Horizon Europe Project MAMMOth is supported by UKRI grant number 10041914 (Trilateral Research LTD).

References

[1] J. R. Foulds, R. Islam, K. N. Keya, S. Pan, An intersectional definition of fairness, in: 2020 IEEE 36th International Conference on Data Engineering (ICDE), IEEE, 2020, pp. 1918–1921.

[2] D. Biddle, Adverse impact and test validation: A practitioner's guide to valid and defensible employment testing, Routledge, 2017.

[3] P. Ala-Pietilä, Y. Bonnet, U. Bergmann, M. Bielikova, C. Bonefeld-Dahl, W. Bauer, L. Bouarfa, R. Chatila, M. Coeckelbergh, V. Dignum, et al., The assessment list for trustworthy artificial intelligence (ALTAI), European Commission, 2020.

[4] N. AI, Artificial intelligence risk management framework (ai rmf 1.0) (2023).

[5] B. Li, P. Qi, B. Liu, S. Di, J. Liu, J. Pei, J. Yi, B. Zhou, Trustworthy ai: From principles to practices, ACM Computing Surveys 55 (2023) 1–46.

[6] E. Ntoutsi, P. Fafalios, U. Gadiraju, V. Iosifidis, W. Nejdl, M.-E. Vidal, S. Ruggieri, F. Turini, S. Papadopoulos, E. Krasanakis, et al., Bias in data-driven artificial intelligence systems—an introductory survey, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 10 (2020) e1356.

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

[8] S. Mitchell, E. Potash, S. Barocas, A. D'Amour, K. Lum, Algorithmic fairness: Choices, assumptions, and definitions, Annual Review of Statistics and Its Application 8 (2021) 141–163.

[9] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, A survey on bias and fairness in machine learning, ACM computing surveys (CSUR) 54 (2021) 1–35.

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

[11] M. J. Kusner, J. Loftus, C. Russell, R. Silva, Counterfactual fairness, Advances in neural information processing systems 30 (2017).

[12] A. N. Carey, X. Wu, The causal fairness field guide: Perspectives from social and formal sciences, Frontiers in Big Data 5 (2022) 892837.

[13] L. Rosenblatt, R. T. Witter, Counterfactual fairness is basically demographic parity, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, 2023, pp. 14461–14469.

[14] V. Xinying Chen, J. Hooker, A guide to formulating fairness in an optimization model, Annals of Operations Research (2023) 1–39.

[15] J. Kleinberg, S. Mullainathan, M. Raghavan, Inherent trade-offs in the fair determination of risk scores, arXiv preprint arXiv:1609.05807 (2016).

[16] T. Miconi, The impossibility of" fairness": a generalized impossibility result for decisions, arXiv preprint arXiv:1707.01195 (2017).

[17] R. K. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilović, et al., Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias, IBM Journal of Research and Development 63 (2019) 4–1.

[18] S. Bird, M. Dudík, R. Edgar, B. Horn, R. Lutz, V. Milan, M. Sameki, H. Wallach, K. Walker,

Fairlearn: A toolkit for assessing and improving fairness in ai, Microsoft, Tech. Rep. MSR-TR-2020-32 (2020).
[19] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viégas, J. Wilson, The what-if tool: Interactive probing of machine learning models, IEEE transactions on visualization and computer graphics 26 (2019) 56–65.
[20] P. Saleiro, B. Kuester, L. Hinkson, J. London, A. Stevens, A. Anisfeld, K. T. Rodolfa, R. Ghani, Aequitas: A bias and fairness audit toolkit, arXiv preprint arXiv:1811.05577 (2018).
[21] A. Castelnovo, R. Crupi, G. Greco, D. Regoli, I. G. Penco, A. C. Cosentini, The zoo of fairness metrics in machine learning (2021).
[22] T. Calders, S. Verwer, Three naive bayes approaches for discrimination-free classification, Data mining and knowledge discovery 21 (2010) 277–292.
[23] M. B. Zafar, I. Valera, M. Gomez Rodriguez, K. P. Gummadi, Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment, in: Proceedings of the 26th international conference on world wide web, 2017, pp. 1171–1180.
[24] A. Roy, J. Horstmann, E. Ntoutsi, Multi-dimensional discrimination in law and machine learning-a comparative overview, in: Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, 2023, pp. 89–100.
[25] M. Hardt, E. Price, N. Srebro, Equality of opportunity in supervised learning, Advances in neural information processing systems 29 (2016).
[26] M. Donini, L. Oneto, S. Ben-David, J. S. Shawe-Taylor, M. Pontil, Empirical risk minimization under fairness constraints, Advances in neural information processing systems 31 (2018).
[27] S. Tsioutsiouliklis, E. Pitoura, P. Tsaparas, I. Kleftakis, N. Mamoulis, Fairness-aware pagerank, in: Proceedings of the Web Conference 2021, 2021, pp. 3815–3826.
[28] T. Calders, A. Karim, F. Kamiran, W. Ali, X. Zhang, Controlling attribute effect in linear regression, in: 2013 IEEE 13th international conference on data mining, IEEE, 2013, pp. 71–80.
[29] J. Gardner, C. Brooks, R. Baker, Evaluating the fairness of predictive student models through slicing analysis, in: Proceedings of the 9th international conference on learning analytics & knowledge, 2019, pp. 225–234.
[30] A. Ghosh, L. Genuit, M. Reagan, Characterizing intersectional group fairness with worst-case comparisons, in: Artificial Intelligence Diversity, Belonging, Equity, and Inclusion, PMLR, 2021, pp. 22–34.
[31] M. Kearns, S. Neel, A. Roth, Z. S. Wu, Preventing fairness gerrymandering: Auditing and learning for subgroup fairness, in: International conference on machine learning, PMLR, 2018, pp. 2564–2572.
[32] M. Kearns, S. Neel, A. Roth, Z. S. Wu, An empirical study of rich subgroup fairness for machine learning, in: Proceedings of the conference on fairness, accountability, and transparency, 2019, pp. 100–109.
[33] J. Ma, J. Deng, Q. Mei, Subgroup generalization and fairness of graph neural networks, Advances in Neural Information Processing Systems 34 (2021) 1048–1061.
[34] N. L. Martinez, M. A. Bertran, A. Papadaki, M. Rodrigues, G. Sapiro, Blind pareto fairness and subgroup robustness, in: International Conference on Machine Learning, PMLR, 2021, pp. 7492–7501.

[35] C. Shui, G. Xu, Q. Chen, J. Li, C. X. Ling, T. Arbel, B. Wang, C. Gagné, On learning fairness and accuracy on multiple subgroups, Advances in Neural Information Processing Systems 35 (2022) 34121–34135.
[36] M. P. Kim, A. Korolova, G. N. Rothblum, G. Yona, Preference-informed fairness, arXiv preprint arXiv:1904.01793 (2019).
[37] J. Gartner, A new metric for quantifying machine learning fairness in healthcare, https://www.closedloop.ai/blog/a-new-metric-for-quantifying-machine-learning-fairness-in-healthcare/, 2020. Accessed: 17-2-2024.
[38] M. Zehlike, K. Yang, J. Stoyanovich, Fairness in ranking: A survey, arXiv preprint arXiv:2103.14000 (2021).
[39] E. Pitoura, K. Stefanidis, G. Koutrika, Fairness in rankings and recommendations: an overview, The VLDB Journal (2022) 1–28.
[40] E. A. Watkins, M. McKenna, J. Chen, The four-fifths rule is not disparate impact: a woeful tale of epistemic trespassing in algorithmic fairness, arXiv preprint arXiv:2202.09519 (2022).
[41] I. Sarridis, C. Koutlis, S. Papadopoulos, C. Diou, Flac: Fairness-aware representation learning by suppressing attribute-class associations, arXiv preprint arXiv:2304.14252 (2023).
[42] E. Krasanakis, A. L. Symeonidis, Forward oriented programming: A meta-dsl for fast development of component libraries, Available at SSRN 4180025 (????).
[43] W. McKinney, et al., pandas: a foundational python library for data analysis and statistics, Python for high performance and scientific computing 14 (2011) 1–9.
[44] C. R. Harris, K. J. Millman, S. J. Van Der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, et al., Array programming with numpy, Nature 585 (2020) 357–362.
[45] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32 (2019).
[46] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., Tensorflow: Large-scale machine learning on heterogeneous distributed systems, arXiv preprint arXiv:1603.04467 (2016).
[47] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, Q. Zhang, JAX: composable transformations of Python+NumPy programs, 2018. URL: http://github.com/google/jax.
[48] J. Rauber, M. Bethge, W. Brendel, Eagerpy: Writing code that works natively with pytorch, tensorflow, jax, and numpy, arXiv preprint arXiv:2008.04175 (2020).
[49] C. Bokeh Development Team, Bokeh: Python library for interactive visualization, 2014.