Evolved Sample Weights for Bias Mitigation: Effectiveness Depends on Optimization Objectives

ANIL K. SAINI, Cedars-Sinai Medical Center, Los Angeles, USA
JOSE GUADALUPE HERNANDEZ, Cedars-Sinai Medical Center, Los Angeles, USA
EMILY F. WONG, Cedars-Sinai Medical Center, Los Angeles, USA
DEBANSHI MISRA, University of California, Los Angeles, USA
JASON H. MOORE, Cedars-Sinai Medical Center, Los Angeles, USA

Machine learning models trained on real-world data may inadvertently make biased predictions that negatively impact marginalized communities. Reweighting is a method that can mitigate such bias in model predictions by assigning a weight to each data point used during model training. In this paper, we compare three methods for generating these weights: (1) evolving them using a Genetic Algorithm (GA), (2) computing them using only dataset characteristics, and (3) assigning equal weights to all data points. Model performance under each strategy was evaluated using paired predictive and fairness metrics, which also served as optimization objectives for the GA during evolution. Specifically, we used two predictive metrics (accuracy and area under the Receiver Operating Characteristic curve) and two fairness metrics (demographic parity difference and subgroup false negative fairness). Using experiments on eleven publicly available datasets (including two medical datasets), we show that evolved sample weights can produce models that achieve better trade-offs between fairness and predictive performance than alternative weighting methods. However, the magnitude of these benefits depends strongly on the choice of optimization objectives. Our experiments reveal that optimizing with accuracy and demographic parity difference metrics yields the largest number of datasets for which evolved weights are significantly better than other weighting strategies in optimizing both objectives.

CCS Concepts: • **Computing methodologies** → **Genetic algorithms**; • **Applied computing** → *Health informatics.*

Additional Key Words and Phrases: genetic algorithm, fairness, reweighting

ACM Reference Format:
Anil K. Saini, Jose Guadalupe Hernandez, Emily F. Wong, Debanshi Misra, and Jason H. Moore. 2025. Evolved Sample Weights for Bias Mitigation: Effectiveness Depends on Optimization Objectives. 1, 1 (November 2025), 15 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

While machine learning (ML) has revolutionized numerous industries, it has also demonstrated the ability to perpetuate racial, gender, and other biases captured within a dataset [1]. In areas where these ML systems are used to make high-stakes decisions, such as healthcare, algorithmic bias can have unintended negative consequences (e.g., widening

Authors' Contact Information: Anil K. Saini, anil.saini@cshs.org, Cedars-Sinai Medical Center, Los Angeles, California, USA; Jose Guadalupe Hernandez, Cedars-Sinai Medical Center, Los Angeles, California, USA, jose.hernandez8@cshs.org; Emily F. Wong, Cedars-Sinai Medical Center, Los Angeles, California, USA, emily.wong@cshs.org; Debanshi Misra, University of California, Los Angeles, California, USA, debanshi@ucla.edu; Jason H. Moore, Cedars-Sinai Medical Center, Los Angeles, California, USA, jason.moore@csmc.edu.

Manuscript submitted to ACM

health disparities). All ML, regardless of whether the models are learning through supervised, unsupervised, or semi-supervised approaches, requires data. As such, there is a risk of algorithmic bias for all ML methods, given that the biases captured within the data may be unknown. Bias can arise from various sources, such as the use of incorrect features and the lack of diversity during sampling [19]. Bias may also be introduced by the configuration of an algorithm (e.g., the choice of optimization functions or regularization) [19].

Bias can be ameliorated at various stages of using the ML model: pre-processing, which modifies data prior to training and evaluation; in-processing, which involves tuning the algorithm during the training process; and post-processing, which adjusts predictions after training. Reweighting is a widely used pre-processing approach to mitigate bias in model predictions. It involves assigning weights to data points in the training set (called 'sample weights' hereafter) that are utilized by machine learning models during training to adjust the contribution of different data points to the loss function of the model.

Two prominent strategies exist for computing sample weights. The first is a deterministic approach, where weights are derived directly from dataset characteristics (e.g., Kamiran & Calders [15]). The second is an optimization-based approach, where weights are evolved through an evolutionary algorithm to jointly improve predictive performance and fairness with respect to a specific machine learning model (e.g., La Cava [18]). Deterministic reweighting has been shown to improve model fairness [22, 24], and recent work suggests that evolutionary approaches to learning sample weights can produce stronger error–fairness trade-offs than methods such as GerryFair [16]. However, the relative performance of evolved weights versus deterministic weights, particularly under different combinations of predictive (e.g., accuracy) and fairness (e.g., demographic parity difference) metrics, has not been systematically examined.

In this work, we address this gap by making the following contributions:

(1) We compare three methods for computing sample weights: (i) evolving them using a Genetic Algorithm, (ii) deriving them from dataset characteristics, and (iii) assigning equal weights.
(2) We evaluate these reweighting strategies under multiple combinations of predictive and fairness metrics.

Across eleven publicly available datasets, our experiments show that evolved sample weights yield models with better trade-offs between predictive performance and fairness compared to alternative weighting methods. Importantly, the extent of these improvements depends on the specific predictive and fairness metrics used during optimization.

2 Background and Related Work

When working with real-world data, machine learning models can exhibit algorithmic bias, which refers to systematic errors in the modeling process that produce lower-quality or less desirable predictions for certain historically disadvantaged communities, such as people of color and women [17].

Beyond data quality and variable selection, algorithmic bias can also result from a particular model's decision boundary. To illustrate this point, consider a case study on hiring decisions. Employers typically prefer to streamline administrative processes, such as hiring, especially when they receive a large volume of applications. In this scenario, it would be useful to build an ML model that predicts the success of prospective candidates. An ML model trained on previous hiring decisions may have many potential decision boundaries. Figure 1 displays two illustrative decision boundaries. Applicant groups are represented by circles and triangles, and each point's true label (accepted or rejected) is indicated by its solid fill color. For both boundaries, the area above the diagonal (shaded in red) denotes a rejection prediction, and the area below (shaded in blue) denotes an acceptance prediction.

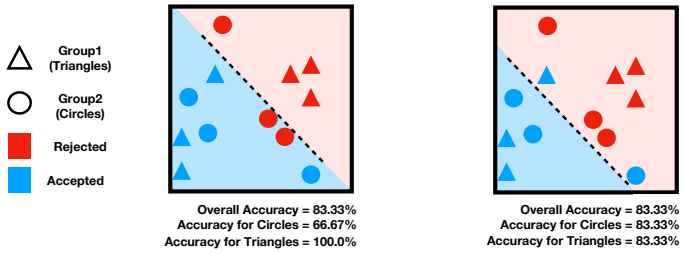

Fig. 1. An example of two different decision boundaries producing the same overall accuracy during training on a given dataset. The decision boundary on the left makes less fair predictions than the decision boundary on the right.

While both decision boundaries yield the same overall accuracy, their predictive performance varies between the two groups: the accuracy is higher for Triangles than Circles for the leftmost boundary in Figure 1, whereas the accuracy is the same for both groups with the rightmost boundary. This example illustrates how modeling decisions, such as selecting decision boundaries, can negatively bias predictions against certain groups, while also showing how modeling decisions can reduce disparities without sacrificing overall predictive performance.

2.1 Measuring Bias

Multiple metrics exist to quantify the fairness of predictions made by an ML model [1]. Each metric is specific to the application context and attempts to quantify unique properties (false negative rate, accuracy, etc.) of the predictions for individuals belonging to different groups. Generally, since the fairness metrics try to measure the disparity in the model predictions between groups, lower values on these metrics indicate less disparity and hence greater fairness. From the available metrics, we choose two metrics, one that does not consider the types of incorrect predictions (demographic parity difference) and one that does take such information into account (subgroup false negative rate).

Before describing the metrics, we first must specify *sensitive* and *non-sensitive* features in the dataset under consideration. 'Sensitive' (or 'protected') features specify the attributes of individuals a model should be sensitive to when assessing group fairness (e.g., race, sex, or gender), whereas all other features are considered non-sensitive. Based on the values of sensitive attributes, each data point can fall into one of the groups defined by a combination of those sensitive attributes. For example, 'Black women younger than 25' would be one of the groups when the sensitive attributes are race, gender, and age. Let $G \in \mathcal{G}$ be one such group. For some metrics, in a binary classification setting, one value of the target variable is considered 'favorable' (or desired), while the other value is not. This favorable value is usually represented by 1. For example, in a dataset containing records of applicants applying for loans, having received an approval would be considered favorable (and assigned a value of 1)

Demographic Parity Difference (DPD): This metric measures the difference between the largest and the smallest group-level 'acceptance rate', which is defined as the proportion of individuals belonging to the group receiving a favorable prediction. DPD can be given mathematically as

$$DPD = \max_{G_1, G_2} [Pr(\hat{Y} = 1 | X \in G_1) - Pr(\hat{Y} = 1 | X \in G_2)],$$

where $\hat{Y}$ is the predicted value of the target variable, G_1, G_2 are two subgroups withing the dataset, X is a given data point in the dataset, and $Pr(.)$ is the probability estimate.

Subgroup False Negative Fairness (SFN) [16, 18]: It captures the maximum deviation of a model's performance in terms of false negative rate among any one group in $\mathcal{G}$. The deviation is normalized by the probability of observing an individual from that group having positive labels. The SFN can be given mathematically by

$$SFN = \max_{G \in \mathcal{G}} [\alpha(G)\beta(G)].$$

The term α denotes the probability of getting positive labels in the group G, and the term β refers to the absolute difference between the overall false negative rate and the false negative rate within the group G.

Note that for many real-world applications (including medical diagnosis), a missed detection can be more costly than a false alarm. Therefore, we used the Subgroup False Negative Fairness as a proof of concept in our experiments. Due to computational constraints, we did not use Subgroup False Positive Fairness, although it can be computed in a similar manner to SFN.

2.2 Mitigating Bias

Many techniques have been proposed to make ML algorithms less biased towards certain communities [5]. Most of these methods can fall into three categories:

- **Pre-processing:** Methods in this category either change some properties of the data, modify some values in the data, or change the loss function contribution of data points used by the model. Examples include assigning weights to data points in the training data to be used by the loss function [14, 15], modifying labels for some data points in the dataset [15], feature selection [25], and feature transformation [3].
- **In-processing:** These methods modify the ML models directly to reduce bias in their predictions. For example, in Zhang et al [26], the authors add an adversarial model that predicts the sensitive attributes for a given predicted output. The combination of the original predictor model and the adversarial model gives rise to a model with better values on fairness metrics.
- **Post-processing:** These methods adjust model predictions to make them less discriminatory. Examples include methods that change the threshold for risk scores of an already existing, possibly discriminatory, predictor [10].

2.3 Reweighting

In this work, we discuss the Reweighting method [15], which mitigates bias by assigning weights to data points in the training dataset. These weights are used while computing the loss function during the training phase and hence adjust each data point's contribution to the training loss. In this work, we will focus on the classifiers that support the use of weights in this way.

In its most basic form, through reweighting, we assign higher weights (> 1) to data points that possess one of the following two characteristics: (1) individuals belonging to 'deprived' communities that possess desirable values of the target variable (e.g., female candidates getting accepted for a STEM job), and (2) individuals belonging to 'favored' communities that possess undesirable values of the target variable (e.g., male candidates getting rejected for a STEM job)[15]. All other data points are assigned lower weights (< 1). Determining which communities are favored and which target values are desirable is subjective and dependent on the dataset being used. The weight assigned for a given data point X is defined by:

$$W(X) = \frac{P(S = X(S)) \cdot P(Class = X(Class))}{P(S = X(S) \wedge Class = X(Class))}, \tag{1}$$

where $X(S)$ is the value of the sensitive feature in X, $X(Class)$ is the value of the target variable in X, $P(S = X(S))$ is the probability of observing a measurement in the dataset with the value of sensitive feature being $X(S)$, $P(Class = X(Class))$ is the probability of a given observation having value of the target variable as $X(Class)$, and $P(S = X(S) \wedge Class = X(Class))$ is the probability of a given observation having the value of sensitive feature as $X(S)$ and target variable as $X(Class)$.

Table 1. Example dataset to illustrate the Reweighting Method. The column titled 'Weight' is calculated using the Reweighting procedure.

Race	Position	Oral	Written	Combined	Promotion	Weight
W	Captain	89.52	95	92.808	1	0.84
W	Captain	80	95	89	1	0.84
W	Captain	82.38	87	85.152	1	0.84
W	Captain	88.57	76	81.028	0	1.4
H	Lieutenant	76.19	84	80.876	0	0.6
H	Captain	76.19	82	79.676	0	0.6
W	Captain	76.19	82	79.676	1	0.84
H	Lieutenant	70	84	78.4	1	1.8
W	Captain	73.81	81	78.124	0	1.4
W	Lieutenant	84.29	72	76.916	1	0.84

Reweighting can be illustrated with the following example. Table 1 shows a dataset containing the oral, written, and combined test scores for a promotion exam for a Fire Department (adapted from the RICCI dataset [20]). The race and current position of each test taker are also given.

Using 'Race' as the sensitive feature and 'Promotion' as the target variable yields four combinations of Race and Promotion. For each combination, we can calculate the weights according to Equation 1 (see Table 2). The resulting weights for each data point in Table 1 can be found in the last column. Note that even though weights are assigned to all points in the training dataset, only $2^{(|sensitive\_attributes|+1)}$ weights (combination of different values for sensitive features and the target variable) are computed for binary classification datasets with binary sensitive features.

We refer to the weighting strategy given by Kamiran & Calders[15] as 'Deterministic Reweighting' in order to differentiate it from 'Evolved Reweighting' described in the next section.

3 Genetic algorithm to optimize sample weights

Genetic Algorithms (GAs) [21] are a collection of methods that draw inspiration from the theory of natural selection to solve optimization problems by initializing a set of potential solutions (i.e., population) and evolving those solutions

Table 2. Intermediate and final values calculated during the reweighting procedure for the example given in Section 2.3.

Race (S)	Promotion (C)	$P(S = X(S))$	$P(C = X(C))$	$P(S = X(S) \wedge C = X(C))$	$W(X)$
W	1	7 / 10	6/10	5/10	0.84
W	0	7 / 10	4/10	2/10	1.4
H	1	3 / 10	6/10	1/10	1.8
H	0	3 / 10	4/10	2/10	0.6

to optimize one or more objective functions. GAs have been successfully used in prior research for multi-objective optimization [6].

In this work, we describe a GA for discovering a set of sample weights optimized for both predictive performance (e.g., AUROC) and fairness (e.g., subgroup false negative fairness); the predictive metric is maximized while the fairness metric is minimized. The sample weights evolved using the GA are used during training to mitigate bias. The GA requires the following inputs: population size, weight dimensionality, maximum number of generations, a machine learning method, and training data. The dimensionality of weights (*ind_size*) is determined by the total possible combination of values from sensitive attributes and the target variable; all other inputs are user-specified.

Algorithm 1 Genetic Algorithm for Reweighting

```
1:  procedure Evaluate(pop, ml, X, y)
2:      for ind in pop do
3:          assigned_weight = Expand_and_Assign(ind)        ▷ Assign weights for all data points
4:          ind['auroc'] = AUROC(ml, X, y, assigned_weight)        ▷ AUROC on validation set
5:          ind['fairness'] = FNSF(ml, X, y, assigned_weight)      ▷ Fairness on validation set
6:  procedure ReweightingGA(pop_size, ind_size, max_gen, ml, X, y)
7:      pop = InitializePop(pop_size, ind_size)        ▷ generate pop_size weights of length ind_size
8:      Evaluate(pop, ml, X, y)
9:      NonDominatedScores(pop)
10:     for g in {0,...,max_gens} do
11:         offspring = []
12:         for p in 1...pop_size do
13:             parent_a, parent_b = NonDominatedBinaryTournament(pop)
14:             child = Crossover(parent_a, parent_b)
15:             child = Mutation(child)
16:             offspring.append(child)
17:             Evaluate(offspring, ml, X, y)
18:             NonDominatedScores(pop+offspring)
19:         pop = SurvivalSelection(pop+offspring)
20:     Return all_evaluated_individuals        ▷ All evaluated sample weights
```

The procedures implemented by the GA can be found in Algorithm 1. At the beginning of an evolutionary run, the starting population is initialized with *pop_size* sample weights of size *ind_size*; values for these weights are drawn at random from a uniform distribution between $[0.0, 2.0)$. Each sample weight's length corresponds to the number of unique combinations for sensitive features and the target variables: $2^{(|sensitive\_attributes|+1)}$. Once the initial population is constructed, each set of weights and training data is used to train a machine learning model. When a sample weight is evaluated, we assign the appropriate weight to all the data points in the training dataset that correspond to the specific combination of sensitive features and target variable (line 3 in Algorithm 1). This weight assignment is similar to how weights are calculated and assigned in the example from Section 2.3.

We apply 10-fold cross-validation to compute predictive performance (e.g., AUROC) and fairness (e.g, False Negative Subgroup Fairness) scores on the training data. These scores (i.e, the average predictive performance and the average fairness across 10 folds) determine each individual's Pareto front rank and crowding distance (line 9 in Alg. 1; see Deb et al.[7], Section III-B). An individual's rank reflects its Pareto optimality relative to others: the first (best) front consists of nondominated solutions, where solution X dominates solution Y if X is at least as good as Y in all objectives and

strictly better in at least one. Subsequent fronts are formed iteratively from the remaining solutions. The crowding distance measures how isolated a pipeline is within its front; higher distances are preferred as they help preserve diversity. These two attributes—Pareto front rank and crowding distance—guide parent selection (line 6 in Alg. 1) using Nondominated Binary Tournament Selection. In each selection event, two individuals are compared: those with higher ranks are discarded first, then among those remaining, pipelines with lower crowding distance are removed. If multiple candidates remain, one is randomly chosen. This approach balances exploitation of individuals close to the true Pareto front and exploration of diverse regions.

Two parents are needed to generate a single offspring (i.e., parent selection is used twice per offspring created). There is a 0.8 probability we use crossover to generate a single offspring from both parents, where each value of the offspring's weights comes from either parent with equal probability. Otherwise, the first parent is directly returned as the offspring. Once the offspring is constructed, there is a 0.1 probability of a point mutation applied to individual values within the set of weights; the magnitude of a point mutation comes from a normal distribution with a mean of 0.0 and a standard deviation of 1.0. The value of each element is capped on both sides to be in the $[0.0, 2.0]$ range. A total of *pop_size* offspring are constructed and then set as the new population.

After generating offspring, we evaluate them like their parents (line 17 in Alg. 1) and assign Pareto front ranks and crowding distances relative to the combined set of parents and offspring (line 18). Survival selection (line 19) then reduces this combined set back to the original population size by retaining Pareto-optimal individuals. The individuals from the first (best) front are selected first; if this front contains too many, those with higher crowding distance are prioritized. If more individuals are needed, pipelines from subsequent fronts are added in order of front rank, again preferring those with higher crowding distance. The resulting survivors form the next generation, and the same evolutionary cycle of evaluation, selection, and reproduction is repeated for a specified number of generations. The values we use for crossover and mutation probabilities are typical for the GA literature; prior GA works suggest a low mutation probability (<= 0.1) and a high crossover probability (>=0.5) [11].

After the final generation, a Pareto-front is constructed from all sample weights evaluated throughout the evolutionary search as follows. We retrain all evaluated models and sample weights with the entire training data, and then obtain their predictive and fairness scores on the test set; these scores on the test set are used to construct the Pareto-front.

The method to evolve weights described here is very similar to the one proposed by La Cava [18], with one important distinction: we do not directly optimize the weight vector for the whole dataset, but instead a weight vector whose length corresponds to the combination of possible values of the sensitive features and the target variable, which is typically less than the total number of samples in the training dataset. Additionally, our experiments in this paper are different from those in La Cava [18] in two major ways: we compare evolved weights with deterministic weights, and we used more combinations of performance and fairness metrics for that comparison.

4 Methods

4.1 Comparing Reweighting Methods

We conducted experiments to evaluate the three reweighting methods across four combinations of predictive and fairness metrics. As discussed in the following sections, each method is assessed based on its ability to jointly optimize both the predictive and fairness objectives. We used the following three methods for calculating sample weights: (1) equal weights, (2) deterministic weights [15], and (3) evolved weights. As predictive performance, we used two commonly used metrics [23]: (a) accuracy (ACC), which focuses on the number of accurate predictions without considering the

kind of errors (false negatives or false positives), and (b) Area Under the Receiver Operating Characteristic curve (ROC), which takes into account the true positive and false positive rates.

In a similar vein for fairness metrics, we use one metric that does not consider the types of incorrect predictions (demographic parity difference, DPD) and one that does take such information into account (false negative subgroup fairness, SFN). We evaluated each of the three weighting methods on 12 datasets (including two medical datasets). The collection of three reweighting methods, four combinations of evaluation metrics, and 12 datasets yields 108 experiments. We conducted 20 replicates for each of these experiments. Each replicate corresponds to a unique test-train split and includes 1000 model evaluations. We restricted our work to a predetermined configuration of a Random Forest Classifier model (see supplementary material), where different seeds can lead to unique forest structures for a given data split.

Parameters specific to each reweighting method are give as follows:

(1) **Equal Weights (EQ):** All data points are weighted equally (with a weight of 1.0). For each of the 20 replicates, the model was evaluated 1000 times using unique seeds.
(2) **Deterministic Weights (DW):** Sample weights are calculated using the procedure described in Section 2.3. For each of the 20 replicates, the model was evaluated 1000 times using unique seeds.
(3) **Evolved Weights (EW):** A population of 20 individuals (sets of sample weights) is evolved for 50 generations, resulting in 1000 total model evaluations with our GA approach (Algorithm 1). For each of the 20 replicates, the same model (i.e. all models use the same seed in a replicate) was run 1000 times with different sample weights.

4.2 Real World Medical Datasets

We use medical data on perinatal mood and anxiety disorders (PMADs) from Wong et al. [24] to validate the effectiveness of our method. This data was collected through the Postpartum Depression Screening, Education, and Referral Quality Improvement Initiative at Cedars-Sinai Medical Center (CSMC) in Los Angeles, California, between 2020 and 2023. We include data from birthing individuals who were admitted to the postpartum unit or the maternal-fetal care unit after delivery; individuals from the prenatal/pre-delivery time point or those who experienced stillbirth were not included. The CSMC's Institutional Review Board approved the use of de-identified patient data. We evaluated each of the three weighting methods on two datasets corresponding to two methods of screening for PMAD. In February 2022, screening for PMAD transitioned from the PHQ-9 to the EPDS, which was originally developed to screen for perinatal depression, but further assesses anxiety symptoms [2]. Below, we describe both screening questionnaires for PMAD where higher scores indicate greater severity of depression.

Patient Health Questionnaire (PHQ-9): The PHQ-9 consists of nine questions on a 4-point Likert scale (i.e., 'not at all,' 'several days,' 'more than half the days,' and 'nearly every day'). Aggregated scores range between $[0, 27]$.

Edinburgh Postnatal Depression Scale (EPDS-10): The EPDS consists of ten questions on a 4-point Likert scale regarding the frequency with which respondents experienced symptoms of depression (e.g., 'I have blamed myself unnecessarily when things go wrong'). Aggregated scores range between $[0, 30]$.

Target Variable: Scores from the PHQ-9 and the EPDS were dichotomized into 'low risk' (i.e., negative) or 'moderate to high risk' (i.e., positive) according to each scale's scoring criteria. Screening positive for depression risk was determined by endorsement of at least one of the following conditions being true: (1) suicidal ideation, (2) PHQ-9 ≥ 5, or (3) EPDS ≥ 8.

Table 3. Set of publicly available datasets used in this study.

name	#rows	#cols	target name	favorable label	sensitive attributes
heart_disease	303	13	target	1	{age }
student_math	395	32	g3_ge_10	1	{sex, age}
student_por	649	32	g3_ge_10	1	{sex, age}
creditg	1,000	20	class	good	{personal...*, age}
titanic	1,309	13	survived	1	{sex}
us_crime	1,994	102	crimegt70pct	0	{blackgt6pct}
compas_violent	4,020	51	two_year_recid	0	{sex, race}
nlsy	4,908	15	income96gt17	1	{age, gender}
compas	6,172	51	two_year_recid	0	{sex, race}

As recommended in Wong et al. [24], to address the issues related to class imbalance, random undersampling was performed on the training set; the validation and test sets were not modified.

4.3 Publicly Available Benchmark Datasets

To further validate our method, we evaluated it on a set of other open-access datasets for fairness evaluation from Hirzel & Feffer [13]. Here, we focus on the datasets with binary targets and select the top 9 datasets[1] when ordered by the number of rows. Table 3 describes the open-access datasets used in this work. Note that random undersampling was not performed on these datasets.

4.4 Hypervolume

We use hypervolume [8] to assess the quality of the Pareto fronts generated by each weighting method. Hypervolume summarizes each Pareto front into a single value by calculating the area of the objective space covered by each solution's performance within the front relative to a reference point. Figure 2 visualizes the hypervolume calculation for an example Pareto front optimized to minimize two objectives; the shaded area represents the hypervolume.

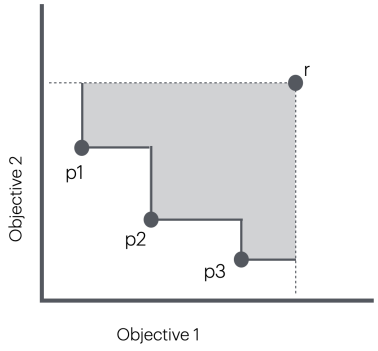

Fig. 2. Hypervolume for a given Pareto front comprised of three solution performances: $p1$, $p2$, and $p3$. The reference point (r) is used to calculate the hypervolume (shaded region) of the front.

[1]We skip the *ricci* dataset as most models produce a training AUROC score of 1.0 with it.

Pareto fronts with larger hypervolume are better for the following reasons. (a) Diversity [9]: A larger hypervolume indicates that the Pareto front contains solutions that cover a wider range of trade-offs between the objectives. This diversity offers more options for decision-makers. (b) Proximity [9]: A larger hypervolume often indicates that the Pareto front is closer to the true optimal front. This means that the solutions are both diverse and closer to optima. (c) Pareto-compliant [4]: Whenever the points on a Pareto front dominates the points on another Pareto front, the hypervolume of the former is more than that of the latter.

The procedure we use to compute the hypervolume is similar to the one described in La Cava [18]. For every dataset and every experimental condition combination, we launch 20 replicates as described earlier in Section 4. Then, for each replicate, we consider the Pareto front obtained from all the evaluated models in that run using their values on performance and fairness metrics on the test set, and calculate the hypervolume of that Pareto front. To ensure consistency in hypervolume computation, both metrics are converted into minimization objectives: $(1 - predictive\_metric)$ and $fairness\_metric$. A reference point of (1.0,1.0) is used for hypervolume calculation. We use these hypervolume values for all datasets and experimental conditions in the analysis in the following sections.

4.5 Statistical analysis

We conducted a Friedman test to detect significant differences in the hypervolume of the Pareto front between the weighting methods. If the Friedman test reported significant differences, we then performed a post-hoc paired Wilcoxon signed-rank test with a Bonferroni correction for multiple comparisons to identify differences among specific weighting method pairs. A significance level of 0.05 was used for all statistical tests.

4.6 Software availability

Our supplemental material is hosted on GitHub[2] and contains all files related to software, data analysis, figure visualization, and documentation for this work.

5 Results

By using the statistical tests in Section 4.5, for each dataset under consideration, we determine whether the evolved weights perform significantly better than other reweighting methods in terms of the hypervolume of the Pareto fronts (see Section 4.4) for different combinations of metrics. In Table 4, for each dataset that yielded significant differences according to the Friedman test, we present the results of the pairwise comparison between evolved weights and other methods using Wilcoxon rank-sum test.

We denote the experimental conditions as follows. EW, DW, and EQ refer to the Evolved Weights, Deterministic Weights, and Equal Weights reweighting methods, respectively. ACC and ROC represent the predictive metrics accuracy and area under the ROC curve (AUROC), while DPD and SFN correspond to the fairness metrics demographic parity difference and subgroup false negative rate, respectively. The last four columns in Table 4 list the combinations of predictive and fairness metrics used as experimental conditions. For example, (ACC, DPD) denotes the condition combining accuracy and demographic parity difference. These results are discussed in the following section.

Additionally, since the (ACC, DPD) experimental condition leads to the maximum number of datasets where evolved weights performed significantly better than other methods, we also plot the hypervolume values for all datasets and weighting methods for this condition in Figure 3. Each plot in the figure denotes the hypervolume of the Pareto front

[2]https://github.com/theaksaini/Comparing-Reweighting-Methods-arxiv

Table 4. Statistical significance of performance differences across datasets (EW: Evolved Weights, DT: Deterministic Weights, EQ: Equal Weights). EW>DT and EW>EQ indicate, respectively, that evolved weights yield significantly higher hypervolume than deterministic and equal weights (Wilcoxon rank-sum). Cells are populated only when the Friedman test detects significant differences (Sec. 4.5).

Dataset	(ACC, DPD)	(ACC, SFN)	(ROC, DPD)	(ROC, SFN)
heart_disease	EW>DT, EW>EQ		EW>DT, EW>EQ	EW>DT, EW>EQ
student_math	EW>DT, EW>EQ		EW>DT, EW>EQ	EW>DT, EW>EQ
student_por	EW>DT, EW>EQ		EW>DT, EW>EQ	EW>DT, EW>EQ
creditg	EW>DT, EW>EQ		EW>DT, EW>EQ	
titanic	EW>DT, EW>EQ	EW>DT, EW>EQ	EW>DT, EW>EQ	EW>DT, EW>EQ
us_crime	EW>DT, EW>EQ		EW>DT, EW>EQ	
compas_violent	EW>DT, EW>EQ		EW>DT, EW>EQ	
nlsy	EW>EQ		EW>EQ	
compas			EW>DT, EW>EQ	EW>DT, EW>EQ
pmad_phq	EW>DT, EW>EQ	EW>DT, EW>EQ		EW>DT
pmad_epds	EW>DT, EW>EQ	EW>DT, EW>EQ		

constructed from all the evaluated models in the corresponding dataset and weighting method. Note that we use the values of the metrics on the test set to compute the Pareto front.

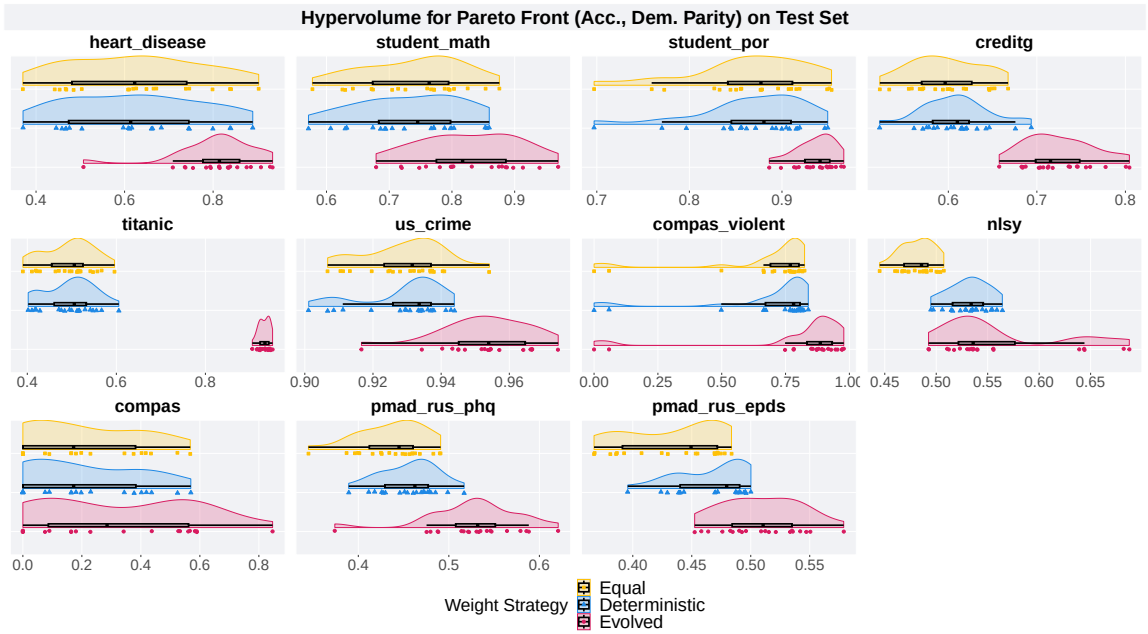

Fig. 3. Hypervolume for the Pareto fronts constructed from the performance on the test data for various datasets. The figure shows the experimental condition where accuracy and demographic parity difference have been used as the predictive and fairness metrics, respectively. Each point in the figure represents a single run.

6 Discussion

6.1 Evolution facilitates greater hypervolume

The results given in Table 4 demonstrate that for each of the datasets studied in this work, under at least one experimental condition, evolved weights lead to better optimization of predictive and fairness metrics than other reweighting methods: (1) equal weights, and (2) deterministic weights calculated solely based on the data characteristics as outlined in Kamiran & Calders [15]. In other words, the hypervolume of Pareto fronts obtained from models trained with evolved sample weights is higher than the hypervolume generated by alternative weighting methods for most of the combinations of dataset and experimental conditions. Our results are consistent with other works where sample weights evolved using evolutionary algorithms have been shown to perform better than other fairness-enhancing methods, such as game-theoretic intervention GerryFair [16], in optimizing fairness and predictive performance [18].

The success of the evolved weights approach is likely due to the more sophisticated search for optimal sample weights through the evolutionary mechanism, relative to deterministic weights calculated based on dataset characteristics. The GA presented here is designed to simultaneously optimize sample weights for both predictive performance and fairness, thereby optimizing a Pareto front. Additionally, the GA selects parents based on both predictive performance and fairness for a particular ML model, which results in evolved sample weights tailored for that model; alternative weighting methods are model-agnostic. As a result, these compounding effects contribute to the construction of Pareto fronts with greater hypervolume with the evolved weights approach compared to other methods.

One limitation of the genetic algorithm approach is that it requires multiple training calls to the model to evaluate the evolved sample weights, whereas other methods do not require as many calls. However, in circumstances where algorithmic fairness is paramount—such as healthcare or criminal justice—the reduction in algorithmic bias may justify the additional computational resources.

6.2 Benefits of Evolved Weights are influenced by Optimization Objectives

As shown in Table 4, the number of datasets for which Evolved Weights (EW) outperform the two baseline methods depends strongly on the choice of optimization objectives (metrics). Specifically, EW yields significantly better results on 9 datasets under the (ACC, DPD) objective pair, 3 datasets under (ACC, SFN), 8 datasets under (ROC, DPD), and 5 datasets under (ROC, SFN). This pattern indicates that the effectiveness of evolving sample weights varies across different predictive–fairness metric combinations.

One possible reason behind this disparity in the performance across various combinations of metrics is that for some combinations, it might be easier to improve one metric without considerably affecting the other metric, thereby improving the Pareto front and the corresponding hypervolume. In contrast, other metric pairs might impose stronger trade-offs, making simultaneous improvement more difficult.

The possibility of improvement in one metric without affecting the other is especially evident for fairness metrics such as demographic parity difference (DPD), which compare predictions *within* subgroups (e.g., acceptance rate) rather than *between* subgroups and the overall model performance. As a result, these metrics can sometimes be improved not by making the model more accurate or equitable, but by uniformly worsening the predicted outcomes for all subgroups, without affecting overall accuracy. This creates the appearance of fairness improvement even when subgroup-level predictions degrade.

The toy example in Figure 5 illustrates this effect. Consider a dataset with six instances from two groups, A and B, with true labels y and predicted labels $\hat{y}$. Before reweighting, the model's accuracy is 0.67. The acceptance rates are 0

Table 5. Toy Example

Group	Before Reweighting		After Reweighting	
	y	$\hat{y}$	y	$\hat{y}$
A	1	0	1	0
A	0	0	0	0
B	0	0	0	0
B	0	1	0	0
B	1	1	1	0
B	1	1	1	1

for group A and 0.75 for group B, giving a DPD of 0.75. After reweighting, the model's accuracy remains 0.67, but group B's acceptance rate falls to 0.25 while group A's remains at 0. This reduces the DPD to 0.25. From a metric standpoint, fairness has improved; however, the actual outcomes for both groups have become strictly worse. This highlights a broader issue: some fairness metrics can be improved through degenerate strategies that do not genuinely benefit the affected groups.

These observations suggest that the choice of optimization objectives plays a crucial role not only in how easily fairness can be improved, but also in what kind of improvements are encouraged by the metric definitions themselves. Understanding these interactions is essential when evaluating fairness-enhancing interventions such as reweighting.

Future work can look more deeply into the interactions between performance metrics (accuracy, auroc) and fairness metrics (demographic parity difference, subgroup false negative fairness) to determine which combinations are most compatible with reweighting-based methods, and to identify when metric improvements genuinely correspond to improved outcomes for all subgroups.

In addition to the objectives used during optimization, the performance of evolved weights may also depend on the particular parameter settings used in the GA configuration. Accordingly, future investigation could replicate our experiments using different ML models and explore alternative GA configurations, including adjustments to parameters such as the parent-selection strategy (e.g., replacing NSGA-II with lexicase selection [12]).

7 Conclusions

In this paper, we compared three approaches for assigning sample weights during the training of machine learning models. Such a process, known as reweighting, aims to improve model fairness by modulating the relative influence of different subgroups within a dataset in the loss function used by the model. Using eleven publicly available datasets (including two from the medical domain), we evaluated a Genetic Algorithm approach based on NSGA-II for evolving sample weights, termed Evolved Weights (EW), against two baseline strategies: Equal Weights (EQ), which assigns equal weights to all samples, and Deterministic Weights (DW), which computes weights directly from data characteristics without optimization. We evaluated these three reweighting strategies under four combinations of predictive and fairness metrics, using accuracy and area under the ROC curve (AUROC) as predictive metrics, and demographic parity difference and subgroup false negative fairness as fairness metrics. Pareto fronts were computed for each dataset in the corresponding predictive–fairness objective space. Our experiments demonstrate that Evolved Weights lead to more effective optimization of the trade-offs between fairness and predictive performance. Specifically, EW produced significantly better Pareto fronts for all the datasets under at least one combination of metrics. However, the magnitude

of these improvements—the number of datasets for which Evolved Weights (EW) outperform the two baseline methods—depends on the specific objectives (metrics) used during optimization. Overall, these findings suggest that evolutionary optimization of sample weights via NSGA-II can enhance fairness without substantial loss in predictive accuracy, while also highlighting the critical role of metric selection in determining fairness–performance outcomes. Future work should investigate additional metric combinations, analyze the mechanisms underlying metric-dependent improvements, and explore the generalizability of the EW approach across different model architectures and application domains.

Acknowledgments

We thank members of the Department of Computational Biomedicine at Cedars-Sinai Medical Center for their helpful comments on this work. Cedars-Sinai Medical Center provided computational resources through its High Performance Computing clusters. The work was supported by NIH grants R01 LM010098 and U01 AG066833.

References

[1] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2023. *Fairness and machine learning: Limitations and opportunities*. MIT Press.

[2] Evelien PM Brouwers, Anneloes L van Baar, and Victor JM Pop. 2001. Does the Edinburgh postnatal depression scale measure anxiety? *Journal of psychosomatic research* 51, 5 (2001), 659–663.

[3] Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. 2017. Optimized pre-processing for discrimination prevention. *Advances in neural information processing systems* 30 (2017).

[4] Yongtao Cao, Byran J Smucker, and Timothy J Robinson. 2015. On using the hypervolume indicator to compare Pareto fronts: Applications to multi-criteria optimal experimental design. *Journal of Statistical Planning and Inference* 160 (2015), 60–74.

[5] Richard J Chen, Judy J Wang, Drew FK Williamson, Tiffany Y Chen, Jana Lipkova, Ming Y Lu, Sharifa Sahai, and Faisal Mahmood. 2023. Algorithmic fairness in artificial intelligence for medicine and healthcare. *Nature biomedical engineering* 7, 6 (2023), 719–742.

[6] Carlos A Coello. 2000. An updated survey of GA-based multiobjective optimization techniques. *ACM Computing Surveys (CSUR)* 32, 2 (2000), 109–143.

[7] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.

[8] Carlos M Fonseca, Luís Paquete, and Manuel López-Ibánez. 2006. An improved dimension-sweep algorithm for the hypervolume indicator. In *2006 IEEE international conference on evolutionary computation*. IEEE, 1157–1163.

[9] Andreia P Guerreiro, Carlos M Fonseca, and Luís Paquete. 2020. The hypervolume indicator: Problems and algorithms. *arXiv preprint arXiv:2005.00515* (2020).

[10] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems* 29 (2016).

[11] Ahmad Hassanat, Khalid Almohammadi, Esra'a Alkafaween, Eman Abunawas, Awni Hammouri, and VB Surya Prasath. 2019. Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach. *Information* 10, 12 (2019), 390.

[12] Thomas Helmuth, Lee Spector, and James Matheson. 2014. Solving uncompromising problems with lexicase selection. *IEEE Transactions on Evolutionary Computation* 19, 5 (2014), 630–643.

[13] Martin Hirzel and Michael Feffer. 2023. A suite of fairness datasets for tabular classification. *arXiv preprint arXiv:2308.00133* (2023).

[14] Heinrich Jiang and Ofir Nachum. 2020. Identifying and correcting label bias in machine learning. In *International conference on artificial intelligence and statistics*. PMLR, 702–712.

[15] Faisal Kamiran and Toon Calders. 2012. Data preprocessing techniques for classification without discrimination. *Knowledge and information systems* 33, 1 (2012), 1–33.

[16] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2018. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International conference on machine learning*. 2564–2572.

[17] Nima Kordzadeh and Maryam Ghasemaghaei. 2022. Algorithmic bias: review, synthesis, and future research directions. *European Journal of Information Systems* 31, 3 (2022), 388–409.

[18] William G La Cava. 2023. Optimizing fairness tradeoffs in machine learning with multiobjective meta-models. In *Proceedings of the Genetic and Evolutionary Computation Conference*. New York, NY, USA, 511–519.

[19] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)* 54, 6 (2021), 1–35.

[20] Weiwen Miao. 2010. Did the results of promotion exams have a disparate impact on minorities? Using statistical evidence in ricci v. destefano. *Journal of Statistics Education* 18, 3 (2010).

[21] Melanie Mitchell. 1998. *An introduction to genetic algorithms*. MIT press.

[22] Yoonyoung Park, Jianying Hu, Moninder Singh, Issa Sylla, Irene Dankwa-Mullan, Eileen Koski, and Amar K. Das. 2021. Comparison of Methods to Reduce Bias From Clinical Prediction Models of Postpartum Depression. *JAMA Network Open* 4, 4 (April 2021), e213909. doi:10.1001/jamanetworkopen.2021.3909

[23] Oona Rainio, Jarmo Teuho, and Riku Klén. 2024. Evaluation metrics and statistical tests for machine learning. *Scientific Reports* 14, 1 (2024), 6086.

[24] Emily F Wong, Anil K Saini, Melissa Wong, Eynav E Accortt, Melissa S Wong, Jason H Moore, and Tiffani J Bright. 2024. Evaluating Bias-Mitigated Predictive Models of Perinatal Mood and Anxiety Disorders Across Racially Diverse Birthing Patients. *JAMA Network Open* (2024).

[25] Xiaoying Xing, Hongfu Liu, Chen Chen, and Jundong Li. 2021. Fairness-aware unsupervised feature selection. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3548–3552.

[26] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. 335–340.