Falcon: Fair Active Learning using Multi-armed Bandits

Ki Hyun Tae
kihyun.tae@kaist.ac.kr
KAIST
Daejeon, Republic of Korea

Hantian Zhang
hantian.zhang@gatech.edu
Georgia Institute of Technology
Atlanta, USA

Jaeyoung Park
jypark@kaist.ac.kr
KAIST
Daejeon, Republic of Korea

Kexin Rong
krong@gatech.edu
Georgia Institute of Technology
Atlanta, USA

Steven Euijong Whang*
swhang@kaist.ac.kr
KAIST
Daejeon, Republic of Korea

ABSTRACT

Biased data can lead to unfair machine learning models, highlighting the importance of embedding fairness at the beginning of data analysis, particularly during dataset curation and labeling. In response, we propose Falcon, a scalable fair active learning framework. Falcon adopts a data-centric approach that improves machine learning model fairness via strategic sample selection. Given a user-specified group fairness measure, Falcon identifies samples from "target groups" (e.g., (attribute=female, label=positive)) that are the most informative for improving fairness. However, a challenge arises since these target groups are defined using ground truth labels that are not available during sample selection. To handle this, we propose a novel trial-and-error method, where we postpone using a sample if the predicted label is different from the expected one and falls outside the target group. We also observe the trade-off that selecting more informative samples results in higher likelihood of postponing due to undesired label prediction, and the optimal balance varies per dataset. We capture the trade-off between informativeness and postpone rate as policies and propose to automatically select the best policy using adversarial multi-armed bandit methods, given their computational efficiency and theoretical guarantees. Experiments show that Falcon significantly outperforms existing fair active learning approaches in terms of fairness and accuracy and is more efficient. In particular, only Falcon supports a proper trade-off between accuracy and fairness where its maximum fairness score is 1.8–4.5x higher than the second-best results.

PVLDB Reference Format:
Ki Hyun Tae, Hantian Zhang, Jaeyoung Park, Kexin Rong, and Steven Euijong Whang. Falcon: Fair Active Learning using Multi-armed Bandits. PVLDB, 17(5): XXX-XXX, 2024.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/khtae8250/Falcon.

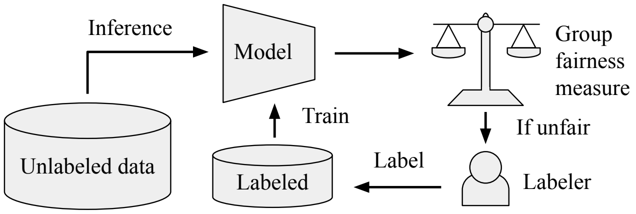


Figure 1: Fair active learning involves selecting samples that, when labeled, would enhance the fairness of a machine learning model according to a specific group fairness measure.

1 INTRODUCTION

AI fairness is becoming essential as AI is widely used and needs to be trustworthy. Critical applications of fairness include AI-based hiring, AI-based judging, self-driving cars, and more. Recognizing that biased data is often the source of unfairness or discrimination in the downstream machine learning model, this work adopts a data-centric approach to improve fairness, as supported by previous studies [30, 40, 66, 69]. Specifically, the data-centric approach mitigates unfairness in machine learning models by improving dataset curation and labeling, rather than improving model training.

Manual data labeling in supervised learning is an expensive process, so active learning frameworks [1, 29, 39, 48, 51–54] have been introduced to reduce the cost of data annotation. Traditional active learning focuses on selecting samples that lead to a maximal increase in model accuracy under a fixed labeling budget. This process of sample selection could worsen model fairness if not done carefully. For example, suppose there are two demographic groups men and women where it is desirable to have similar model accuracies for fairness, but women are currently under-represented. If no women samples are included during active learning, the accuracy disparities across groups might worsen as a result of data labeling. This scenario highlights the importance of incorporating fairness constraints during active learning to mitigate potential biases in the resulting model.

However, adding a fairness objective to the active learning framework is challenging for two reasons. First, while conventional fairness techniques can improve fairness by balancing training samples from different data subgroups [31, 32, 45, 46, 68], this labeled data is not available in the active learning setting. Hence, it is difficult to decide which samples are beneficial for fairness. Using pseudo-labels does not work well since active learning prioritizes uncertain samples, which tend to have inaccurate pseudo-labels. Second, as

*Corresponding author

Proceedings of the VLDB Endowment, Vol. 17, No. 5 ISSN 2150-8097.
doi:XX.XX/XXX.XX

observed in prior works [19, 41, 46, 70], the fairness objective is sometimes at odds with the accuracy objectives unless we are in ideal circumstances [23]. For example, if fairness means having the same positive prediction rates among two ethnic groups (referred to as demographic parity [26]), then a perfectly-accurate classifier will not guarantee the same positive prediction rates if one ethnic group has positive samples more frequently than the other. The trade-off between accuracy and fairness is increasingly becoming a critical decision to make in practice. For example, an IT company recently scrapped its AI recruiting tool when it discriminated women [21]. The problem was biased data where the majority of previous hires were men. A solution is to balance accuracy and fairness to reflect the increase of women in the workforce and ensure an unbiased evaluation, which may ultimately benefit the company.

In this work, we propose a fair active learning framework, Falcon, which accepts a group fairness measure and automatically learns policies for selecting samples that improve the fairness the most. A user can first specify custom or well-known group fairness measures [12] including the prominent measures demographic parity [26], equalized odds [28], equal opportunity [28], predictive parity [20], and equalized error rate [59]. Given the measure, Falcon identifies "target groups" defined using sensitive attribute values and labels. Falcon prioritizes getting labels for samples from the target groups to improve the trained model's accuracy for that group, which in turn improves the overall fairness. As a running example, suppose that we use demographic parity as the fairness measure, where it is desirable to have similar positive prediction rates across demographic groups (e.g., men versus women). Suppose that the female group has a lower positive prediction rate at the moment. Note that the target group that requires more labels might change over the course of the training as we acquire more labels. Falcon would attempt to get more samples from the target group `(attribute=female, label=positive)` to improve the positive prediction rate. We can generalize this approach to find target groups for any group fairness measure specified by the user.

To address the first challenge of identifying target groups in fair active learning when ground truth labels are not available, Falcon proposes a *trial-and-error method* for handling unknown labels efficiently. Instead of solely relying on traditional active learning measures like entropy or confidence that determine a sample's informativeness by its proximity to the decision boundary, Falcon adds a fairness objective to select samples from specific groups (e.g., the `(attribute=female, label=positive)` group). However without ground truth labels, we can not always identify samples in the target group. Our key observation is that adding more labels does not necessarily lead to improved fairness, and it is better to delay using samples that would negatively impact fairness. We thus use a trial-and-error approach where we select samples in a sensitive group to label, but delay using them in model training if they have undesired labels for that group. For example, Falcon selects an informative sample in the female group to label, but only includes it in model training if it has a positive label since adding negative labels in the female group worsens demographic parity.

In order to find the most informative samples for fairness, we introduce a policy selection framework on top of the trial-and-error method to optimize model fairness. Specifically, we observe that the more informative samples are, the more likely they are postponed due to undesired label predictions. We capture how aggressively we should select informative samples using a *policy*, and our solution is to learn the optimal policy using a multi-armed bandit (MAB) approach based on the rewards in terms of improved fairness. However, the policies are not independent of each other, and the rewards also vary as we label more data. As a result, the ideal policy depends on the dataset and the stage of the labeling process. In order to handle this complicated scenario, Falcon uses adversarial MAB methods to dynamically select the best sampling policy. Adversarial MABs provide a principled approach to selecting amongst competing strategies that share a limited set of resources and have strong theoretical bounds of regret. To further improve stability and performance, Falcon rewards the nearest policies to the chosen one, ensuring that the unknown best policy still receives rewards, even if it is not directly selected.

To address the second challenge of balancing accuracy and fairness objectives, we extend Falcon to improve accuracy by combining it with traditional active learning techniques where fair and accurate labeling are alternated probabilistically. Our approach does not require any modifications for other active learning methods and effectively controls an accuracy-fairness trade-off, as we show in the experiments. When there is no ambiguity, we refer to the combined version as Falcon as well.

In our experiments, we show that Falcon significantly outperforms various fair active learning baselines on real datasets in terms of model fairness and accuracy and is faster.

We summarize our contributions as follows:

- We propose Falcon, a fair active learning framework that selects samples to improve fairness and accuracy. Falcon (1) uses a novel labeling strategy where it first selects subgroups to label and handles unknown ground truth labels using a trial-and-error strategy; (2) automatically selects the best sampling policy using adversarial MABs; and (3) balances fairness and accuracy by alternating its selection with traditional active learning.
- Falcon is efficient by using MABs and requires much fewer model trainings than other fair active learning approaches.
- We empirically show that Falcon drastically outperforms fair active learning baselines w.r.t. fairness and accuracy and is faster.

2 BACKGROUND AND OVERVIEW

We focus on an active learning scenario where the labeling budget is limited, and we would like to improve both fairness and accuracy. We target any application where there are not enough labels and discrimination is a potential problem. In the following sections, we explain preliminaries (Section 2.1), define our problem (Section 2.2), and provide an overview of Falcon (Section 2.3). We focus on improving fairness for now and later discuss how to also improve accuracy in Section 5.

2.1 Preliminaries

In this work, we focus on a binary classification setting and assume a training dataset $D_{train} = \{x_i, z_i, y_i\}_{i=1}^{n}$ where x_i is a training sample, $z_i \in \mathbb{Z}$ is a sensitive attribute (e.g., gender), and y_i is its label having a value of 0 or 1. We also denote the unlabeled, validation, and test datasets as D_{un}, D_{val}, and D_{test}, respectively. A classifier h is trained on D_{train}, and its prediction on a test sample is $\hat{y} \in \{0, 1\}$.

Group fairness definitions. To illustrate the fairness issues we aim to address, we begin by defining group fairness. Group fairness ensures that a trained model h has equal or similar statistics across different sensitive groups. Here we list five representative fairness measures [62] as follows:

- Demographic Parity (DP) [26] is satisfied if a trained model has an equal positive prediction rate across sensitive groups.
$$\forall z_i, z_j \in \mathbb{Z}, p(\hat{y} = 1|z_i) \simeq p(\hat{y} = 1|z_j) \quad (1)$$
- Equalized Odds (ED) [28] is satisfied if a trained model has an equal accuracy across sensitive groups conditioned on the true label $\mathrm{y} \in \{0, 1\}$, i.e., having equal false positive rate (FPR) and false negative rate (FNR).
$$\forall z_i, z_j \in \mathbb{Z}, \mathrm{y} \in \{0, 1\}, p(\hat{y} = 1|y = \mathrm{y}, z_i) \simeq p(\hat{y} = 1|y = \mathrm{y}, z_j) \quad (2)$$
- Equal Opportunity (EO) [28] is a relaxed version of ED that only consider conditioning on $y = 1$, i.e., having equal FNR.
$$\forall z_i, z_j \in \mathbb{Z}, p(\hat{y} = 1|y = 1, z_i) \simeq p(\hat{y} = 1|y = 1, z_j) \quad (3)$$
- Predictive Parity (PP) [20] is satisfied if a trained model has an equal probability of having positive labels across sensitive groups conditioned on the model prediction $\hat{\mathrm{y}} \in \{0, 1\}$, i.e., having equal false omission rate (FOR) and false discovery rate (FDR).
$$\forall z_i, z_j \in \mathbb{Z}, \hat{\mathrm{y}} \in \{0, 1\}, p(y = 1|\hat{y} = \hat{\mathrm{y}}, z_i) \simeq p(y = 1|\hat{y} = \hat{\mathrm{y}}, z_j) \quad (4)$$
- Equalized Error Rate (EER) [59] is satisfied if a trained model has an equal classification error rate across sensitive groups.
$$\forall z_i, z_j \in \mathbb{Z}, p(\hat{y} \neq y|z_i) \simeq p(\hat{y} \neq y|z_j) \quad (5)$$

To evaluate the fairness of the trained model h, we define a fairness score as one minus the maximum fairness disparity [15] between any two sensitive groups on the unseen test set. In the extreme case, a fairness score of 1 indicates that the classifier is perfectly fair according to the given fairness measure.

Active learning. The goal of active learning (AL) is to minimally label samples while maximizing model accuracy. A standard approach in AL is to choose samples that have the lowest confidence or highest entropy. Intuitively, we would like the labeler to label the most challenging samples. Within AL research, there are several approaches on how to evaluate each sample: uncertainty-based [47, 55, 64], diversity-based [42, 50], and hybrid approaches [8, 24]. In comparison, fair active learning adds one more dimension of difficulty where we would like to also select samples that improve fairness. As a default, we assume batch active learning where a set of samples are selected for labeling.

Informativeness for Fairness. In traditional AL, one representative approach is to consider uncertain samples as informative, and choosing such samples improves overall accuracy the most. Informativeness is estimated using entropy or confidence. In our work, we define an analogous notion of *informativeness for fairness.* It is well-known that prominent group fairness measures can be expressed as a sum of the subgroup accuracies, where a subgroup is defined using a label and a sensitive attribute [46, 68] like (`attribute=female, label=positive`) (details are in Section 3.1). Based on this key insight, we define a sample to be informative for fairness if it can improve the accuracy of specific subgroups. However, estimating this information score accurately is challenging due to the lack of labels in unlabeled data. To address this problem, we propose a novel data-driven technique for identifying the most informative samples for fairness in Section 3.2.

2.2 Problem Definition

Our goal is to select samples to label for the purpose of improving fairness of a trained model. Given an unlabeled dataset D_{un}, a train dataset D_{train}, a validation set D_{val}, a loss function l_θ, a group fairness measure $F(\theta, D)$ that measures fairness when using the model parameters θ on the dataset D, a labeling process H that receives unlabeled data and returns labeled data (e.g., human labeling), and a labeling budget b at every round with a total budget of B, *fair active learning* (fair AL) solves the following optimization problem at each round of labeling:

$$\begin{aligned} \underset{S \subseteq D_{un}}{\arg\max} \quad & F(\underset{\theta}{\arg\min}\ l_\theta(D_{train} \cup H(S)), D_{val}) \\ \text{s.t.} \quad & |S| \leq b. \end{aligned}$$

Here, in each step, we want to find a set S with at most b samples, such that, after labeling S and adding them to the current training data, the model trained on this training data would achieve the highest fairness on the validation set.

In Section 5, we describe how to extend the above problem formulation to jointly optimize accuracy and fairness by combining our approach with traditional AL methods.

2.3 Falcon Overview

Figure 2 gives an overview of the Falcon framework. As an input, the user provides a group fairness measure. Then Falcon determines which "target" groups of samples need to be labeled first. This approach is more general than class imbalance works with fixed minority groups where the minority groups themselves may become majority groups as samples are labeled for the minority group. When selecting samples for a specific group, the selection itself may be undesired due to the lack of labels, so Falcon learns the right policy to select samples that are informative and yet are not likely to end up having undesired labels. Finally, we extend Falcon with traditional AL methods to improve model accuracy. When there is no ambiguity, we refer to the extended version as Falcon as well.

In the following sections, we introduce an effective trial-and-error labeling strategy (Section 3), propose automatic policy searching methods (Section 4), and present Falcon's algorithm (Section 5).

3 TRIAL-AND–ERROR LABELING STRATEGY

We discuss Falcon's labeling strategy to improve fairness where it first selects subgroups to label (Section 3.1) and handles unknown ground truth labels using a trial-and-error approach (Section 3.2).

3.1 Subgroup Labeling to Improve Fairness

Our key strategy for improving fairness is to increase the labeling of specific subgroups of the data. We take inspiration from the minibatch selection approaches OmniFair [68] and FairBatch [46], which adjust subgroup sampling ratios within a minibatch to improve fairness.

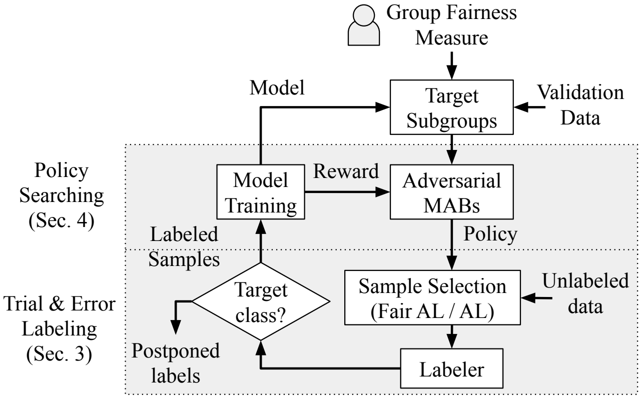


Figure 2: Overview of FALCON workflow.

For illustration purposes, we assume two sensitive groups (i.e., $\mathbb{Z} = \{0, 1\}$) for now, and discuss extensions to multiple sensitive groups later. Suppose the fairness metric is EO, which requires the trained model to have similar accuracies on two sensitive groups for positive samples. Suppose that one of the groups currently has a lower accuracy than the other and is thus underrepresented. Then providing more samples to the underrepresented group can enhance the model's performance on that group because the model assigns more weight to the underrepresented group samples when optimizing its objective function, naturally boosting accuracy.

Using an analysis in FairBatch [46], optimizing for EO can be formulated as a quasi-convex optimization problem, which intuitively means that increasing the underrepresented group's samples first improves EO and then does not from some point. This setup justifies the approach of increasing the labeling of a subgroup if it is underrepresented.

The subgroup decomposition analysis can extend to other popular group fairness metrics beyond EO. OmniFair [68] proposed a similar reweighting solution to improve group fairness as well. Based on this key observation, we can identify the target subgroups that need to be labeled in order to improve target fairness. We explain the subgroups to target when the fairness measures are DP and EO. The analysis for other group fairness metrics can be found in our technical report [60].

Example 1. Target Subgroups for DP. *For DP (Equation 1), let us assume that $p(\hat{y} = 1|z = 0) < p(\hat{y} = 1|z = 1)$ without loss of generality. We also know that*

$$\begin{aligned} p(\hat{y} = 1) &= p(y = 1, \hat{y} = 1) + p(y = 0, \hat{y} = 1) \\ &= p(y = 1)p(\hat{y} = 1|y = 1) + p(y = 0)p(\hat{y} = 1|y = 0) \\ &= p(y = 1)p(\hat{y} = 1|y = 1) + p(y = 0)(1 - p(\hat{y} = 0|y = 0)). \end{aligned}$$

Then $p(\hat{y} = 1|z = 0) < p(\hat{y} = 1|z = 1)$ can be rewritten as

$$\begin{aligned} &p(y = 1|z = 0)\underline{p(\hat{y} = 1|y = 1, z = 0)}+ \\ &p(y = 0|z = 0)(1 - p(\hat{y} = 0|y = 0, z = 0)) < \\ &p(y = 1|z = 0)p(\hat{y} = 1|y = 1, z = 1)+ \\ &p(y = 0|z = 1)(1 - \underline{p(\hat{y} = 0|y = 0, z = 1)}). \end{aligned}$$

Hence, we can see that improving $p(\hat{y} = 1|y = 1, z = 0)$ results in increasing $p(\hat{y} = 1|z = 0)$, while improving $p(\hat{y} = 0|y = 0, z = 1)$ results in decreasing $p(\hat{y} = 1|z = 1)$. Both strategies can reduce the disparity, so $(y = 1, z = 0)$ and $(y = 0, z = 1)$ subgroups should be targeted for additional labeling to improve the DP score.

Metric	Target Subgroups (y, z)
DP	$(1, z^*)$ or $(0, 1 - z^*)$
EO	$(1, z^*)$
ED	$(0, 1 - z^*)$, if FPR gap ≥ FNR gap $(1, z^*)$, otherwise
PP	$(0, 1 - z^*)$ or $(1, 1 - z^*)$, if FOR gap ≥ FDR gap $(0, z^*)$ or $(1, z^*)$, otherwise
EER	$(0, 1 - z^*)$ or $(1, 1 - z^*)$

Table 1: Target subgroups for each group fairness measure when a sensitive group $z^* \in \{0, 1\}$ has a lower fairness value.

Example 2. Target Subgroups for EO. *For EO (Equation 3), the goal is to close the gap between $p(\hat{y} = 1|y = 1, z = 0)$ and $p(\hat{y} = 1|y = 1, z = 1)$. Let us assume that the first term is smaller. Then, improving $p(\hat{y} = 1|y = 1, z = 0)$ directly improves fairness.*

Table 1 provides a summary of the target subgroups for each fairness measure when a sensitive group $z^* \in \{0, 1\}$ has a lower fairness value. For example, for DP, the target groups are $(1, z^*)$ or $(0, 1 - z^*)$ when $p(\hat{y} = 1|z = z^*) < p(\hat{y} = 1|z = 1 - z^*)$. For ED and PP, the target groups are determined by the subgroups that have a larger disparity gap.

Other Fairness Measures. In addition to the group fairness measures in Table 1, FALCON can support any group fairness measure that can be expressed as the subgroup accuracies. Similar to the previous examples, one can identify the target groups that have low fairness values and then perform more labeling on those groups.

Dynamic Target Group Selection. As we continue labeling data, the groups requiring more labels might change. This is a key difference from existing works that only focus on solving class imbalance, where the underrepresented group is fixed and contains fewer samples. Another complication arises from the inter-group influence, where labeling certain groups might positively or negatively impact the accuracy of other groups. The direction of this influence depends on the data; similar groups might positively affect each other, while different ones may have a negative impact. Although we don't model this influence directly, it justifies our strategy of *dynamically* selecting the appropriate groups to improve.

A key challenge is that ground truth labels are not available in an AL setting, making it difficult to determine whether an unlabeled sample belongs to the target groups. In the next section, we propose a simple yet effective solution, which explicitly labels samples and then uses only those with the label of interest.

3.2 Handling Unknown Ground Truth

The problem now shifts to selecting samples from the target groups when the labels are unavailable. A naïve approach for handling this issue is to generate pseudo labels [37] using model predictions and prioritize samples with higher informativeness. If the pseudo labels are perfect, then there would be no need to be concerned about using those samples at all. However, there is a fundamental limit to their correctness if they are informative for the trained model. Indeed, only samples with high confidence are likely to obtain correct pseudo labels.

Our solution is to select samples that are likely to be informative and label them, but postpone using them for training when they turn out to have undesirable labels. We refer to this strategy as trial-and-error labeling. This approach may sound counter-intuitive at first, given labeling is an expensive process. However, using samples with undesired labels can negatively affect fairness, so it is better not to use samples with undesired labels immediately. Later on, if we actually need these labels to further improve fairness or accuracy, we can use them. Here we show how trial-and-error labeling improves fairness when using DP and EO.

Example 3. Trial-and-error labeling for DP. *Continuing from Example 1 where* $p(\hat{y} = 1|z = 0) < p(\hat{y} = 1|z = 1)$*, we should obtain samples from* $(y = 1, z = 0)$ *or* $(y = 0, z = 1)$ *to improve DP. However, some samples can potentially be acquired from* $(\underline{y = 0}, z = 0)$ *or* $(\underline{y = 1}, z = 1)$ *with undesired labels. These samples directly worsen the DP gap by decreasing* $p(\hat{y} = 1|z = 0)$ *or increasing* $p(\hat{y} = 1|z = 1)$*. Hence, it is important to postpone using them to improve fairness.*

Example 4. Trial-and-error labeling for EO. *Continuing from Example 2, suppose the targeted group is set to* $(y = 1, z = 0)$ *to address the EO disparity, assuming that* $p(\hat{y} = 1|y = 1, z = 0) < p(\hat{y} = 1|y = 1, z = 1)$*. In this case, we can possibly obtain data from the* $(\underline{y = 0}, z = 0)$ *subgroup, which improves* $p(\hat{y} = 0|y = 0, z = 0)$*. Although training a model on these samples does not directly decrease* $p(\hat{y} = 1|y = 1, z = 0)$*, more training samples from* $(y = 1, z = 0)$ *can lead to overfitting the model's predictions for the* $(z = 0)$ *data towards* $\hat{y} = 0$*. As a result, there can be an "indirect" decrease in* $p(\hat{y} = 1|y = 1, z = 0)$ *in the end. Hence, it is also better to delay the usage of these samples to improve fairness.*

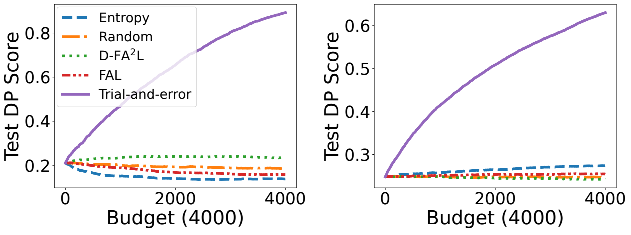
Informativeness for Fairness. The remaining question is which sample is the most "informative for fairness" when utilizing trial-and-error labeling. That is, the target sample should improve the specific subgroup's accuracy the most (and thus improve the overall fairness the most) and also have the label of interest.

Informativeness can be measured in several ways, and we first propose a simple solution based on an information theoretic approach using Shannon's entropy [55]. The information obtained by a sample labeled as x, $I_{entropy}(x)$, can be expressed as follows:

$$I_{entropy}(x) = p(+|x) \log \frac{1}{p(+|x)} + (1 - p(+|x)) \log \frac{1}{1 - p(+|x)}$$

where $p(+|x) = p(\hat{y} = 1|x)$ is the predicted probability of the sample x being labeled as 1 by the trained model. The first term means the expected information score when x is labeled 1, and the second term means the expected information score when the label is 0. $I_{entropy}$ is maximized when $p(+|x) = 0.5$.

Entropy is typically used to identify a sample that can improve overall accuracy, but it can also be adapted to improve fairness using the trial-and-error labeling strategy. Consider the example where the target subgroup is `(attribute=female, label=positive)`. In this case, we first select a sample that has the closest $p(+|x)$ value to 0.5 from the `(attribute=female)` group. We then include this sample in the training set only if its true label is positive. If there are multiple target groups like DP, we randomly select one of these groups for each iteration and apply the same approach. In Section 4.1, we generalize the notion of selecting samples with a desired label value as a *policy* to capture how much "risk" we are willing to make for finding samples with desired labels.



(a) TravelTime (DP)

(b) Employ (DP)

Figure 3: Comparing trial-and-error approach with the baselines on the TravelTime [22] and Employ [22] datasets where the target fairness is demographic parity (DP). Only the trial-and-error solution actually improves the DP score.

We now show that this simple trial-and-error approach surprisingly outperforms all fair AL baselines including two state-of-the-art algorithms *FAL* [6] and *D-FA*^{2}*L* [17]. We train a logistic regression model on two fairness datasets, TravelTime [22] and Employ [22]. For the *FAL* and *D-FA*^{2}*L* algorithms, we tune their hyperparameters to achieve the highest fairness score. Figure 3 shows the fairness results where the target fairness metric is DP. The x-axis is the labeling budget where 10 samples are selected for labeling in each iteration, and the y-axis is the fairness score on the test set. As a result, our simple solution significantly outperforms all the baselines and actually improves fairness while other baselines are not as effective, as we detail in Section 6.2. This result clearly demonstrates the importance of postponing undesired samples for improving fairness. For the TravelTime experiment in Figure 3a, it even delays about 2,900 samples out of the 4,000 labeling budget.

Trade-off between Informativeness and Postpone Rate. While the simple trial-and-error solution is effective, it is not general because it selects samples whose $p(+|x)$ is closest to a fixed threshold of 0.5. However, the bigger picture is that there are two competing factors: the more informative a sample is for improving the target group's accuracy, the less likely it has the target label. So depending on how we set this threshold, there is a trade-off between sample informativeness and postpone rate.

Consider the example in Figure 4 where the training data is divided into two groups by their classes, which are either positive or negative. Suppose we want to reduce the accuracy gap between the groups for fairness where the positive class has fewer samples and lower accuracy. Hence, we set the positive class as the target group. The goal is to select a sample from the target group that can effectively shift the decision boundary towards the negative class, which improves the accuracy of the positive class and thus the overall fairness. Comparing the unlabeled samples A and B, A has a larger impact on the decision boundary if it turns out to have a positive label and thus more informative, but also has a higher chance to be labeled negatively.

The remaining challenge is how to balance the informativeness of the selected samples and the risk of finding undesired labels, which we cover in the next section.

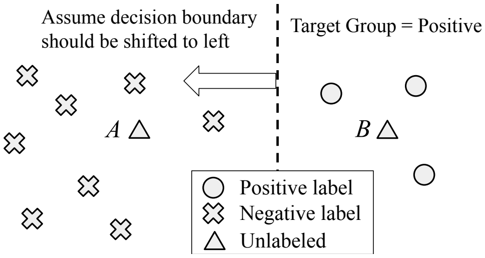


Figure 4: If the positive class is our target group, sample A increases the target group accuracy more than B if positively labeled and is thus more informative. However, A is less likely to have the desired target label, leading to a higher postpone rate. It is non-trivial to balance between these two factors.

4 AUTOMATIC POLICY SEARCHING

We now discuss how to find a desirable policy that balances informativeness and postpone rate. Ideally, we would like to analytically estimate the informativeness and postpone rate, but this is just as hard as determining the labels themselves. A more practical solution, therefore, is to utilize a data-driven method to identify the most effective policy. In this section, we introduce a multi-armed bandit based solution for selecting the best policy.

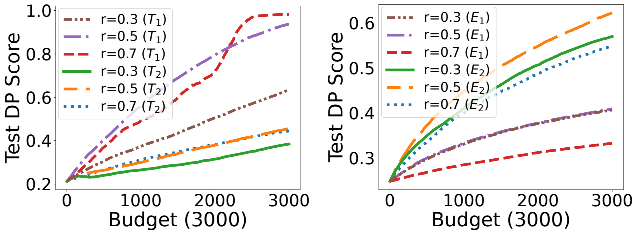
4.1 Policies for Sample Selection

Depending on the relative importance we assign to selecting informative samples versus samples with desirable labels, we can have different policies for sample selection. The more "risk" we are willing to take for finding an informative sample, the less likely it has the desired label. For each target group, we capture this risk taking as a policy, which is defined as follows:

Definition 1. ***Policy.*** *Given a target group (y, z) and a level of risk-taking c, a policy $r = c$ selects a sample x from the z group whose predicted probability for the label y, $p(\hat{y} = y|x)$, is closest to $1 - c$.*

For example, EO has one target group (e.g., (`attribute=female`, `label=positive`)), and we can use a policy set like $[r = 0.3, r = 0.5, r = 0.7]$, where $r = 0.7$ is a most risk-taking policy and selects a female sample whose predicted probability for a positive label is closest to 0.3. If we use DP, there are two target groups (e.g., (`attribute=female`, `label=positive`) and (`attribute= male`, `label=negative`)), so we consider twice the number of policies, e.g., $[r = 0.3, r = 0.5, r = 0.7]$ for each target group.

Choosing the right policy is non-trivial because the decision boundary may shift as more samples get labeled, which means the most effective policy may change as well. Furthermore, this outcome varies by dataset. To demonstrate these points, we compare the performances of individual policies using a policy set of $[r = 0.3, r = 0.5, r = 0.7]$ per target group. Figure 5 shows the fairness results on the TravelTime and Employ datasets where the target fairness is DP. For simplicity, we denote the i^{th} target subgroup for TravelTime and Employ as T_i and E_i, respectively. As a result, no single policy is always the best, and the best policy sometimes changes as we label more samples. For example, on the TravelTime dataset, $r = 0.5$ for the target group T_1 performs well at the early stages, but as more samples are labeled, $r = 0.7$ starts to perform better (Figure 5a). In comparison, $r = 0.5$ for E_2 consistently performs the best for Employ (Figure 5b). Another observation is that there is a significant difference in fairness improvement between the target groups. For both datasets, even the worst policy in the best target group outperforms the best policy in the other group. Hence, we cannot rely on a fixed strategy for finding the best policy and need an adaptive approach instead.



(a) TravelTime-DP **(b)** Employ-DP

Figure 5: Policy comparison using the TravelTime and Employ datasets where the i^{th} target groups are denoted as T_i and E_i, respectively. DP fairness is used. The best policy depends on the dataset and how much labeling has been done.

Our solution is to use a multi-armed bandit (MAB) approach, which balances exploration and exploitation to find the optimal policy based on the rewards in terms of improved fairness. We discuss details of the MAB-based approach in the next section.

4.2 Adversarial MAB for Policy Search

We utilize and extend existing multi-armed bandit (MAB) techniques to choose the best policy based on previous rewards. Using an MAB is a standard approach for allocating limited resources to competing choices (i.e., pulling arms) when the resulting rewards are only partially known [63]. In our setup, the competing choices are selecting the right policies where the reward is the fairness improvement, and the labeling effort is the limited resource. The key challenge is to balance exploration and exploitation where too much exploration of choices may lead to not utilizing the knowledge we already have, while too much exploitation may lead to missing opportunities of discovering better choices.

In particular, we use adversarial MABs [10, 14], which do not make assumptions about the reward distribution and only make choices to pull arms based on rewards. Traditional MABs [9] assume that the rewards follow a fixed and time-invariant distribution and provide theoretical bounds on regret, which is the expected difference between the sum of rewards in an optimal offline strategy versus the actual rewards obtained. In our setup, however, the fairness improvement changes as we label more samples, as shown in Figure 5. That is, the reward function does not follow statistical assumptions anymore. In addition, choosing one policy may have unpredictable effects on the rewards of using other policies. The reason is that using a policy results in actual data labeling, which improves the fairness of the current model and influences which samples to label for any other policy in the future. More specialized MABs like rotting bandits [38] make the assumption that rewards are independent and always decreasing, but this does not hold in our setup either. For example, if one subgroup is targeted and labeled more, then another subgroup's accuracy may actually decrease due to the influence. Then the next reward of the second subgroup can actually increase, as there is more opportunity to improve fairness. Since adversarial MABs make no assumption about the rewards, their behaviors are more conservative while having strong theoretical guarantees on regret bounds in the adversarial setup.

Although Falcon can be paired with any adversarial MAB, we choose the EXP3 [10] algorithm, which achieves an expected regret of $O(\sqrt{KTlnK})$ where K is the number of arms and T is the time horizon. Our choice is based on the fact that EXP3 is a representative method whose empirical performance is no worse than more recent MABs, which yield the same regret bounds with a smaller variance (referred to as high probability regret bound), as we detail in Section 6.8. Algorithm 1 shows how EXP3 learns the optimal policy for given rewards. At each time step, EXP3 chooses an arm according to the selection probability (Lines 3–4). This probability is a combination of the uniform distribution and another distribution that assigns probabilities to each action proportional to the exponential of the cumulative rewards for that action. Since some arms may later be useful, mixing in the uniform distribution ensures that the algorithm keeps on giving each arm a chance to be selected. In addition, the estimated gain is calculated by dividing the actual gain by the selection probability, which compensates for the reward of actions that are unlikely to be chosen (Line 7). EXP3 then updates the policies based on the rewards (Line 8) and repeats these steps for T iterations. Here the only assumption is that the reward value should be within the range of [0, 1] (Line 5).

For example, consider two policies P_1 and P_2, where P_1 initially yields high rewards, while P_2 starts with very low rewards but becomes more beneficial in later iterations. EXP3 begins increasing the probability of selecting P_1 but within a certain bound, mixing its probability with a uniform distribution. As a result, EXP3 still offers P_2 a chance to be chosen in later phases and will adapt the selection probabilities to changing rewards.

Efficiency. One advantage of the MAB-based approach is its high efficiency, as updating the MAB does not require model training. In contrast, other baselines need multiple model trainings for sample selection, which is not as efficient as Falcon (see Section 6.3).

4.3 Reward Design

The choice of the reward signal directly impacts the quality of MAB. We define the reward as the fairness improvement on the validation set. However, there are largely two challenges with the reward.

First, the reward obtained after labeling a few samples is usually very small, which has a negligible impact on updating policies. Hence, we propose a simple solution where we allocate the first L iterations to run the algorithm and then use the initial rewards to normalize the upcoming reward values. This normalization ensures that the reward has a more significant impact on updating the selection probabilities while still being within the [0, 1] range.

In addition, pulling one arm improves the fairness of the current model, but potentially limits the chances for other policies to be updated. The reason is that the total budget is limited, and the fairness improvement decreases as we label more data. If one sub-optimal policy is picked by chance in the early stages and receives a reasonable reward, the MAB may continue selecting that policy instead of searching for the optimal one. The dependency between policies makes it more challenging for the EXP3 algorithm to identify the best policy if it is not chosen sufficiently in previous iterations.

Algorithm 1: EXP3 algorithm [10].

```
  Input: Real $\gamma \in (0, 1]$, Arms $K$, Time horizon $T$
  Output: Updated probabilities
1 Initialize weights $w = 1$ ;
2 for $t = 1, 2, \ldots, T$ do
3     Set $p_i(t) = (1 - \gamma)\frac{w_i(t)}{\sum_{j=1}^{K} w_j(t)} + \frac{\gamma}{K}$;
4     Draw $i_t$ randomly accordingly to the probabilities
        $p_1(t), \ldots, p_K(t)$;
5     Receive reward $r_{i_t} \in [0, 1]$;
6     for $j = 1, \ldots, K$ do
7         $\hat{r_j}(t) = \begin{cases} \frac{r_j(t)}{p_{i_t}} & \text{if } j = i_t \\ 0, & \text{otherwise} \end{cases}$
8         $w_j(t+1) = w_j(t)exp(\gamma\hat{r_j}(t)/K)$ ;
```

We thus propose a reward propagating scheme that distributes the reward of a selected arm to its neighbors. This approach is inspired by a previous MAB-based data acquisition framework [18] that also assumes dependent arms. Intuitively, if two policies are close to each other (i.e., similar r values), their actual reward values tend to be similar. We thus propagate half of the obtained reward to the nearest policies, ensuring that the unknown best policy still receives some rewards even if it is not selected. Specifically, if policy P_i is selected and obtains a reward value of $\hat{r_i}(t)$ at time step t, we compute rewards for the remaining policies P_j as:

$$\hat{r_j}(t) = \begin{cases} \hat{r_i}(t) \times 0.5 & \text{if } P_j \in NN(P_i) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $NN(P_i)$ denotes the nearest policies to P_i that has the same target group. For example, suppose Falcon chooses $r = 0.5$ for (attribute=female, label=positive) and gets a reward of 1.0. We then assign a reward of 0.5 each to the $r = 0.4$ and $r = 0.6$ for the female group. We use this simple design, although one can propose other reward functions that capture a similar intuition.

5 FALCON ALGORITHM

We now describe how Falcon can be extended with traditional AL for selecting samples for the purpose of improving both fairness and accuracy. We explain the extension of Falcon with AL and present the overall algorithm.

Improving Both Fairness and Accuracy. Our approach is to alternate between fair and accurate labeling probabilistically. Specifically, we improve fairness with λ probability and accuracy with $1 - \lambda$ probability, where $\lambda \in [0, 1]$ is a hyperparameter that balances between fairness and accuracy. Here, a higher value of λ means better fairness. We can thus naturally integrate Falcon with other AL methods without requiring significant modifications. A similar blending idea has also been proposed in a fair adaptive sampling work [3], where the goal is to sample "labeled" data to improve accuracy and a specialized fairness metric called min-max fairness. A full analysis on how our MAB approach converges and deriving the regret bound in this setup is interesting future work, but we show extensive empirical results on how one can indeed balance fairness and accuracy by adjusting the interleaving degrees in Section 6.2.

Algorithm 2: Overall FALCON algorithm.

```
Input: Train data $D_{train}$, Validation data $D_{val}$, Unlabeled
       data $D_{un}$, Labeling budget $B$, Batch size $b$, Set of
       policies $P$, Fairness measure $F$, AL method $A$,
       Blending parameter $\lambda \in [0, 1]$
Output: Trained model $M$, Updated datasets $D_{train}$, $D_{un}$
1  $M$ = TrainModel($D_{train}$, $D_{val}$) ;
2  for $t = 1, 2, \ldots, \frac{B}{b}$ do
3      x ~ Bernoulli($\lambda$) ;
       // Improve fairness ($\lambda$) or accuracy (1-$\lambda$)
4      if $x = 0$ then
           // Improve accuracy using AL
5          $D_{train}, D_{un}$ = LabelData($D_{train}$, $D_{un}$, $M$, $A$, $b$) ;
6          $M_{new}$ = TrainModel($D_{train}$, $D_{val}$) ;
7      else
           // Improve fairness
8          $target\_groups$ = GetTargetGroups($M$, $D_{val}$, $F$);
9          $target\_MAB$ = GetMAB($target\_groups$, $P$) ;
10         $P_k$ = SelectPolicy($target\_MAB$) ;
11         $D_{train}, D_{un}$ = LabelData($D_{train}$, $D_{un}$, $M$, $P_k$, $b$) ;
12         $M_{new}$ = TrainModel($D_{train}$, $D_{val}$) ;
13         $R$ = GetReward($M_{new}$, $M$, $D_{val}$, $F$, $k$) ;
14         UpdateMAB($target\_MAB$, $R$) ;
15     $M = M_{new}$
```

FALCON Algorithm with AL. We now present the FALCON algorithm including AL in Algorithm 2. We have a fixed labeling budget of B where b samples are selected for labeling per iteration. Within each iteration, we first decide the labeling strategy to use based on the λ probability (Line 3). If we perform labeling for the purpose of accuracy, we run the given AL algorithm (Lines 5–6). Otherwise, we improve fairness. We first determine which subgroups to label for improving the target fairness metric F (Line 8) and construct an MAB for those target groups (Line 9). Then, the MAB selects a policy for labeling b samples and obtains a reward (Section 4.3), which measures the fairness improvement after labeling samples using the chosen policy (Lines 10–13). The reward is used to update the MAB (Line 14). We repeat these steps until we run out of budget.

Running Time Analysis. The primary cost in FALCON is retraining the model with newly labeled data. This process requires $\frac{B}{b}$ number of model training. Such retraining is a standard requirement in any active learning technique. Other components in FALCON include identifying target subgroups, selecting samples based on policies, and updating the MAB, only incur a small overhead (see Section 6.3).

Choice of Policy Set. We discuss how we select the policy set for FALCON. Overall, the more candidate policies, the more likely there is an optimal policy within them. However, using an infinite number of policies is not practical because the labeling budget is limited. Even if we do have an infinite budget, we would have to use infinitely many-armed bandits [33, 65] or Bayesian optimization [58] for modeling continuous policies, but they are not designed for adversarial rewards. We thus need to select a reasonable number of policies that are not extreme and use $[r = 0.3, r = 0.4, r = 0.5, r = 0.6, r = 0.7]$ as a default policy set throughout the paper. We discuss the effect of the number of policies in Section 6.6.

Dataset	$\lvert D_{train} \rvert / \lvert D_{un} \rvert / \lvert D_{test} \rvert / \lvert D_{val} \rvert$	Sen. Attr	Batch	B
TravelTime	2,446/48,940/24,470/2,446	gender	10	4K
Employ	5,432/162,960/81,480/5,432	disability	10	4K
Income	3,188/63,760/31,880/3,188	race	10	4K
COMPAS	294/2,356/1,178/294	gender	1	200

Table 2: Parameters for the four datasets.

Batch Size Effect. We discuss the effect of the batch size on FALCON's performance. The larger the batch size, the better reward signals we can utilize. On the other hand, there are fewer chances to adjust the policies based on the rewards. If labeling a sample only has a small effect in improving the model performance, we prefer using a larger batch size. In our experiments, we use larger batch sizes on larger datasets for efficiency. There are also other MABs [25] that are designed for handling batches. However, a technical hurdle is that we need to know the rewards of individual actions for each batch. The rewards are not readily available in our setup, as we only have an aggregated reward per batch. Investigating how to utilize batch-specialized MABs is an interesting future work.

Multiple Sensitive Attributes. While the previous examples assumed binary sensitive attributes, FALCON can be readily extended to support multiple sensitive groups (i.e., $\mathbb{Z} = \{0, 1, ..., n_z - 1\}$). Again, the fairness score is computed as one minus the maximum disparity value among any sensitive groups, as we explain in Section 2.1. Hence, we can construct an MAB based on the two groups that currently show the highest disparity value and continue to follow the same procedure. If the target group pair changes during the labeling process, we can simply switch the target MAB accordingly.

6 EXPERIMENTS

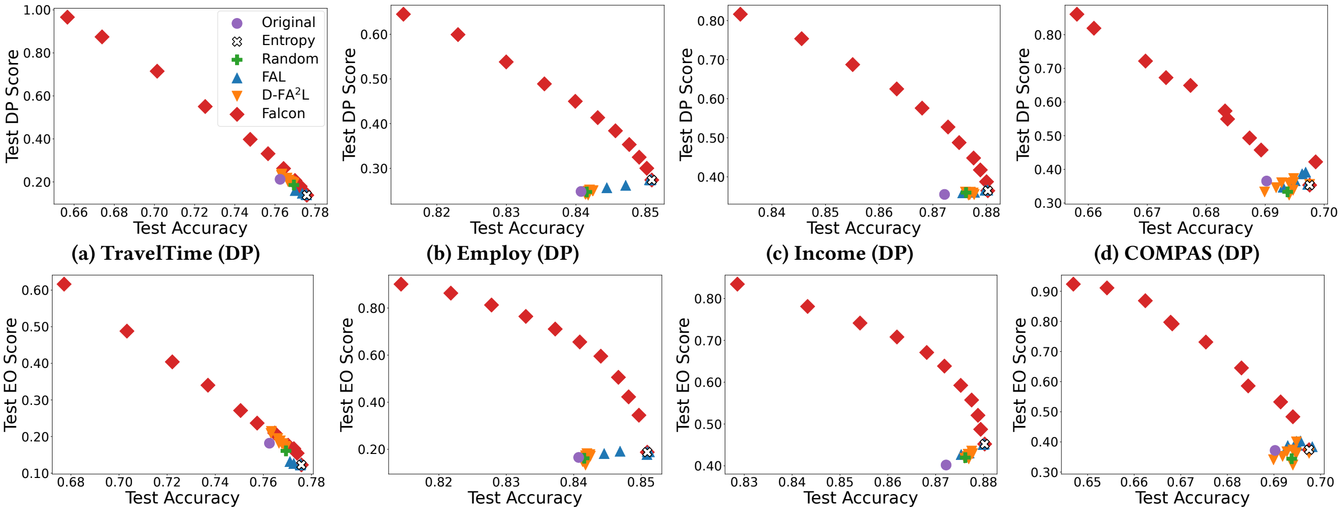
In this section, we evaluate FALCON on real datasets and address the following key questions: (1) How does FALCON compare with the baselines in terms of model accuracy, fairness, and running time? (2) Does FALCON find the best policy and perform in various scenarios? and (3) How useful is each component of FALCON?

We implement FALCON in Python, use Scikit-learn [43] for model training, and run all experiments on Intel Xeon Silver 4210R CPUs using ten different random seeds.

6.1 Setting

Datasets. We use four popular datasets in the fairness literature. For the first three datasets, we use the same feature pre-processing as in the Folktables [22] package, and for the COMPAS dataset, we apply the method provided by IBM's AI Fairness 360 toolkit [13]. For a detailed evaluation, we consider various sensitive attributes, including scenarios with multiple sensitive attributes.

- TravelTime [22]: Used to predict whether an employee has a commute to work that is longer than 20 minutes. We use `gender` as the sensitive attribute.
- Employ [22]: Used to predict whether an individual is employed. We use `disability` as the sensitive attribute.
- Income [22]: Used to predict whether an individual's income exceeds $50K per year. We use `race` as the sensitive attribute and consider three distinct groups: White, Asian, and Others.



(a) TravelTime (DP) **(b) Employ (DP)** **(c) Income (DP)** **(d) COMPAS (DP)**

(e) TravelTime (EO) **(f) Employ (EO)** **(g) Income (EO)** **(h) COMPAS (EO)**

Figure 6: Accuracy-fairness trade-off results on the four datasets and two fairness measures. In addition to the baselines, we add the result of model training without labeling any additional data and call it "Original." As a result, only Falcon significantly improves fairness and shows clear accuracy and fairness trade-offs. Note that Falcon using $\lambda = 0$ is the same as *Entropy*.

- COMPAS [7]: Used to predict an individual's criminal recidivism risk. We use gender as the sensitive attribute.

For each dataset, we split the dataset into training, test, unlabeled, and validation sets, as shown in Table 2. In addition, we use a total labeling budget of $B = 4K$ except for the COMPAS dataset where $B = 200$ is already enough to obtain high fairness. See our technical report [60] for detailed configurations.

Fairness evaluation. We consider five group fairness measures in Section 2.1. To quantify fairness, we define a fairness score as one minus the maximum fairness disparity [15] across any sensitive groups on the test set. A higher score indicates better fairness, and we provide the detailed equations in our technical report [60].

Parameters. We assume a batch AL setup and use different default batch sizes for the datasets as shown in Table 2. We use a default policy set of $[r = 0.3, r = 0.4, r = 0.5, r = 0.6, r = 0.7]$. We investigate the impact of batch sizes and the number of policies in Section 6.5 and Section 6.6, respectively.

Baselines Compared. We compare Falcon with existing fair AL and AL algorithms.

- *Entropy* [55]: A standard AL algorithm that selects the most uncertain samples based on entropy.
- *Random*: A randomized algorithm that uniformly selects unlabeled data samples.
- *FAL* [6]: The first fairness-aware AL algorithm that optimizes both group fairness and accuracy. *FAL* selects the top m points with the highest entropy value and then chooses samples that also have the maximum expected reduction in unfairness. A higher m favors better fairness.
- *D-FA^{2}L* [17]: A disagreement-based fairness-aware AL algorithm. *D-FA^{2}L* selects samples for which the decoupled models, trained separately on different sensitive groups, provide different predictions. *D-FA^{2}L*'s primary goal is to improve DP, but we also consider other fairness measures for a detailed comparison.

For the fair AL baselines, there are hyperparameters that can control an accuracy-fairness trade-off. We start with the default hyperparameters as described in the original papers, and tune them to provide the best results (see our technical report [60] for details).

Model Setup. We use logistic regression (LR) and neural network (NN) models. For NN, we use a multi-layer perceptron with one hidden layer consisting of 10 nodes. We tune the model hyperparameters such that the trained model has the highest validation accuracy. More detailed settings are in our technical report [60].

6.2 Accuracy and Fairness

We compare the accuracy and fairness results of Falcon with the other baselines using the four datasets. In the main experiments, we use demographic parity (DP) and equal opportunity (EO). The results for other fairness metrics are similar and can be found in our technical report [60]. Figure 6 shows the trade-off results with logistic regression models where the x-axis is the accuracy, and the y-axis is the fairness score on the test set. *Original* is where we train a model on the original data without performing labeling. For *FAL* and *D-FA^{2}L*, we employ 5 and 9 different sets of hyperparameters, respectively. For Falcon, we use 11 different λ values ranging from 0.0 to 1.0. As a result, Falcon shows the best accuracy and fairness trade-off compared to the baselines, which have noisy and even overlapping results for different hyperparameters. Concretely, Falcon improves the fairness score by up to 0.81 with up to 0.12 decrease in test accuracy. Notice that only Falcon is able to obtain high fairness when needed, whereas the baselines cannot. If the accuracy needs to be improved more than fairness, then one can simply lower the blending parameter λ so that AL is invoked more frequently. The results for a neural network are in our technical report [60], and the results are similar to Figure 6 where Falcon shows a better trade-off than the baselines.

Table 3 provides a more detailed comparison with the fair AL algorithms. We report the maximum fairness score that each method

Datasets	Fairness	Max. Fairness Score			
		Original	FAL	D-FA^{2}L	Falcon
TravelTime	DP	0.212	0.160	0.237	**0.966**
	EO	0.182	0.132	0.214	**0.616**
Employ	DP	0.248	0.275	0.252	**0.645**
	EO	0.165	0.192	0.181	**0.901**
Income	DP	0.355	0.366	0.361	**0.816**
	EO	0.402	0.453	0.435	**0.834**
COMPAS	DP	0.365	0.392	0.373	**0.861**
	EO	0.372	0.403	0.400	**0.924**

Table 3: Detailed fairness comparison of methods by tuning their hyperparameters to achieve the highest fairness scores.

Datasets	Avg. Running time (sec)				
	Entropy	Random	FAL	D-FA^{2}L	Falcon
TravelTime	139	91	1,420	179	**126**
Employ	114	76	1,411	140	**98**
Income	244	149	1,965	290	**205**
COMPAS	6.1	5.5	153	12	**5.9**

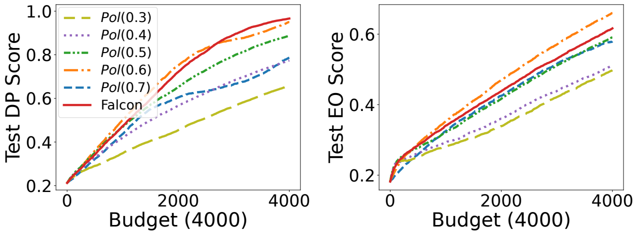
Table 4: Running time comparison of all methods on the four datasets using DP fairness. For each method, we show the average running time for all experiments in Figure 6.

can achieve when using the entire labeling budget. Note that Falcon is the only method where fairness actually improves, while the baselines do not show significant improvements in fairness compared to *Original* and, in some cases, even have worse fairness results. For all the scenarios considered, Falcon's maximum fairness score is 1.8–4.5x higher than the second-best results. The baselines do not perform well because they do not attempt to predict the actual label values and use them even if they have undesired labels. Thus, postponing undesired labels is critical, and it is important to find samples that are informative for fairness.

6.3 Running Time

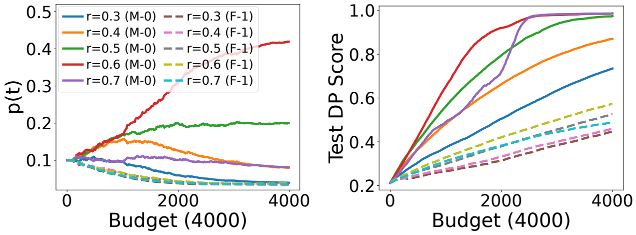
We compare the running time of Falcon with the baselines in Table 4. For each method, we show the average end-to-end running time for all experiments in Figure 6 where the target fairness metric is DP. The end-to-end labeling process consists of model training, identifying target groups, and selecting samples. While the first two steps take similar amounts of time, the sample selection is where there are time differences. As expected, *Random* is the fastest since there is no cost for selecting samples. For all the datasets, Falcon is 1.4–10x faster than the other fair AL algorithms because Falcon does not require additional model trainings for sample selection. In contrast, *FAL* computes the expected fairness reduction over all possible labels for the top m uncertain unlabeled samples, which requires $2 \times m$ additional model trainings. In addition, *D-FA^{2}L* retrains the model for each sensitive group to find samples that receive conflicting predictions.

Another interesting observation is that Falcon even performs better than *Entropy*. While *Entropy* needs to calculate entropy for all unlabeled samples, Falcon computes the predicted probability for the target group only. For example, if the target subgroup is (`attribute=female, label=positive`), then Falcon focuses on the female group instead of the entire dataset. This result indicates that the cost of managing the MAB is not significant compared to the overall labeling process, and that Falcon is practically efficient.



(a) Test DP Score **(b) Test EO score**

Figure 7: Fairness comparison of Falcon against a set of single policy baselines on the TravelTime dataset.



(a) Policy Selection Probability **(b) All Single Policies**

Figure 8: A detailed analysis for Figure 7a. (a) Falcon increases the selection probability of $r = 0.6$ for the subgroup $(M-0)$, where we denote the sensitive attribute (Male or Female) and label of the target subgroup in parentheses. (b) Fairness improvements for all single policies. The policy $r = 0.6$ for $(M-0)$ is the most effective in improving fairness.

6.4 Automatic Policy Search

We compare Falcon with a set of single policy baselines to show its ability to identify the most efficient policy during the labeling process. Here a baseline method $Pol(c)$ for policy $r = c$ randomly selects a target group (if there are more than one) for each round and selects a sample whose probability for the desirable label is closest to $1-c$. Figure 7 makes a comparison in terms of improved fairness on the TravelTime dataset where we use DP and EO. As a result, Falcon performs the best when using DP and the second best when using EO. Achieving the second-best performance is reasonable because Falcon has no prior knowledge of optimal policy and needs to allocate its budget to explore all policies.

We analyze Figure 7a in more detail where we show how Falcon updates its policies. Figure 8a shows the selection probabilities of the policies over labeling iterations where the target group for each policy is indicated in parentheses. Here, we have two target subgroups, (`attribute=female, label=positive`) and (`attribute=male, label=negative`), denoted as $(F-1)$ and $(M-0)$, respectively. Initially, Falcon randomly selects a policy, but starts to update the selection probabilities based on the obtained rewards. In the end, Falcon increases the probability for $r = 0.6$ for $(M-0)$ the most. Figure 8b shows the performance of all individual

Datasets	Test DP Score				
	b = 1	b = 10	b = 20	b = 50	b = 100
TravelTime	**0.970**	0.966	0.964	0.939	0.928
Employ	0.588	**0.645**	0.639	0.631	0.620
Income	0.808	**0.816**	0.815	0.807	0.803
COMPAS	**0.861**	0.840	0.812	0.828	0.830

Table 5: Batch size impact on Falcon.

Datasets	Fairness	Fairness Score			
		|P| = 2	|P| = 3	|P| = 5	|P| = 9
TravelTime	DP	0.956	**0.968**	0.966	0.944
	EO	0.583	0.610	**0.616**	0.606
Employ	DP	0.633	**0.645**	**0.645**	0.638
	EO	0.896	0.901	0.901	**0.902**

Table 6: Varying the number of policies.

policies. We observe that $r = 0.6$ for the subgroup $(M - 0)$ is the most effective policy, which is consistent with Falcon's findings. In comparison, the baseline $Pol(c)$ has a fundamental limitation as it relies on a fixed $r = c$ value for every target group.

We also perform the above experiments on the Employ dataset, and the key trends are similar where Falcon correctly updates the policies and achieves the best or second-best performance among the policies (see our technical report [60]).

6.5 Varying Batch Size

We evaluate Falcon when varying the batch size b from 1 to 100 when using DP fairness. Table 5 shows the DP scores of different batch sizes for the four datasets. The overall trend is that a larger batch size results in less runtime because we can reduce the number of sample selection iterations, and the reward of each batch becomes more substantial. At the same time, a batch size that is too large makes it difficult to find the optimal policy due to fewer chances to update the MAB given a limited labeling budget. In our results, the best batch size depends on the datasets. For the Employ and Income datasets, a batch size of 10 is the best. For the TravelTime and COMPAS datasets, a batch size of 1 is the best where each sample has enough impact on fairness improvement, so the MAB can be updated more frequently with a small batch size. For TravelTime, however, we use a batch size of 10 in the other experiments because Falcon runs much faster without sacrificing fairness significantly.

6.6 Varying Policies

We vary the number of policies ($|P|$) by adjusting the spacing between the r values of the two nearest policies. As discussed in Section 5, we set the lower and upper bounds of the r value to 0.3 and 0.7, respectively, to ensure that the policy set does not include extreme policies for better results. So if we use two policies, the policy set is $[r = 0.3, r = 0.7]$, and for five policies, it is $[r = 0.3, r = 0.4, r = 0.5, r = 0.6, r = 0.7]$, which is our default set.

Table 6 shows the fairness results when varying the number of policies in the range of $[2, 9]$ on the TravelTime and Employ datasets. As a result, using 3 or 5 policies usually yields the best performance compared to other cases. This result is expected because increasing policies initially helps to find good ones, but having too many makes it more difficult to find the good ones using a limited labeling budget. For the Employ dataset and EO, there is no significant difference between the last three options.

	TravelTime		Employ	
Method	DP	EO	DP	EO
Original	0.212	0.182	0.248	0.165
Falcon w/o trial-and-error & MAB	0.138	0.123	0.274	0.188
Falcon w/o MAB	0.887	0.591	0.630	0.900
Falcon w/o reward norm. & propag.	0.863	0.602	0.614	0.900
Falcon w/o reward propag.	0.959	0.611	0.635	0.899
Falcon	**0.966**	**0.616**	**0.645**	**0.901**

Table 7: Ablation study of Falcon.

Datasets	Fair.	Fairness Score				
		FAL	FAL$^+$	D-FA^{2}L	D-FA^{2}L$^+$	Falcon
TravelTime	DP	0.160	0.957	0.237	0.867	**0.966**
	EO	0.132	0.300	0.214	0.352	**0.616**
Employ	DP	0.275	0.500	0.252	0.486	**0.645**
	EO	0.192	0.727	0.181	0.538	**0.901**

Table 8: Comparison of Falcon against fair AL baselines combined with trial-and-error.

We perform additional experiments using more diverse policy sets to further investigate their impact on Falcon in our technical report [60]. As a result, the key trends are similar to Table 6 where our default set outperforms others.

6.7 Ablation Study

In Table 7, we perform an ablation study to investigate the effectiveness of each component in Falcon using the TravelTime and Employ datasets. We consider the ablation scenarios of removing reward propagation, reward normalization, MAB, and trial-and-error, in that order cumulatively. As a result, each ablation scenario leads to worse fairness. Not propagating rewards worsens performance by reducing the chance to find the best policy. Not normalizing the rewards leads to fairness improvements that are not large enough to make a difference in the policy selection. Not using MAB is equivalent to using the simple trial-and-error algorithm in Section 3.2 where we can no longer search policies dynamically. The benefit of using MABs is relatively smaller than that of trial-and-error, but is still significant as shown in Figure 7a where Falcon can achieve similar fairness as Falcon without MABs while saving up to 25% of the labeling budget. Finally, not using the trial-and-error strategy is equivalent to the *Entropy* method, which sometimes performs worse than *Original*. Thus, all functionalities are necessary.

To better understand the effectiveness of trial-and-error labeling, we also extend fair AL baselines with trial-and-error to see how they perform compared to Falcon. Table 8 shows the fairness scores of different methods, where we combine *FAL* and *D-FA^{2}L* with trial-and-error and refer to them as *FAL$^+$* and *D-FA^{2}L$^+$*, respectively. As a result, we observe trial-and-error sampling significantly improves the fairness of baselines. However, Falcon still consistently outperforms all baselines, which highlights that the MAB components of

Datasets	Fairness	Fairness Score			
		Original	**EXP3**	**EXP3-IX**	**EXP4.P**
TravelTime	DP	0.212	0.966	**0.970**	0.936
	EO	0.182	**0.616**	0.610	0.609
Employ	DP	0.248	0.645	**0.651**	0.628
	EO	0.165	0.901	0.901	**0.902**

Table 9: Fairness of Falcon with other adversarial MABs.

Falcon are also important for optimizing fairness. The results for other datasets are similar and shown in our technical report [60].

6.8 Other Adversarial MAB Algorithms

We now evaluate the empirical performance of EXP3 compared to other adversarial MABs, EXP3-IX [36] and EXP4.P [14], which are designed to achieve a high probability regret bound by reducing the variance of EXP3. Table 9 below shows the fairness results when using different adversarial MAB methods on the TravelTime and Employ datasets. The results show that there is no single best algorithm, and EXP3 empirically works as well as other methods. Even if EXP3 is not always the best, its fairness score is usually very close to the best score. We make similar observations in other datasets as well (see our technical report [60]).

7 RELATED WORK

Data-centric AI. Data-centric AI techniques boost various ML performances by improving the training data through better data management [27, 34, 44]. Our work falls into the category of data-centric AI for better model fairness [35, 49, 68, 69] by proposing a novel data labeling mechanism.

AI Fairness. Conventional fairness techniques for fair training can be largely categorized into pre-processing, in-processing, and post-processing techniques. Falcon can be categorized as a pre-processing approach and assumes that data labels are not available. Related works are OmniFair [68] and FairBatch [46], which improve group fairness using sample weighting. In comparison, Falcon focuses on the data labeling problem, which has quite different issues as the ground truth labels are unavailable. In this setup, trial-and-error sampling allows us to apply the subgroup sampling strategy. However, prioritizing only fairness-informative samples results in excessive postponing of undesired samples, wasting the limited labeling budget. To navigate this trade-off, Falcon leverages adversarial MABs, effectively balancing the informativeness of samples and the postpone rate by dynamically selecting various policies.

Fair Active Learning. A recent line of work addresses the fair active learning problem. FAL [6] extends conventional active learning to estimate the resulting fairness of selecting a sample. The estimation is a probabilistic analysis assuming that the label can have any of the possible values with equal probability. PANDA [56] uses reinforcement learning for the selection, but this can be expensive. Another related line of work is resolving class imbalance [4, 5]. However, an implicit assumption here is that there is a fixed under-represented group whose model accuracy needs to be improved. In contrast, improving fairness is more complicated where the groups that need to be improved may change as we label more data.

Fair Adaptive Sampling. Another line of research is sampling training data for the purpose of improving minimax fairness [2, 57], which takes samples from the group that has the worst model's accuracy. Here the group is defined only as a combination of sensitive attributes. In comparison, Falcon supports any group fairness measure beyond minimax fairness and addresses the more challenging problem where the target group can be defined using labels.

Active Learning. Active learning [1, 39, 48, 51–54] has been studied for decades for efficient labeling. A standard approach is sampling based on uncertainty, e.g., least confidence [64] or highest entropy [55]. Diversity-based methods have also been proposed to find representative samples using a clustering [42] or coreset selection [50]. Hybrid approaches [8, 24] that combine both criteria have been studied as well. A conceptually close work to our framework is active learning by learning [29], which selects the best labeling policy among the set of predefined AL algorithms. In comparison, we solve the new problem of improving fairness in active learning by learning the best self-generated policies.

Adversarial MAB. Adversarial MABs are used to choose policies only based on rewards where the rewards can be any value. As explained in Section 4.2, EXP3 [10] chooses arms probabilistically and updates the probabilities based on the rewards. EXP3-IX [36] reduces the variance of EXP3 at the price of introducing bias. EXP4.P [14] improves the performance by accepting expert advice on which arms are more promising. MABs have recently been widely used in data-centric AI works like data charging [18] and entity augmentation [16], as well as in visualization recommendation [61] and video database management systems [11]. Falcon can be compatible with any adversarial MAB algorithm.

8 CONCLUSION

We propose the fair active learning framework Falcon, which selects labels to improve model fairness and accuracy by learning policies. Falcon determines which groups of data need to be labeled first for better fairness and uses a trial-and-error strategy to find labeled data of the groups. Falcon also learns policies to determine how much risk to take in selecting fairness-informative samples, which can be done with adversarial MAB methods. The sample selection for fairness is blended with active learning methods to also improve accuracy. Falcon is efficient and requires much fewer model trainings than other fair active learning approaches. Experiments show how Falcon drastically outperforms the state-of-the-art fair active learning baselines on real benchmark datasets. Falcon is the only method that supports a proper trade-off between fairness and accuracy where its maximum fairness score is 1.8–4.5x higher than the second-best results while being more efficient.

ACKNOWLEDGMENTS

This work was supported by a Google Research Award, by the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2022-0-00157, Robust, Fair, Extensible Data-Centric Continual Learning), and by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2022R1A2C2004382).

REFERENCES

[1] Naoki Abe and Hiroshi Mamitsuka. 1998. Query Learning Strategies Using Boosting and Bagging. In *ICML*. San Francisco, CA, USA, 1–9.

[2] Jacob Abernethy, Pranjal Awasthi, Matthäus Kleindessner, Jamie Morgenstern, Chris Russell, and Jie Zhang. 2022. Active Sampling for Min-Max Fairness. In *ICML*.

[3] Jacob D. Abernethy, Pranjal Awasthi, Matthäus Kleindessner, Jamie Morgenstern, Chris Russell, and Jie Zhang. 2022. Active Sampling for Min-Max Fairness. In *ICML*. 53–65.

[4] Umang Aggarwal, Adrian Popescu, and Céline Hudelot. 2020. Active Learning for Imbalanced Datasets. In *WACV*. 1417–1426.

[5] Umang Aggarwal, Adrian Popescu, and Céline Hudelot. 2021. Minority Class Oriented Active Learning for Imbalanced Datasets. In *ICPR*. 9920–9927.

[6] Hadis Anahideh, Abolfazl Asudeh, and Saravanan Thirumuruganathan. 2022. Fair active learning. *Expert Systems with Applications* 199 (2022), 116981.

[7] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. 2016. Machine bias: There's software used across the country to predict future criminals. And its biased against blacks.

[8] Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2020. Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds. In *ICLR*.

[9] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. 2002. Finite-Time Analysis of the Multiarmed Bandit Problem. 47, 2–3 (2002), 235–256.

[10] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. 2002. The Nonstochastic Multiarmed Bandit Problem. *SIAM J. Comput.* 32, 1 (2002), 48–77.

[11] Jaeho Bang, Gaurav Tarlok Kakkar, Pramod Chunduri, Subrata Mitra, and Joy Arulraj. 2023. Seiden: Revisiting Query Processing in Video Database Systems. *Proc. VLDB Endow.* 16, 9 (may 2023), 2289–2301.

[12] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2019. *Fairness and Machine Learning: Limitations and Opportunities*. fairmlbook.org. http://www.fairmlbook.org.

[13] Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. 2018. AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias.

[14] Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert E. Schapire. 2011. Contextual Bandit Algorithms with Supervised Learning Guarantees. In *AISTATS (JMLR Proceedings)*, Vol. 15. 19–26.

[15] Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. 2020. *Fairlearn: A toolkit for assessing and improving fairness in AI*. Technical Report MSR-TR-2020-32. Microsoft.

[16] Christopher Buss, Jasmin Mousavi, Mikhail Tokarev, Arash Termehchy, David Maier, and Stefan Lee. 2023. Effective Entity Augmentation by Querying External Data Sources. *Proc. VLDB Endow.* 16, 11 (jul 2023), 3404–3417.

[17] Yiting Cao and Chao Lan. 2022. Fairness-Aware Active Learning for Decoupled Model. In *IJCNN*. 1–9.

[18] Chengliang Chai, Jiabin Liu, Nan Tang, Guoliang Li, and Yuyu Luo. 2022. Selective Data Acquisition in the Wild for Model Charging. *Proc. VLDB Endow.* 15, 7 (2022), 1466–1478.

[19] Irene Y. Chen, Fredrik D. Johansson, and David A. Sontag. 2018. Why Is My Classifier Discriminatory?. In *NeurIPS*. 3543–3554.

[20] Alexandra Chouldechova. 2017. Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments. *Big Data* 5, 2 (2017), 153–163.

[21] Jeffrey Dastin. 2018. Amazon scraps secret AI recruiting tool that showed bias against women. Reuters (2018).

[22] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. 2021. Retiring Adult: New Datasets for Fair Machine Learning. In *NeurIPS*. 6478–6490.

[23] Sanghamitra Dutta, Dennis Wei, Hazar Yueksel, Pin-Yu Chen, Sijia Liu, and Kush R. Varshney. 2020. Is There a Trade-Off Between Fairness and Accuracy? A Perspective Using Mismatched Hypothesis Testing. In *ICML*, Vol. 119. 2803–2813.

[24] Ehsan Elhamifar, Guillermo Sapiro, Allen Y. Yang, and S. Shankar Sastry. 2013. A Convex Optimization Framework for Active Learning. In *ICCV*. 209–216.

[25] Hossein Esfandiari, Amin Karbasi, Abbas Mehrabian, and Vahab Mirrokni. 2021. Regret Bounds for Batched Bandits. 35, 8 (May 2021), 7340–7348.

[26] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and Removing Disparate Impact. In *KDD*. 259–268.

[27] Stefan Grafberger, Shubha Guha, Paul Groth, and Sebastian Schelter. 2023. Mlwhatif: What If You Could Stop Re-Implementing Your Machine Learning Pipeline Analyses over and Over? *Proc. VLDB Endow.* 16, 12 (aug 2023), 4002–4005.

[28] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of Opportunity in Supervised Learning. In *NeurIPS*. 3315–3323.

[29] Wei-Ning Hsu and Hsuan-Tien Lin. 2015. Active Learning by Learning. In *AAAI*. 2659–2665.

[30] Ihab F. Ilyas and Theodoros Rekatsinas. 2022. Machine Learning and Data Cleaning: Which Serves the Other? *ACM J. Data Inf. Qual.* 14, 3 (2022), 13:1–13:11.

[31] Vasileios Iosifidis and Eirini Ntoutsi. 2019. AdaFair: Cumulative Fairness Adaptive Boosting. In *CIKM*. ACM, 781–790.

[32] Faisal Kamiran and Toon Calders. 2011. Data preprocessing techniques for classification without discrimination. *Knowl. Inf. Syst.* 33, 1 (2011), 1–33.

[33] Jung-Hun Kim, Milan Vojnovic, and Se-Young Yun. 2022. Rotting Infinitely Many-Armed Bandits. In *ICML (Proceedings of Machine Learning Research)*, Vol. 162. PMLR, 11229–11254.

[34] Sanjay Krishnan, Michael J Franklin, Ken Goldberg, and Eugene Wu. 2017. Boostclean: Automated error detection and repair for machine learning. *arXiv preprint arXiv:1711.01299* (2017).

[35] Preethi Lahoti, Krishna P. Gummadi, and Gerhard Weikum. 2019. Operationalizing Individual Fairness with Pairwise Fair Representations. *Proc. VLDB Endow.* 13, 4 (Dec. 2019), 506–518.

[36] Tor Lattimore and Csaba Szepesvári. 2020. *Bandit Algorithms*. Cambridge University Press.

[37] Dong-Hyun Lee. 2013. Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks.

[38] Nir Levine, Koby Crammer, and Shie Mannor. 2017. Rotting Bandits. In *NeurIPS*. 3074–3083.

[39] David D. Lewis and William A. Gale. 1994. A Sequential Algorithm for Training Text Classifiers. In *SIGIR*. 3–12.

[40] Zifan Liu, Zhechun Zhou, and Theodoros Rekatsinas. 2022. Picket: guarding against corrupted data in tabular data during learning and inference. *VLDB J.* 31, 5 (2022), 927–955.

[41] Aditya Krishna Menon and Robert C. Williamson. 2018. The cost of fairness in binary classification. In *FAT*, Vol. 81. 107–118.

[42] Hieu Tat Nguyen and Arnold W. M. Smeulders. 2004. Active learning using pre-clustering. In *ICML*.

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

[44] Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. *Proc. VLDB Endow.* 11, 3, 269.

[45] Yuji Roh, Kangwook Lee, Steven Whang, and Changho Suh. 2021. Sample Selection for Fair and Robust Training. In *NeurIPS*. 815–827.

[46] Yuji Roh, Kangwook Lee, Steven Euijong Whang, and Changho Suh. 2021. Fair-Batch: Batch Selection for Model Fairness. In *ICLR*.

[47] Dan Roth and Kevin Small. 2006. Margin-Based Active Learning for Structured Output Spaces. In *ECML*, Vol. 4212. Springer, 413–424.

[48] Nicholas Roy and Andrew McCallum. 2001. Toward Optimal Active Learning Through Sampling Estimation of Error Reduction. In *ICML*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 441–448.

[49] Babak Salimi, Luke Rodriguez, Bill Howe, and Dan Suciu. 2019. Interventional fairness: Causal database repair for algorithmic fairness. In *SIGMOD*. 793–810.

[50] Ozan Sener and Silvio Savarese. 2018. Active Learning for Convolutional Neural Networks: A Core-Set Approach. In *ICLR*.

[51] Burr Settles. 2012. *Active Learning*. Morgan & Claypool Publishers.

[52] Burr Settles and Mark Craven. 2008. An Analysis of Active Learning Strategies for Sequence Labeling Tasks. In *EMNLP*. 1070–1079.

[53] Burr Settles, Mark Craven, and Soumya Ray. 2007. Multiple-instance Active Learning. In *NIPS* (Vancouver, British Columbia, Canada). Curran Associates Inc., USA, 1289–1296.

[54] H. S. Seung, M. Opper, and H. Sompolinsky. 1992. Query by Committee. In *COLT* (Pittsburgh, Pennsylvania, USA). ACM, New York, NY, USA, 287–294.

[55] C. E. Shannon. 2001. A Mathematical Theory of Communication. *SIGMOBILE Mob. Comput. Commun. Rev.* 5, 1 (jan 2001), 3–55.

[56] Amr Sharaf, Hal Daumé III, and Renkun Ni. 2022. Promoting Fairness in Learned Models by Learning to Active Learn under Parity Constraints. In *FAccT*. ACM, 2149–2156.

[57] Shubhanshu Shekhar, Greg Fields, Mohammad Ghavamzadeh, and Tara Javidi. 2021. Adaptive Sampling for Minimax Fair Classification. In *NeurIPS*. 24535–24544.

[58] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. 2012. Practical Bayesian Optimization of Machine Learning Algorithms. In *NeurIPS (NIPS'12)*. 2951–2959.

[59] Ki Hyun Tae and Steven Euijong Whang. 2021. Slice Tuner: A Selective Data Acquisition Framework for Accurate and Fair Machine Learning Models. In *SIGMOD*. ACM, 1771–1783.

[60] Ki Hyun Tae, Hantian Zhang, Jaeyoung Park, Kexin Rong, and Steven Euijong Whang. 2023. Falcon: Fair Active Learning using MABs. https://github.com/as-anonymous/Falcon/blob/main/techreport.pdf.

[61] Manasi Vartak, Sajjadur Rahman, Samuel Madden, Aditya Parameswaran, and Neoklis Polyzotis. 2015. Seedb: Efficient data-driven visualization recommendations to support visual analytics. *Proc. VLDB Endow.* 8, 13, 2182.

[62] Sahil Verma and Julia Rubin. 2018. Fairness definitions explained. In *FairWare@ICSE*. 1–7.

[63] Joannès Vermorel and Mehryar Mohri. 2005. Multi-armed Bandit Algorithms and Empirical Evaluation. In *ECML*. Springer Berlin Heidelberg, 437–448.

[64] Dan Wang and Yi Shang. 2014. A new active labeling method for deep learning. In *IJCNN*. 112–119.

[65] Yizao Wang, Jean-yves Audibert, and Rémi Munos. 2008. Algorithms for Infinitely Many-Armed Bandits. In *NeurIPS*, Vol. 21. Curran Associates, Inc.

[66] Steven Euijong Whang, Yuji Roh, Hwanjun Song, and Jae-Gil Lee. 2023. Data Collection and Quality Challenges in Deep Learning: A Data-Centric AI Perspective. *VLDB J.* (2023).

[67] Tsung-Han Wu, Hung-Ting Su, Shang-Tse Chen, and Winston H. Hsu. 2022. Fair Robust Active Learning by Joint Inconsistency. arXiv:2209.10729

[68] Hantian Zhang, Xu Chu, Abolfazl Asudeh, and Shamkant B. Navathe. 2021. OmniFair: A Declarative System for Model-Agnostic Group Fairness in Machine Learning. In *SIGMOD*. ACM, 2076–2088.

[69] Hantian Zhang, Ki Hyun Tae, Jaeyoung Park, Xu Chu, and Steven Euijong Whang. 2023. iFlipper: Label Flipping for Individual Fairness. *Proc. ACM Manag. Data* 1, 1 (2023), 8:1–8:26.

[70] Han Zhao and Geoffrey J. Gordon. 2019. Inherent Tradeoffs in Learning Fair Representations. In *NeurIPS*. 15649–15659.

A TARGET GROUPS FOR OTHER METRICS.

We continue from Section 3.1 and explain the target subgroups for Equalized Odds (ED), Predictive Parity (PP) and Equalized Error Rate (EER). For ease of reference, we present the complete results for each fairness metric in Table 10 again.

Target Subgroups for ED. In addition to the EO case in Example 2, for ED (Equation 2), the goal is to also close the gap between $p(\hat{y} = 1|y = 0, z = 0)$ and $p(\hat{y} = 1|y = 0, z = 1)$. Then, improving $p(\hat{y} = 0|y = 0, z = 1)$ directly decreases $p(\hat{y} = 1|y = 0, z = 1)$ and reduces the disparity. Hence, improving the accuracy on $(y = 1, z = 0)$ or $(y = 0, z = 1)$, whichever minimizes the larger gap, will result in fairness improvement.

Target Subgroups for PP. For PP (Equation 4), let us assume that the FDR disparity is dominant, and the goal is to close the gap between $p(y = 1|\hat{y} = 1, z = 0)$ and $p(y = 1|\hat{y} = 1, z = 1)$ where the first term is smaller. We know that

$$p(y = 1|\hat{y} = 1) = \frac{p(y = 1, \hat{y} = 1)}{p(\hat{y} = 1)} = \frac{p(y = 1)p(y = 1|\hat{y} = 1)}{p(\hat{y} = 1)}$$

Then $p(y = 1|\hat{y} = 1, z = 0) < p(y = 1|\hat{y} = 1.z = 1)$ can be rewritten as $\frac{p(y=1|z=0)p(y=1|\hat{y}=1,z=0)}{p(\hat{y}=1|z=0)} < \frac{p(y=1|z=1)p(y=1|\hat{y}=1,z=1)}{p(\hat{y}=1|z=1)}$. We also know that the denominator term $p(\hat{y} = 1)$ can be expressed as $p(y = 1)p(\hat{y} = 1|y = 1) + p(y = 0)(1 - p(\hat{y} = 0|y = 0))$. By arranging the terms, we get the following inequality:

$$\frac{p(y = 0|z = 1)(1 - p(\hat{y} = 0|y = 0, z = 1))}{p(y = 1|z = 1)p(\hat{y} = 1|y = 1, z = 1)} < \frac{p(y = 0|z = 0)(1 - \underline{\underline{p(\hat{y} = 0|y = 0, z = 0))}}}{p(y = 1|z = 0)\underline{p(\hat{y} = 1|y = 1, z = 0)}}$$

In this case, we can see that improving $p(\hat{y} = 0|y = 0, z = 0))$ or $p(\hat{y} = 1|y = 1, z = 0)$ results in deceasing the right term. Therefore, both $(y = 0, z = 0)$ and $(y = 1, z = 0)$ can be the target subgroups to improve PP.

Consider the other case where the FOR disparity is larger, and $p(y = 1|\hat{y} = 0, z = 1) < p(y = 1|\hat{y} = 0, z = 1)$. Using a similar approach, we can derive that the target groups are $(y = 0, z = 1)$ and $(y = 1, z = 1)$ to close the FOR gap. We summarize the results for all possible scenarios in Table 10.

Target Subgroups for EER. For EER (Equation 5), the goal is to ensure a similar classification error rate across different sensitive groups. Similar to the previous examples, we assume that $p(\hat{y} \neq y|z = 0) < p(\hat{y} \neq y|z = 1)$, which can also be expressed as $p(\hat{y} = y|z = 1) < p(\hat{y} = y|z = 0)$. In order to improve $p(\hat{y} = y|z = 1)$ in the left term, we need to label data from $(y = 1, z = 1)$ or $(y = 0, z = 1)$, as $p(\hat{y} = y|z = 1)$ can be written as $p(y = 1, z = 1)\underline{p(\hat{y} = 1|y = 1, z = 1)} + p(y = 0, z = 1)\underline{p(\hat{y} = 0|y = 0, z = 1)}$. On the other hand, if $p(\hat{y} \neq y|z = 1)$ is smaller, both $(y = 1, z = 1)$ and $(y = 0, z = 1)$ subgroups should be targeted to improve EER.

Trial-and-error for PP and EER. For the target groups for PP and EER in Table 10, we observe that an undesirable label with one of the target groups corresponds to another target subgroup. Hence, trial-and-error does not postpone any samples in these cases.

Metric	Target Subgroups (y, z)
DP	$(1, z^*)$ or $(0, 1 - z^*)$
EO	$(1, z^*)$
ED	$(0, 1 - z^*)$, if FPR gap ≥ FNR gap $(1, z^*)$, otherwise
PP	$(0, 1 - z^*)$ or $(1, 1 - z^*)$, if FOR gap ≥ FDR gap $(0, z^*)$ or $(1, z^*)$, otherwise
EER	$(0, 1 - z^*)$ or $(1, 1 - z^*)$

Table 10: Target subgroups for each group fairness measure when a sensitive group $z^* \in \{0, 1\}$ has a lower fairness value.

B IMPLEMENTATION DETAILS

B.1 Dataset Configurations

We continue from Section 6.1 and provide more details on the data configurations. Table 11 shows how we construct the data distributions for the four datasets. We observe that the original distributions of the datasets are not heavily biased. That is, we can achieve a nearly perfect fair classifier using a relatively small labeling budget. Although Falcon also performs well in this scenario, it becomes difficult to compare the performance of different methods because labeling a few samples is sufficient to improve fairness. Hence, we increase the bias by taking a smaller subset of the minority group or a larger subset of the majority group, and then conduct experiments using a larger labeling budget.

The sensitive groups used for each dataset are as follows:

- TravelTime: Female and Male.
- Employ: Disability and Able-bodied.
- Income: White, Asian, and Others.
- COMPAS: Female and Male.

B.2 Baselines

We continue from Section 6.1 and provide more details on the fair AL algorithms, *FAL* [6] and *D-FA^{2}L* [17].

- *FAL*: The first fairness-aware AL algorithm that optimizes both group fairness and accuracy. *FAL* linearly combines entropy with the expected unfairness reduction, which estimates the expected fairness improvement for each unlabeled sample over all possible labels. This approach, however, requires retraining the model $2 \times |D_{un}|$ times per iteration. In order to improve efficiency, *FAL* computes the reduction in unfairness only for m samples with the highest entropy value, and then chooses top b samples from this subset. As a result, a higher m favors better fairness, but requires more computation time. In our experiments, we set an upper bound for the m value to 64, as it has been reported to provide the best performance [17].
- *D-FA^{2}L*: A disagreement-based fairness-aware AL algorithm. For a binary-valued sensitive attribute (i.e., $\mathbb{Z} = \{0, 1\}$), a decouple model is a pair of models (h_0, h_1), where each model h_i is trained on a specific sensitive group z_i.

Datasets	Sizes	Sen. Group 1		Sen. Group 2		Sen. Group 3	
		Label 0	Label 1	Label 0	Label 0	Label 0	Label 0
TravelTime (gender)	$\lvert D_{train} \rvert$	1,115	181	441	709	-	-
	$\lvert D_{un} \rvert$	22,300	3,630	8,820	14,190	-	-
	$\lvert D_{test} \rvert$	11,150	1,815	4,410	7,095	-	-
	$\lvert D_{val} \rvert$	1,115	181	441	709	-	-
Employ (disability)	$\lvert D_{train} \rvert$	579	81	1,673	3,292	-	-
	$\lvert D_{un} \rvert$	17,370	2,430	50,190	98,760	-	-
	$\lvert D_{test} \rvert$	8,685	1,215	25,095	49,380	-	-
	$\lvert D_{val} \rvert$	579	81	1,673	3,292	-	-
Income (race)	$\lvert D_{train} \rvert$	2,019	268	169	314	309	109
	$\lvert D_{un} \rvert$	40,380	5,360	3,380	6,280	6,180	2,180
	$\lvert D_{test} \rvert$	20,190	2,680	1,690	3,140	3,090	1,090
	$\lvert D_{val} \rvert$	2,019	268	169	314	309	109
COMPAS (gender)	$\lvert D_{train} \rvert$	86	158	37	13	-	-
	$\lvert D_{un} \rvert$	688	1,264	300	104	-	-
	$\lvert D_{test} \rvert$	344	632	150	52	-	-
	$\lvert D_{val} \rvert$	86	158	37	13	-	-

Table 11: Detailed configurations for the four datasets, with the sensitive attribute in parentheses.

D-FA^{2}L chooses a sample x that receives significantly different predictions from the decoupled models h_0 and h_1, i.e., $|p(h_0(x) = 1|x) - p(h_1(x) = 1|x)| > \alpha$ for a predefined hyperparameter α. For handling multiple sensitive attributes, we extend *D-FA^{2}L* to find a sample x that receives conflicting predictions from any two models, i.e., $\max_{z_i,z_j \in \mathbb{Z}} |p(h_i(x) = 1|x) - p(h_j(x) = 1|x)| > \alpha$. In our experiments, we vary the α value from 0.1 to 0.9.

One technique we do not make a comparison is PANDA [56], which is a meta-learning based algorithm that learns a selection policy that maximizes accuracy and fairness. We exclude this method due to the prohibitively high computational cost of meta-learning, as also noted in [67].

B.3 Fairness Evaluation

We continue from Section 6.1 and provide more details on the fairness evaluation. We consider five group fairness measures including demographic parity (DP), equal opportunity (EO), equalized odds (ED), predictive parity (PP), and equalized error rate (EER). To quantify fairness, we define a fairness score as one minus the maximum fairness disparity [15] across any sensitive groups on the test set, as described in Section 2.1. A higher value is better.

- DP score: $1 - \max_{z_i,z_j \in \mathbb{Z}} |p(\hat{y} = 1|z = z_i) - p(\hat{y} = 1|z = z_j)|$
- EO score: $1 - \max_{z_i,z_j \in \mathbb{Z}} |p(\hat{y} = 1|y = 1, z = z_i) - p(\hat{y} = 1|y = 1, z = z_j)|$
- ED score: $1 - \max_{z_i,z_j \in \mathbb{Z}, \mathrm{y} \in \{0,1\}} |p(\hat{y} = 1|y = \mathrm{y}, z = z_i) - p(\hat{y} = 1|y = \mathrm{y}, z = z_j)|$
- PP score: $1 - \max_{z_i,z_j \in \mathbb{Z}, \hat{\mathrm{y}} \in \{0,1\}} |p(y = 1|\hat{y} = \hat{\mathrm{y}}, z = z_i) - p(y = 1|\hat{y} = \hat{\mathrm{y}}, z = z_j)|$
- EER score: $1 - \max_{z_i,z_j \in \mathbb{Z}} |p(\hat{y} \neq y|z = z_i) - p(\hat{y} \neq y|z = z_j)|$

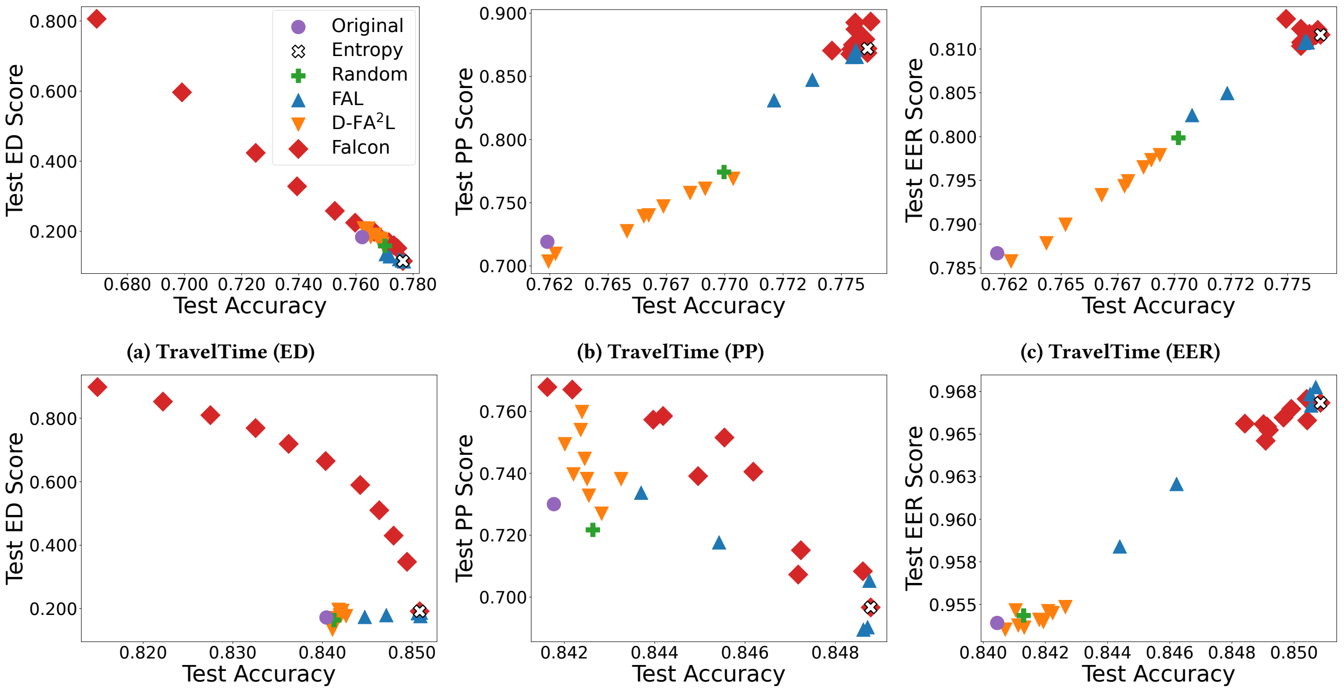
B.4 Other Experimental Settings

We continue from Section 6.1 and provide more details on the experimental settings. We implement logistic regression (LR) and neural network (NN) models using Scikit-learn [43] library. For LR, we set the regularization strength to 1.0. For NN, we use a multi-layer perceptron with one hidden layer consisting of 10 nodes and set the learning rate to 0.0001. We evaluate all models on a separate test set and repeat the experiments with ten different random seeds. For the NN experiments, we use three random seeds due to the long comparison time, which exceeds 30 hours per seed.

C TRADE-OFFS FOR OTHER MEASURES

We continue from Section 6.2 and perform the same experiments using equalized odds (ED), predictive parity (PP), and equalized error rate (EER). Figure 9 shows the accuracy-fairness trade-off results on the TravelTime and Employ datasets. The key trends are still similar to Figure 6 where Falcon outperforms the other fair AL baselines in terms of accuracy and fairness.

Another interesting observation is that we have different shapes of trade-offs for EER (Figure 9c and Figure 9f) and PP (Figure 9b), where fairness and accuracy can be improved at the same time. This is due to the fact that improving fairness sometimes aligns with improving overall accuracy. For EER, the goal is to equalize the accuracy between different sensitive groups. Hence, we need to label more samples from the group with lower accuracy, which leads to an overall accuracy improvement as well. In addition, the result for PP on the TravelTime dataset (Figure 9b) exhibits a similar pattern because the target groups are sometimes the same as those for EER (detailed conditions are specified in Table 10). We also note that Falcon and *Entropy* produce comparable results in these cases, as their underlying objectives are similar, i.e., improving the accuracy of minority groups.



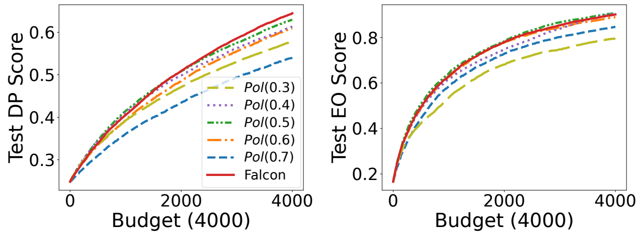
(a) TravelTime (ED) **(b) TravelTime (PP)** **(c) TravelTime (EER)**

(d) Employ (ED) **(e) Employ (PP)** **(f) Employ (EER)**

Figure 9: Accuracy-fairness trade-offs of logistic regression on the two datasets (TravelTime and Employ) using the three fairness measures (ED, PP, and EER). In addition to the baselines, we add the result of model training without labeling any additional data and call it "Original." As a result, Falcon significantly improves fairness compared to the fair AL baselines.

D POLICY SEARCH FOR EMPLOY DATASET

We continue from Section 6.4 and show experimental results on the Employ dataset where we use DP and EO as the target fairness measures. In Figure 10, the key trends are still similar to Figure 7 where Falcon achieves the best or second-best fairness improvement among the single policy baselines. In addition, we provide a detailed analysis for Figure 10a in Figure 11a where we show how Falcon updates the selection probabilities of each policy. Here, the sensitive attribute is `disability`, and we have two target groups, (`attribute=disability, label=positive`) and (`attribute=able-bodied, label=negative`), denoted as $(D-1)$ and $(A-0)$, respectively. As a result, Falcon increases the selection probability for $r = 0.4$ for $(D-1)$ the most. This finding is consistent with Figure 11b, where it shows that $r = 0.4$ for $(D-1)$ and $r = 0.5$ for $(D-1)$ are the most effective policies. Hence, we conclude that Falcon correctly updates the MAB to identify the optimal policy among the candidate policy set.



(a) Test DP Score **(b) Test EO score**

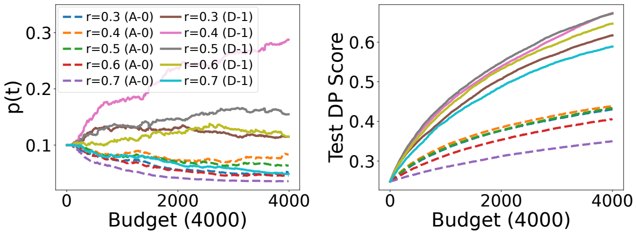
Figure 10: Fairness comparison of Falcon against a set of single policy baselines on the Employ dataset.

E COMPARISON WITH NEURAL NETWORK

In Section 6.2, we compared Falcon with the baselines using logistic regression models. In this section, we perform the same experiments using neural network models. Figure 12 is the trade-off results on the TravelTime and Employ datasets when using DP and EO. For *FAL*, we exclude the results for $m = 32$ and $m = 64$ because the overall running time takes more than 24 hours. The observations are similar to those of Figure 6 where Falcon consistently outperforms the other baselines in terms of fairness and accuracy and provides much cleaner trade-offs. The results clearly demonstrate how Falcon benefits other ML models.

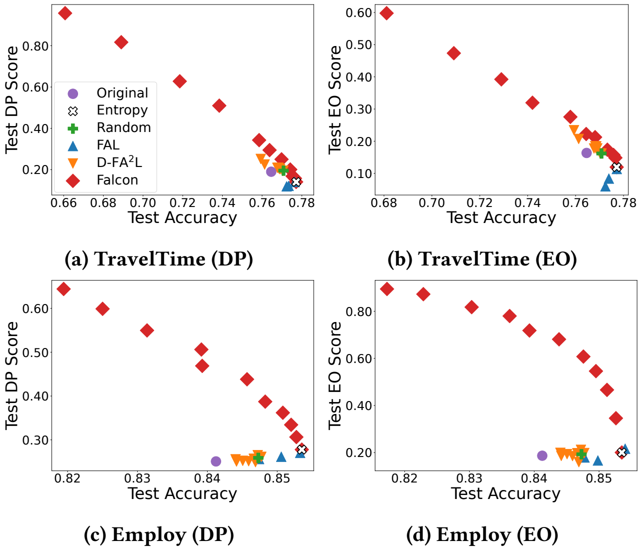
F MORE POLICY SETS

We continue from Section 6.6 and perform additional experiments to investigate the impact of policy sets on the Falcon's performance. We first considered simpler policy sets with only two policies $[r = 0.4, r = 0.7]$, $[r = 0.3, r = 0.8]$ to check the impact of the quality of the policies. We also a policy set that contains extreme policies where we added $r = 0.1, 0.2, 0.8, 0.9$ to our default set to check the impact of adding extreme policies. Table 12 shows the fairness



(a) Policy Selection Probability **(b) All Single Policies**

Figure 11: A detailed analysis for Figure 10a. (a) Falcon increases the selection probability of $r = 0.4$ for the target group $(D-1)$, where we denote the sensitive attribute (Disability or Able-bodied) and label of the target subgroup in parentheses. (b) Fairness improvements for all single policies. The policies $r = 0.4$ for $(D-1)$ and $r = 0.5$ for $(D-1)$ are the most effective in improving the DP score.



(a) TravelTime (DP) **(b) TravelTime (EO)**

(c) Employ (DP) **(d) Employ (EO)**

Figure 12: Accuracy-fairness trade-offs of neural network on the two datasets (TravelTime and Employ) using the two fairness measures (DP and EO). In addition to the baselines, we add the result of model training without labeling any additional data and call it "Original." As a result, only Falcon significantly improves fairness and shows clear accuracy and fairness trade-offs.

results for different policy sets using the TravelTime and Employ datasets. We first observe that $[r = 0.4, r = 0.7]$ is better than $[r = 0.3, r = 0.8]$ because the policies in the former set are closer to the optimal policies for each dataset, which are $r = 0.6$ for TravelTime (Figure 7) and $r = 0.5$ for Employ (Figure 10). However, our default policy set outperforms these alternatives in most cases, as it already includes the optimal policies. In addition, the default set performs better than the set with extreme policies in the last row. This is because extreme policies usually yield a worse trade-off between informativeness and postpone rate. Thus, our default policy set offers a balanced selection of diverse policies that are not too extreme.

	TravelTime		Employ	
Policy Set	**DP**	**EO**	**DP**	**EO**
$[r = 0.4, r = 0.7]$	0.948	**0.622**	0.644	0.899
$[r = 0.3, r = 0.8]$	0.815	0.385	0.626	0.890
$[r = 0.3, \ldots, r = 0.7]$ (default)	**0.966**	0.616	**0.645**	**0.901**
$[r = 0.1, \ldots, r = 0.9]$	0.943	0.554	0.613	0.878

Table 12: Impact of different policy sets on Falcon.

Datasets	Fair.	Fairness Score				
		FAL	FAL$^+$	D-FA^{2}L	D-FA^{2}L$^+$	Falcon
Income	DP	0.366	0.720	0.361	0.682	**0.816**
	EO	0.453	0.777	0.435	0.700	**0.834**
COMPAS	DP	0.392	0.728	0.373	0.760	**0.861**
	EO	0.403	0.584	0.400	0.666	**0.924**

Table 13: Comparison of Falcon against fair AL baselines combined with trial-and-error on the Income and COMPAS datasets.

Datasets	Fairness	Fairness Score			
		Original	EXP3	EXP3-IX	EXP4.P
Income	DP	0.355	0.816	**0.820**	0.801
	EO	0.402	**0.834**	0.829	**0.834**
COMPAS	DP	0.365	0.861	**0.883**	0.856
	EO	0.372	0.924	**0.929**	0.924

Table 14: Fairness results on the Income and COMPAS datasets when using Falcon with other adversarial MABs.

G COMBINING TRIAL-AND-ERROR WITH BASELINES FOR OTHER DATASETS

We continue from Section 6.7 and perform the same experiments using the Income and COMPAS datasets. In Table 13, the key trends are similar to Table 8 where (1) the fair AL baselines can be improved with trial-and-error, but (2) Falcon still outperforms them all, demonstrating the importance of the other Falcon components.

H ADVERSARIAL MABS ON MORE DATASETS

We continue from Section 6.8 and show experimental results on the Income and COMPAS datasets in Table 14. The key observation is similar to Table 9 where EXP3 exhibits comparable performance to other adversarial MABs with high probability regret bounds. But, we re-emphasize that Falcon can be compatible with any MAB capable of handling adversarial rewards.