End-to-End Fairness Optimization with Fair Decision Focused Learning

ANONYMOUS AUTHOR(S)

Many real-world applications make algorithmic decisions based on predictions, and fairness must be considered throughout the entire decision-making pipeline. Traditional two-stage approaches, which handle fairness separately in prediction and optimization, can lead to misalignment between fairness at each stage. We present Fair Decision Focused Learning (FDFL), an end-to-end framework that optimizes predictions and decisions while incorporating fairness at both stages. Our framework uses gradient-based methods adapted for fairness optimization and can handle general fairness measures. We derive closed-form solutions for a specific class of α-fairness measure with linear utility functions that enable exact and efficient gradient computation, significantly reducing computational overhead. Through experiments on healthcare resource allocation, we show FDFL can achieve better fairness outcomes compared to two-stage approaches. Our results show the importance of considering fairness holistically across the prediction-to-decision pipeline and provide a practical framework for applications requiring both accurate and fair data-driven decisions.

CCS Concepts: • **Computing methodologies** → **Machine learning**; • **Applied computing** → *Decision analysis*.

Additional Key Words and Phrases: End-to-End Fairness, Decision Focused Learning

ACM Reference Format:
Anonymous Author(s). 2018. End-to-End Fairness Optimization with Fair Decision Focused Learning. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX).* ACM, New York, NY, USA, 25 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Many real-world applications require consideration of fairness throughout the chain of decisions. When distributing scarce medical resources to healthcare facilities across different regions, public health agencies must forecast demand before developing allocation plans. Fair forecasts ensure that needs of disadvantaged regions are not overlooked, while fair allocations provide appropriate prioritization across all facilities. Similarly, the planning and distribution of other social resources, such as urban transportation and education funding, pose fairness challenges from understanding heterogeneous needs and ensuring equitable benefits. Another domain requiring holistic approach to fairness is necessary, is hiring. During the complicated process of sourcing candidates, screening applications, and selecting whom to interview and hire, bias and unfairness in any step can perpetuate systematic inequalities and undermine organizational diversity. These examples highlight the value of end-to-end fairness integration across the entire decision process.

Drawing motivation from the above examples, we study a generic decision process that consists of a prediction task and an optimization task. The prediction task uses data to forecast uncertain or unknown quantities, which are needed to formulate the optimization model for choosing the optimal actions. The prediction task should be free of unfair discrimination, and the optimization task should pursue the desirable equity performance in the final outcomes. A conventional strategy addresses these two tasks sequentially: predictions are made first and then input into the

Manuscript submitted to ACM

optimization model. In two-stage predict-then-optimize (PTO) approaches, fairness can be handled independently in each task, that is, we apply fair prediction followed by fair optimization. There is a large number of fair machine learning algorithms for seeking group-level or individual-level fairness in predictions [30]. In optimization models, fairness measures capturing various equity perspectives can be formulated as objectives or constraints [9].

Handling fairness in separate stages overlook the potential misalignment and interconnection between prediction and decision fairness. For example, optimizing fair decisions based on fair predictions may still generate unfair outcomes. Consider a policymaker allocating a new type of healthcare service to local communities, where rural areas tend to have lower demands than urban areas due to a historical lack of healthcare access and awareness. Moreover, such historical bias exist in data and cause demand predictions without any fairness consideration to underestimate some rural areas' demands. In this problem, a fair prediction model will try to increase the demand forecasts for rural areas to reduce disparity. However, these forecast increases would cause some rural areas to be less prioritized than what they actually deserve, thus perpetuating the systematic inequity in healthcare access between rural and urban areas. To seek fair outcomes, a more desirable alternative is to integrate fair prediction and fair optimization in an end-to-end manner to link prediction fairness directly to decision fairness.

In this paper, we formulate end-to-end fairness optimization (E2EFO) as a paradigm for considering fairness in end-to-end decisions. E2EFO aims to optimize fairness in decisions generated based on predictions. We develop fair decision-focused learning (FDFL) as an integrated approach to E2EFO. The FDFL framework builds upon standard decision-focused learning (DFL), an end-to-end machine learning approach that trains a prediction model to optimize downstream decision accuracy [16, 19, 45]. DFL is an alternative to the conventional two-stage PTO approaches, where a prediction model is trained to optimize the predictive accuracy separately from the decision process. By contrast, DFL emphasizes obtaining decisions that are accurate in terms of the downstream objective, even in the presence of prediction errors.

In our FDFL framework, prediction fairness focuses on eliminating undesirable disparity in forecasts and is formulated with constraints or regularization terms in predictor training. All fairness formulations in existing in-processing fair supervised machine learning methods [30] are applicable to capture prediction fairness in FDFL. Decision fairness characterizes the equity goal of decision outcomes and is represented in the objective function of the optimization task. The optimization objective is a utility-based fairness measure assessing the desirability of a decision's utility distribution with respect to the underlying fairness perspective. Such fairness measures are broadly applied in operations research and mechanism design [9, 21]. FDFL aims to learn a predictor that yields fair decisions aligning both the prediction and decision objectives. Compared with standard DFL, FDFL aligns prediction and optimization performances in terms of both accuracy and fairness.

We make the following contributions in this paper. First, we formulate the E2EFO problem and develop the FDFL framework to find optimal fair decisions. Second, we propose gradient-based algorithms for FDFL training. We first present a perturbation-based gradient approximation method to support FDFL with a general fair optimization decision model. Then utilizing the closed-form solutions to a specific class of fairness optimization model, we derive exact gradient formulas and design an efficient specialized FDFL algorithm for this class of problem. Lastly, we demonstrate the benefits of FDFL in comparison with traditional two-stage approaches through a stylized example for theoretical intuition and real data based experiments for empirical insights.

The rest of the paper is organized as follows. Section 2 reviews the related works on DFL versus two-stage PTO and fairness methods in machine learning and optimization. Section 3 describes the E2EFO problem and formally presents the FDFL framework. Section 4 discusses training algorithms for implementing FDFL. In Section 5, we apply FDFL and two-stage PTO with fairness consideration on a real application motivated medical resource allocation problem. Besides comparing the performances between FDFL and PTO, we also compare different FDFL algorithms to understand their runtime, training, and decision performances. Lastly, Section 6 concludes the paper and discusses future directions to investigate.

2 Related Works

2.1 Predict-then-Optimize and Decision-Focused Learning

Data-driven decision-making frequently encounters scenarios where optimization models depend on parameters that must be estimated from data. Two primary paradigms have emerged to address this challenge: Prediction-Focused Learning (PFL), also known as Predict-then-Optimize (PTO), and Decision-Focused Learning (DFL), also known as End-to-End Learning (E2E). The conventional PTO approach follows a two-stage process, first employing machine learning to predict unknown parameters from relevant features, then using these predictions as deterministic inputs to optimize decisions. While PTO aligns well with classical stochastic optimization [6], it has fundamental limitations arising from its segregated structure. Most critically, the prediction stage operates independently of downstream decision-making, potentially leading to suboptimal decisions even when predictions appear accurate [19, 38, 45]. Recent works have provided insights on these limitations. For example, [16] demonstrated the possible mismatch between prediction and decision goals and argued that prediction-focused methods are prone to model selection bias. [44] further conceptualized the gaps between good predictions and good decisions due to various factors including treatment effect heterogeneity and feedback loops between decisions and predicted outcomes. DFL addresses these limitations by integrating prediction and optimization into a unified framework. By embedding the optimization problem directly into the training of predictive model, DFL enables the prediction component to anticipate its impact on final decisions.

The key challenge in DFL lies in computing gradients through the optimization procedure to train the predictive model. Three main approaches have emerged to address this challenge: differentiation through optimization, surrogate optimization, and surrogate loss functions. Early DFL methods focused on differentiation through optimization, primarily dealing with linear programming models in the downstream decision task [5, 19, 45]. These approaches have since expanded to handle quadratic programming [1, 3], and general nonlinear optimization [42]. Negative Identity Backpropagation (NID) [40] and Perturbation Gradient (PG) Loss [24] provide more efficient ways to compute gradients through optimization procedures. The challenge of non-differentiability in optimization has also led to the development of surrogate loss approaches. Smart Predict-then-Optimize loss [19] and Noise Contrastive Estimation [33] provide convex surrogates that upper bound the decision regret. Gradient-free methods such as LANCER [48] replaces the composition of a prediction model and a solver with a smooth, neural “landscape surrogate". The surrogate is iteratively fit to solver outputs, then used in place of actual solver calls for training the predictive model. LODL [42] also avoids solver differentiation, but does so by learning a “locally optimized decision loss" that is shaped by oracle solutions. Rather than engineering solver-specific gradients, LODL trains a custom loss directly informed by the solver’s optimal decisions. This learned loss can be made convex, simplifying optimization and improving scalability. Both LANCER and LODL illustrate a broader shift toward learning a smooth or surrogate objective that emulates solver outcomes, streamlining end-to-end training in decision-focused tasks.

Research comparing PTO and DFL has yielded important theoretical insights about their relative performance. Cameron et al. [8] demonstrated that DFL’s advantages stem from its ability to adaptively handle stochastic prediction targets, while PTO must make an a priori choice about which statistics of the target distribution to model. They showed that the performance gap between PTO and DFL is closely related to the price of correlation in stochastic optimization, and identified scenarios where PTO can perform unboundedly worse than DFL. In particular, when multiple prediction targets are combined to obtain unknown parameters in the decision objective, DFL is proved to have superior performances. Elmachtoub et al. [20] revealed when the predictor model class is well-specified and data is sufficient, PTO can outperform integrated approaches (such as DFL) in terms of regret stochastic dominance. These theoretical results have useful implications for our FDFL framework. Since fairness objectives often involve multiple interrelated predictions and operate under some degree of model misspecification, e.g., due to historical biases in training data, the theoretical advantages of DFL may extend to FDFL. Additionally, as the tension between prediction accuracy and fairness creates additional complexity not captured in existing theoretical analyses, our work lays a foundation for further theoretical study related to end-to-end fairness.

Compared to existing literature, our work introduces two key advancements. First, we study fairness decision objectives that are typically nonlinear and explore the applicability of state-of-the-art DFL algorithms to handle nonlinearity. Second, while standard DFL addresses a trade-off between prediction accuracy and decision accuracy, our FDFL framework connects prediction and decision quality along the dimensions of both accuracy and fairness.

2.2 Fairness in Decision Making

Fairness in decision-making has been extensively studied, with growing interests in integrating fairness into machine learning (ML) and optimization. These fields are characterized by different fairness definitions and goals. While fair ML methods focus on achieving parity in predictions to remove discriminative bias against groups or individuals, fair optimization models emphasize equitable outcomes and impacts as measured by utilities. Among the vast literature, we review selected works in each direction to give a necessary overview and highlight papers that are more relevant to end-to-end fair decision-making.

Statistical fairness metrics dominate the field of fair ML. These metrics aim to ensure decisions from ML models, e.g., predictions, are free from discrimination against protected groups. Examples of widely studied measures include demographic parity [17], equalized odds [22], accuracy parity [4], predictive rate parity [25] , and individual fairness [17]. Majority of fair ML methods aim to address biases in standard ML models through pre-, in-, or post-processing techniques. Pre-processing methods modify input data to eliminate potential (e.g., [7, 47]). In-processing methods incorporate fairness during model training by including fairness components as constraints or objective regularizers (e.g., [15, 36, 46]). Lastly, post-processing methods adjust model outputs to attain desirable fairness (e.g., [2, 22]). The survey paper [30] provides a comprehensive review of fairness definitions and techniques in machine learning.

Fairness in optimization employs utility-based metrics grounded in social welfare theory [9]. This is a primary methodology to make fair decisions in operations research and mechanism design. Utility values capture the benefits or costs people associate with decisions of interest, and utility-based fairness metrics evaluate the desirability of utility distributions. There are three broad categories of metrics capturing different fairness perspectives. First, equality can be viewed as a proxy of fairness. Common metrics reflecting this perspective are inequality metrics, such as Gini index [13], which measures the disparity in utility outcomes. The second category emphasizes fairness for the disadvantaged, and the best known definition is the Rawlsian fairness criteria [39] that seek to prioritize people with lower utilities. The third category reflects a combined view balancing fairness and efficiency, which focuses solely on optimizing the overall utilities regardless of individual differences. A popular combined metric is α-fairness [31], which can span the entire spectrum from Rawlsian fairness to efficiency-based solutions. Applications of fairness optimization span various domains, such as, fair resource allocation in telecommunication networks [29, 35], balancing fairness and efficiency in assigning projects to university students [10], fair food bank operations [18] and disaster preparation [32] in humanitarian operations.

Along with the significant progress in both directions, recent works call for integrating ML and optimization perspectives to address outcome-centric fairness. One key motivation is the potential insufficiency of relying on fair predictions alone to ensure equitable outcomes and impacts. For example, [28] looked into the delayed impacts of fairness in ML, and demonstrated that inserting fairness, which aims to benefit certain protected groups, in ML model does not guarantee long-term improvements for the targeted groups. [41] proposed a conceptual framework distinguishing the roles of prediction modeler and decision maker in a decision-making system, and argued the importance of a holistic approach for embedding fairness throughout the entire system. Another strong motivation for bridging the two perspectives is that many practical problems naturally require both ML and optimization. The position paper [21] provide a comprehensive discussion of opportunities for combining ML and mechanism design, an important application area of optimization, to address fairness in complex decisions. More concretely, [11, 12] explore the application of a consequentialist approach, where the fairness of a decision algorithm or policy is evaluated based on the produced real-world outcomes. [12] argued that traditional fairness definitions in ML could inadvertently harm the groups they intend to protect, which shares common ground with [28]. Building on the consequentialist principles, [11] presented a practical framework that includes the elicitation of stakeholder preferences and the optimization of preference-informed policies. The authors also developed a contextual bandit algorithm to operationalize this framework.

The integration of fair ML and optimization aligns with the goal of end-to-end decision-making, which indicates the potential of DFL techniques in attaining end-to-end fairness. There have been a few recent works that explored the incorporation of fairness into DFL for the task of learning a fair ranking. For example, [26] included fairness requirements with constraints in the ranking decision model and utilized the linearity of decision objectives to extend the DFL algorithm developed in [19]. In contrast, [14] defined a fair ranking decision model with the optimization of ordered weighted average functions, and proposed a training algorithm with customized forward and backward propagation computation. Our work extends this literature by presenting a general paradigm for fair end-to-end optimization through FDFL algorithms. Compared with the mentioned prior works, we do not restrict the problem setup and study a general utility-based fairness optimization model as the decision problem.

3 Problem Formulation: End-to-End Fairness Optimization

We consider a decision-making problem with n stakeholders indexed by $i \in [n] = \{1, \dots, n\}$. Let $\mathbf{d} = (d_1, \dots, d_n) \in \mathbb{R}^n$ denote the decision vector, where d_i represents the decision (e.g., resource allocation, selection probability) for the stakeholder i. The decisions must satisfy certain constraints represented by a feasible region $\mathcal{S} \subseteq \mathbb{R}^n$. We assume the feasible region $\mathcal{S}$ is non-empty, compact, and convex. Each stakeholder i derives utility from their received decision according to a utility function $U_i : \mathbb{R} \to \mathbb{R}$, which characterizes i's overall gains and costs from the decision. Let $\mathbf{u} = (u_1, \dots, u_n)$ denote the utility vector where $u_i = U_i(d_i)$. The fairness of decisions is evaluated using a fairness measure $W : \mathbb{R}^n \to \mathbb{R}$ that maps utility vectors to a scalar value. Higher values of W indicate more desirable fairness properties. Formally, given W and $\{U_i\}_{i=1}^n$, the fairness of decisions $\mathbf{d}$ is computed as: $W(\mathbf{d}) = W(U_1(d_1), \dots, U_n(d_n))$.

The decision maker's objective is to find decisions that maximize fairness while satisfying feasibility constraints:

$$\max_{\mathbf{d}} \quad W(\mathbf{d}) \text{ s.t. } \quad \mathbf{d} \in \mathcal{S} \tag{1}$$

We assume W is concave in $\mathbf{d}$, which is satisfied when W is a concave function and each utility function U_i is concave and non-decreasing in d_i, or when W is a convex function and each utility function U_i is convex and non-increasing in d_i.

With full information, solving for the optimal decision is straightforward. We consider the case where some parameters of the decision model are not available and need to be estimated using data. Let $\mathbf{r}$ denote unknown parameters, and we suppose these parameters only exist in the decision objective. For clarity, we denote the objective function as $W(\mathbf{d};\mathbf{r})$ to emphasize its dependence on both the decision variables and the unknown parameters. Note that the feasible region $\mathcal{S}$ does not depend on $\mathbf{r}$. In the *full information* case, $\mathbf{r}$ is known, and we represent the true optimal fair solution to (1) with $\mathbf{d}^*(\mathbf{r})$. In the *limited information* case, $\mathbf{r}$ is unknown, we aim to make a data-driven decision, $\hat{\mathbf{d}}$, that is as close to the true decision as possible. To generate this decision, we can follow a **separate view** to first predict the unknown parameters with $\hat{\mathbf{r}}$ then solve the decision optimization model to obtain $\hat{\mathbf{d}} \coloneqq \mathbf{d}^*(\hat{\mathbf{r}})$, or adopt a **direct view** to predict the solution $\hat{\mathbf{d}}$ without explicit parameter estimation.

Let $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ represent available data that contain features relevant to $\mathbf{r}$ or $\mathbf{d}^*(\mathbf{r})$. Due to potential historical biases in $\mathcal{D}$ and the underlying association with parameters or decisions, estimation of $\hat{\mathbf{r}}$ or $\hat{\mathbf{d}}$ using this dataset might perpetuate or amplify unfairness. Let F represent prediction fairness criteria to incorporate. Besides optimizing the utility-based fairness in decision, we also require prediction models to attain satisfactory performance in terms of F, e.g., if F measures a type of undesirable disparity in $\hat{\mathbf{r}}$, then $F(\hat{\mathbf{r}})$ should be sufficiently low. So far, we have formulated the **end-to-end fairness optimization** (E2EFO) problem. Next, we describe two-stage PTO and FDFL as two distinctive frameworks for solving E2EFO.

A two-stage PTO approach, following the separate view, first estimates unknown parameters and then optimizes decisions. The prediction task uses $(\mathcal{D}, \mathbf{r})$ as training data to learn a prediction model f_θ parameterized by $\theta \in \Theta$. Given a feature vector $\mathbf{x}_i$, the model produces parameter estimates $\hat{\mathbf{r}}_i \coloneqq f_\theta(\mathbf{x}_i)$. To train a predictor while accounting for fairness, all fair supervised ML methods, as reviewed in Section 2, are applicable. We focus on in-processing methods which incorporate fairness constraints or regularization into standard prediction-error minimization. Let $L(\theta)$ denote the standard loss function, e.g., mean squared error in regression, and $F(f_\theta(\mathbf{x}), \mathcal{D})$ capture disparity or unfairness to reduce. The training problem can be formulated with a fairness constraint, i.e., $\min_{\theta\in\Theta} L(\theta)$ s.t. $F(f_\theta(\mathbf{x}), \mathcal{D}) \leq \epsilon$ where ϵ is a predefined tolerance level, or fairness in regularization, i.e., $\min_{\theta\in\Theta} L(\theta) + \lambda F(f_\theta(\mathbf{x}), \mathcal{D})$ where the hyperparameter $\lambda > 0$ regulates the trade-off between prediction accuracy and fairness. After training f_θ, decisions are obtained by solving for $\mathbf{d}^*(\hat{\mathbf{r}})$ from (1).

3.1 Fair Decision Focused Learning

We propose Fair Decision Focused Learning (FDFL) as an end-to-end framework that directly optimizes decision quality while accounting for fairness. FDFL aims to generate fair decisions $\hat{\mathbf{d}}$ that are close to the unknown true optimal fair decisions, $\mathbf{d}^*(\hat{\mathbf{r}})$. Higher decision accuracy provides better fairness performance, as measured by $W(\hat{\mathbf{d}};\mathbf{r})$. Within FDFL, there are two primary ways to integrate fair prediction as part of fair optimization.

Separate View. FDFL uses the training data $(\mathcal{D}, \mathbf{r})$ to learn a parametric predictor f_θ for estimating $\hat{\mathbf{r}} \coloneqq f_\theta(\mathbf{x})$. The final decisions are determined separately by solving (1) with $\hat{\mathbf{r}}$ plugged in as parameters. The training model needs to reflect the goal of seeking decision accuracy, and there are different ways to define the training loss function. First, a training model can seek a direct maximization of the decision objective, namely, to minimize the training loss $L_{decision}(\theta) = -W(\mathbf{d}^*(f_\theta(\mathbf{x}));\mathbf{r})$. We can also define the training problem to minimize the decision regret, $L_{regret}(\theta) = W(\mathbf{d}^*(f_\theta(\mathbf{x}));\mathbf{r}) - W(\mathbf{d}^*(\mathbf{r});\mathbf{r})$, or minimize the mean square error between decisions, $L_{error}(\theta) = \|\mathbf{d}^*(f_\theta(\mathbf{x})) - \mathbf{d}^*(\hat{\mathbf{r}})\|^2$. It is easy to observe that all three training losses become zero when decisions are fully accurate, namely $\mathbf{d}^*(f_\theta(\mathbf{x})) = \mathbf{d}^*(\mathbf{r})$. We also remark that standard DFL methods have been proposed using these training loss definitions [43].

Let $L_{DFL}(\mathbf{d}^*(f_\theta(\mathbf{x})))$ denote one of the above loss functions. We again use $F(f_\theta(\mathbf{x}), \mathcal{D})$ as a prediction fairness criterion measuring the disparity to be mitigated. Then training under a separate view involves finding optimal θ in the following models:

$$\begin{aligned} \text{Prediction fairness as constraint: } & \min_{\theta \in \Theta} L_{DFL}(\mathbf{d}^*(f_\theta(\mathbf{x}))) \text{ s.t. } F(f_\theta(\mathbf{x}), \mathcal{D}) \leq \epsilon; \\ \text{Or prediction fairness in regularization: } & \min_{\theta \in \Theta} L_{DFL}(\mathbf{d}^*(f_\theta(\mathbf{x}))) + \lambda F(f_\theta(\mathbf{x}), \mathcal{D}). \end{aligned} \tag{2}$$

Direct View. Alternatively, FDFL learns from the training data $\mathcal{D}, \mathbf{d}^*(\mathbf{r})$ (a solver of (1) is required to generate $\mathbf{d}^*(\mathbf{r})$) to obtain a decision predictor M_θ, from which $\hat{\mathbf{d}} \coloneqq M_\theta(\mathbf{x})$. The previous training loss definitions also apply in this setup. For example, the decision objective loss can be written as $L_{decision}(\theta) = -W(M_\theta(\mathbf{x}); \mathbf{r})$. We use $L_{DFL}(M_\theta(\mathbf{x}))$ to denote a training loss function. Training under a direct view solves the following models.

$$\begin{aligned} \text{Prediction fairness as constraint: } & \min_{\theta \in \Theta} L_{DFL}(M_\theta(\mathbf{x})) \text{ s.t. } F(M_\theta(\mathbf{x}), \mathcal{D}) \leq \epsilon; \\ \text{Or prediction fairness in regularization: } & \min_{\theta \in \Theta} L_{DFL}(M_\theta(\mathbf{x})) + \lambda F(M_\theta(\mathbf{x}), \mathcal{D}). \end{aligned} \tag{3}$$

This approach bypasses explicit parameter estimation. The training problem can be viewed as fitting a *policy* or *decision map* that yields the best fair decisions from the input features. As illustrated in standard DFL literature, e.g., [48], M_θ acts as a smooth surrogate function to approximate the optimal decision objective and is especially useful to support efficient end-to-end learning when (1) is expensive to solve or has non-differentiable solutions.

Compared to PTO which emphasizes accuracy of $\hat{\mathbf{r}}$, FDFL prioritizes the accuracy of prediction-based decisions $\hat{\mathbf{d}}$ to the true decisions $\mathbf{d}^*(\mathbf{r})$. Although both frameworks could integrate the same prediction and decision fairness components, their different training models lead to different fairness performances. In Section 3.3, we present an example where two-stage PTO is insufficient to attain satisfactory decision fairness. Before this example, we explain further details about defining the decision fairness objective.

3.2 Utility-based Decision Fairness

A decision, such as resource allocation, provides benefits or harms to the decision recipient. The overall impact can be quantified by a utility function. The definition of utility functions varies with the decision contexts to reflect the needed impact assessment. We construct a generic class of utility functions that unify a broad range of utility specifications commonly studied in the literature. As discussed in the problem formulation, d_i denotes the decision received by the stakeholder i. In addition, we suppose a stakeholder is characterized by three values: $q_i \in \mathbb{R}_{\geq 0}$, the level of need or desert for the decision of interest; $g_i \in \mathbb{R}_{\geq 0}$, the utility gain rate for the received decision; $a_i \in \mathbb{R}_{\geq 0}$, the base utility level before receiving any decision. Note that q_i accounts for all factors influencing how strongly the stakeholder requests, desires or needs the decision. For example, in the allocation of medical treatments, patients with higher medical risks would have higher desert levels. The other two values, g_i and a_i, quantify a stakeholder's utilities by distinguishing the decision-induced change and the starting position. Variations in these values reflect stakeholders' inherent differences. To continue with the previous example, after receiving the same treatment, two patients with different health levels may experience different health improvements.

For a fixed decision, it should provide a greater utility for stakeholders who need or deserve the decision more strongly (i.e., a larger q_i), gain utilities more effectively (i.e., a larger g_i), or are better off to start with (i.e., a larger a_i). Following this intuition, we compute i's utility from decision d_i as:

$$U_i(d_i) = q_i(g_i d_i + a_i)^\rho. \tag{4}$$

In this function, $\rho \in \mathbb{R}_+$ regulates the shape and rate of utility changes. When $\rho < 1$, the utility shows diminishing return with respect to decision, that is, utility increases more slowly as the decision amount increases. When $\rho = 1$, the utility function is linear, namely, a constant return rate from decision. When $\rho > 1$, the utility has increasing return. While we focus on a single decision for each stakeholder, we remark that this utility definition can be generalized to represent multiple decision types. For recipient i, let $\mathbf{d}_i = (d_i^1, \ldots, d_i^m)$ denote i's assigned decisions for all m types. Such heterogeneous decisions can characterize the allocation of multiple resources. The need/desert level, utility gain rate and base level for a recipient similarly extend to vectors: $\mathbf{q}_i = (q_i^1, \ldots, q_i^m)$, $\mathbf{g}_i = (g_i^1, \ldots, g_i^m)$, $\mathbf{a}_i = (a_i^1, \ldots, a_i^m)$. The power parameter ρ also can vary across decision types. The general utility definition for recipient i becomes: $U_i(\mathbf{d}_i) = \sum_{j=1}^m q_i^j (g_i^j d_i^j + a_i^j)^{\rho_j}$.

Given a utility distribution $\mathbf{u}$ generated from decisions $\mathbf{d}$, a fairness measure W aggregates utility values into a single scalar indicator of the desirability of $\mathbf{u}$ with respect to fairness. As reviewed in Section 2, there are many fairness measures that fall into three broad categories: fairness via equality, fairness for the disadvantaged, and combined measures balancing fairness and efficiency. Recall that we focus on W that gives a concave function of $\mathbf{d}$, which requires W to be convex or concave in $\mathbf{u}$ depending on the shape of utility functions.

Here, we highlight one class of concave fairness measures from literature. α-fairness (5) is a widely studied fairness measure in optimization and mechanism design that balances fairness and efficiency. When $\alpha = 0$, the metric characterizes pure efficiency, as it seeks to maximize the total utility without consideration for fairness. The function assigns greater emphasis on fairness as α increases. At the other extreme when α approaches infinity, α-fairness reduces to the Rawlsian maximin fairness criterion, which prioritizes the worst-off utility to be maximized.

$$W_\alpha(\mathbf{u}) = \begin{cases} \frac{1}{1-\alpha} \sum_{i=1}^n u_i^{1-\alpha}, & \text{if } \alpha \geq 0 \text{ and } \alpha \neq 1; \\ \sum_{i=1}^n \log(u_i), & \text{if } \alpha = 1. \end{cases} \tag{5}$$

The general compatibility with E2EFO does not mean all fairness measures are practical. As we will discuss further in Section 4, different fairness objectives lead to decision-making models of varying complexity, thus affecting the difficulty of designing FDFL algorithms. In a fairness measure that serves as the decision objective, all unknown parameters are included in $\mathbf{r}$. There may be multiple groups of unknown parameters. For instance, when specifying utility functions of resource allocation decisions, it is possible we do not know stakeholders' exact need levels and utility gain rates, but have access to stakeholder features that can be used to estimate these values.

3.3 An Illustrative Example

We use a simple example to illustrate the prediction and decision performance differences between two-stage versus end-to-end methods. Suppose 50 units of resources need to be distributed among 15 people belonging to two groups. Group 1, as the majority group, consists of 10 people. Group 2, as the minority group, has 5 people. Each person is characterized by a single feature $x_i \in [0, 1]$, a need level q_i and a utility gain rate $g_i \in [0, 1]$ for the resource. We generate the data so that Group 2 is the disadvantaged group. To give a concrete context, we can consider the resource as budget to take physical exams, $\{x_i\}$ as people's age, $\{q_i\}$ as overall health risks (older people tend to have higher risks). Due to historical bias, current data tend to under-report the risks of group 2 at older ages and over-report at younger ages.

From receiving resource d_i, person i's utility is defined as $u_i = q_i g_i d_i$. The decision maker aims to allocate all resources to maximize an α-fairness objective with $\alpha = 2$ to reflect a strong emphasis on fairness over efficiency subject to the resource capacity constraint. Namely, the decision optimization model can be stated as: $\max_{\mathbf{d}} \frac{1}{1-2} \sum_{i=1}^{n} u_i^{1-2}$ s.t. $u_i = q_i g_i d_i \ \forall i, \sum_{i=1}^{n} d_i \leq 50, \mathbf{d} \geq 0$. At the time of decision, the need levels $\{q_i\}$ is unknown, but can be estimated using the feature $\{x_i\}$. This means $\mathbf{r} \coloneqq \mathbf{q}$ in the example. For the prediction task to generate $\hat{\mathbf{r}}$, we consider a linear regression model for prediction, and quantify accuracy disparity, i.e., the difference between two groups' mean square error (MSE) of $\mathbf{r}, \hat{\mathbf{r}}$, as the prediction unfairness to reduce.

On this example, we compare four models for fitting the linear regression line. Two of the models adopt a two-stage approach to focus on prediction accuracy in regression: the first one (M1) runs a standard linear regression to minimize the MSE between $\mathbf{r}, \hat{\mathbf{r}}$, and the second one (M2) accounts for prediction fairness by adding accuracy disparity as a regularization term to prediction error. The other two models follow an end-to-end view to emphasize decision accuracy: the third model (M3) fits a regression line to minimize the MSE between $\mathbf{d}^*(\mathbf{r}), \mathbf{d}^*(\hat{\mathbf{r}})$ (i.e., L_{error}), and the last model (M4) adds the accuracy disparity regularization to the decision MSE. We note that the last model accounts for fairness in both decision and prediction. In Figure 1, the green and blue points display the features and the true need levels, and the lines show the regression results from all four models. As we expect, the standard linear regression model reaches the best prediction accuracy, and the end-to-end regression model without prediction fairness attains the highest decision accuracy (Table 1). We also highlight that accounting for both types of fairness in an end-to-end manner significantly benefits prediction performance while maintaining decision accuracy. In Figure 2, we observe additional decision differences between two-stage and end-to-end approaches. Both two-stage models carry over some discriminative bias in predictions to decisions, as the decision errors are higher for the disadvantaged group 2. Accounting for prediction fairness in M2 helps reducing the gap between groups, but is insufficient to fully remove the bias. In contrast, through optimizing decision fairness, both end-to-end models attain more accurate decisions for group 2 while getting closer to the true optimal fair decisions.

	M1	M2	M3	M4
$MSE(\mathbf{r}, \hat{\mathbf{r}})$	**1.283**	1.680	70.274	2.710
$MSE(\mathbf{d}^*(\mathbf{r}), \mathbf{d}^*(\hat{\mathbf{r}}))$	0.314	0.201	**0.186**	0.188

Table 1. Results from Four Linear Regression Fitting Models

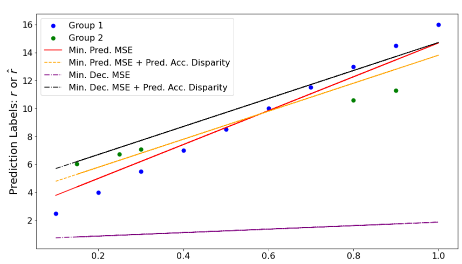

Fig. 1. Data points and Linear Regression Results

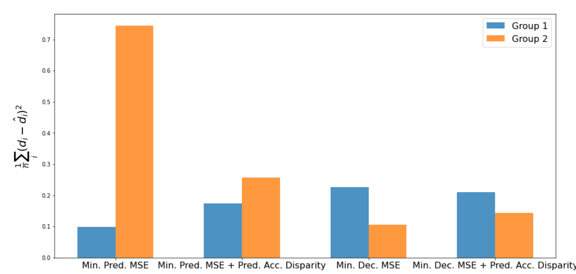

Fig. 2. Errors in Decisions based on Different Regression Models

4 Fair Decision Focused Learning Algorithm

In Section 3.1, we have introduced the FDFL framework and described possible training formulations that characterize the learning goals of FDFL. Building on these foundations, we now detail the algorithmic steps for training the predictor with data. Specifically, we focus on the separate view with $L_{DFL} \coloneqq L_{regret}$ and use regularization to incorporate prediction fairness. Therefore, we develop FDFL algorithm with the following training model.

$$\min_{\theta \in \Theta} W(\mathbf{d}^*(f_\theta(\mathbf{x})); \mathbf{r}) - W(\mathbf{d}^*(\mathbf{r}); \mathbf{r}) + \lambda F(f_\theta(\mathbf{x}), \mathcal{D}). \quad (6)$$

We note that this training loss is generally not convex in θ due to its dependence on $\mathbf{d}^*(f_\theta(\mathbf{x}))$, which lacks consistent structure as it needs to be solved from decision optimization (1) and the target model f_θ can be nonlinear and non-convex. As a generic technique, we can apply the gradient descent algorithm to solve the training problem. Algorithm 1 provides the pseudocode for our FDFL algorithm. In each training epoch, a forward propagation step evaluates the loss function value, then a back propagation step computes the gradient $\nabla_\theta \mathcal{L}$ and applies a gradient-based update to predictor parameters. Following the convention of standard DFL literature, Algorithm 1 is a type of gradient-based method.

4.1 Gradient-Based Method

The key computational challenge of Algorithm 1 is gradient computation. Since $\mathcal{L}(\theta) = L_{regret}(\theta) + \lambda F(\theta)$, we have $\nabla_\theta \mathcal{L} = \nabla_\theta \mathcal{L}_{regret}(\theta) + \lambda \nabla_\theta \mathcal{L} F(\theta)$. Using the chain rule, we can decompose the gradient calculation as follows.

$$\frac{\partial L_{Regret}(\theta)}{\partial \boldsymbol{\theta}} = \frac{\partial L_{Regret}(\mathbf{d}^*(\hat{\mathbf{r}}), \cdot)}{\partial \mathbf{d}^*(\hat{\mathbf{r}})} \cdot \frac{\partial \mathbf{d}^*(\hat{\mathbf{r}})}{\partial \hat{\mathbf{r}}} \cdot \frac{\partial \hat{\mathbf{r}}}{\partial \boldsymbol{\theta}}; \ \frac{\partial F(\theta)}{\partial \boldsymbol{\theta}} = \frac{\partial F(\hat{\mathbf{r}}, \cdot)}{\partial \hat{\mathbf{r}}} \cdot \frac{\partial \hat{\mathbf{r}}}{\partial \boldsymbol{\theta}}. \tag{7}$$

In the decomposition, the term that is the most difficult to handle is $\frac{\partial \mathbf{d}^*(\hat{\mathbf{r}})}{\partial \hat{\mathbf{r}}}$, which involves differentiating through the solver to the decision optimization problem. The other terms are generally easy to compute. We note that $\frac{\partial L_{Regret}(\mathbf{d}^*(\hat{\mathbf{r}}),\cdot)}{\partial \mathbf{d}^*(\hat{\mathbf{r}})} = \frac{\partial W(\mathbf{d}^*(\hat{\mathbf{r}}),\mathbf{r})}{\partial \mathbf{d}^*(\hat{\mathbf{r}})}$, which is straightforward as long as the decision objective $W(\mathbf{d}; \mathbf{r})$ is differentiable with respect to $\mathbf{d}$. Similarly, when the prediction fairness measure F is differentiable with respect to $\hat{\mathbf{r}}$, we can conclude $\frac{\partial F(\hat{\mathbf{r}},\cdot)}{\partial \hat{\mathbf{r}}}$. Lastly, $\frac{\partial \hat{\mathbf{r}}}{\partial \boldsymbol{\theta}}$ is the gradient of the predictor model with respect to the model parameters, which can be generated by modern machine learning frameworks.

Next, we explain our strategy for generating $\frac{\partial \mathbf{d}^*(\hat{\mathbf{r}})}{\partial \hat{\mathbf{r}}}$. This gradient can be approximated using a finite difference approximation method inspired by the differentiable black-box optimizer (DBB) approach for standard DFL with a linear optimization decision model [37]. Given the current predicted parameters $\hat{\mathbf{r}}$, we introduce a perturbation using the gradient of decision objective with respect to decisions, $\mathbf{r}' = \hat{\mathbf{r}} + \epsilon \frac{\partial W(\mathbf{d}^*(\hat{\mathbf{r}}),\mathbf{r})}{\partial \mathbf{d}^*(\hat{\mathbf{r}})}$. Then we can approximate the gradient at $\hat{\mathbf{r}}$ as:

$$\frac{\partial \mathbf{d}^*(\hat{\mathbf{r}})}{\partial \hat{\mathbf{r}}} \approx \frac{1}{\epsilon}(\mathbf{d}^*(\mathbf{r}') - \mathbf{d}^*(\hat{\mathbf{r}})). \tag{8}$$

Here, $\epsilon > 0$ is a perturbation parameter that controls the approximation accuracy. This approach provides non-zero approximate gradients and requires only two calls to decision solver (for generating $\mathbf{d}^*(\mathbf{r}'), \mathbf{d}^*(\hat{\mathbf{r}})$) per gradient computation. Notably, this finite difference approximation is applicable to all fair decision optimization models as long as a decision solver is available.

There are many other gradient approximation methods proposed in standard DFL as we have discussed in section 2. These methods often require linearity of the decision model, and do not extend easily to handle nonlinear fairness optimization. For completeness, in Appendix C, we discuss other gradient computation techniques for the special case where the decision fairness objective is linear in $\mathbf{r}$ and $\mathbf{d}$.

4.1.1 *Closed-Form Decisions and Analytical Gradients for α-Fairness Optimization.*

For special cases where closed-form formulas for $\frac{\partial \mathbf{d}^*(\hat{\mathbf{r}})}{\partial \hat{\mathbf{r}}}$ can be derived to support exact computation, the above gradient approximation is not needed. We demonstrate this strategy for a specific class of α-fairness optimization problems.

In an E2EFO problem, we consider a decision task to allocate $Q > 0$ units of resources among n stakeholders to maximize the α-fairness (5) of decision utilities. The unknown parameters $\mathbf{r}$ represent the need levels $\mathbf{q}$. The decisions

Algorithm 1 Fair Decision Focused Learning with Gradient Descent

Require: Training data $\mathcal{D} = \{\mathbf{x}\}$; Target labels $\{\mathbf{r}\}$; decision objective $W(\mathbf{d};\mathbf{r})$; prediction fairness regularizer $F(\theta)$; learning rate η.

Initialize predictor parameter $\theta \in \Theta$ for f_θ.
for each training epoch **do**
  $\hat{\mathbf{r}} \leftarrow f_\theta(\mathbf{x})$. ▷ Predict parameters in decision model
  $\mathbf{d}^*(\hat{\mathbf{r}}) \leftarrow \arg\max_{\mathbf{d}\in\mathcal{S}} W(\mathbf{d};\hat{\mathbf{r}})$. ▷ Solve optimal decision based on predicted parameters
  $\mathcal{L}(\theta) \leftarrow W(\mathbf{d}^*(\hat{\mathbf{r}});\mathbf{r}) - W(\mathbf{d}^*(\mathbf{r});\mathbf{r}) + \lambda F(\theta)$. ▷ Evaluate loss (end of forward propagation)
  Compute $\nabla_\theta \mathcal{L}$ and update $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$. ▷ Backward propagation
end for
return θ

contribute linearly to utilities, that is, $u_i = r_i g_i d_i$ for all $i \in [n]$. Formally, the optimization problem is stated as:

$$\max_{\mathbf{d}} \sum_{i=1}^{n} W_\alpha(r_i g_i d_i) \quad \text{s.t.} \quad \sum_{i=1}^{n} c_i d_i \leq Q, \quad \mathbf{d} \geq 0. \tag{9}$$

We assume $\mathbf{r}, \mathbf{g} \in \mathbb{R}_+^n$ to ensure that all stakeholders receive positive utilities from the assigned resources. We note that the example in Section 3.3 considers this decision model. (9) is a convex programming model, and its optimal solution can be derived in closed form using the KKT conditions. Proposition 4.1 states the solution formula $\mathbf{d}^*(\mathbf{r})$ for the case where $\alpha > 0$ and $\alpha \neq 1$. Differentiating the solutions with respect to the input parameters leads to the analytical gradient formulas given in Proposition 4.2. In Appendix D, we state these formulas for the special cases where $\alpha = 0$ (efficiency), $\alpha = 1$ (proportional fairness) and $\alpha \to \infty$ (Rawlsian maximin fairness), and provide the complete proofs.

Proposition 4.1 (Closed-Form Decisions). *The optimal solution to* (9) *with parameter* $\mathbf{r}$ *is given by:*

$$d_i^* = \frac{c_i^{-\frac{1}{\alpha}} \cdot (r_i g_i)^{\frac{1}{\alpha}-1} \cdot Q}{\sum_{j=1}^{n} c_j^{1-\frac{1}{\alpha}} \cdot (r_j g_j)^{\frac{1}{\alpha}-1}}, \ \forall i \in [n].$$

Proposition 4.2 (Analytical Gradient Formulas). *Let* $\mathbf{d}^*$ *denote the optimal solution to* (9) *with parameter* $\mathbf{r}$. *The partial derivatives of the optimal decision* d_i^* *with respect to* $\mathbf{r}$ *are:*

$$\frac{\partial d_i^*}{\partial r_i} = \frac{Q\left(-1+\frac{1}{\alpha}\right) c_i^{-\frac{1}{\alpha}} \cdot g_i^{\frac{1}{\alpha}-1} \cdot r_i^{-2+\frac{1}{\alpha}} \left(S - c_i^{1-\frac{1}{\alpha}} \cdot (r_i g_i)^{-1+\frac{1}{\alpha}}\right)}{S^2};$$

$$\frac{\partial d_i^*}{\partial r_k} = \frac{-Q\left(-1+\frac{1}{\alpha}\right) c_i^{-\frac{1}{\alpha}} \cdot (r_i g_i)^{\frac{1}{\alpha}-1} \cdot r_k^{-2+\frac{1}{\alpha}} \cdot c_k^{1-\frac{1}{\alpha}} \cdot g_k^{-1+\frac{1}{\alpha}}}{S^2} \ \forall k \neq i, k \in [n].$$

where $S = \sum_{j=1}^{n} c_j^{1-\frac{1}{\alpha}} \cdot (r_j g_j)^{-1+\frac{1}{\alpha}}$.

5 Experiments

5.1 Problem Setting

We evaluate the FDFL framework and algorithms on a healthcare resource allocation problem. We work with a synthetic data generated in [34] to replicate real-world medical data without protected information. The dataset contains 48,784 patient records with demographic information, comorbidities, historical healthcare costs, biomarkers, risk scores and program enrollment decisions. The *risk scores* are generated from a commercial medical risk prediction algorithm to help identify patients that should be enrolled in *high-risk care management* programs. Currently, program enrollment decisions follow a two-stage process: first, the commercial algorithm generates risk scores from medical data. Then patients scoring above the 97th percentile are automatically enrolled, while those above the 55th percentile are referred to their primary care physicians for enrollment consideration based on the risk score and additional contextual information. The original study [34] revealed significant racial bias in commercial risk prediction algorithms: at equivalent risk scores, Black patients were sicker than White patients but less likely to be identified for program enrollment.

This risk prediction and enrollment decision problem fits in our E2EFO paradigm. The prediction task needs to estimate patients' medical risks from patient records in an unbiased way. Instead of the current threshold based enrollment progress, we consider a utility-based α-fairness optimization model to determine enrollment. For continuity of decisions, we also modify allocating binary enrollment decisions to allocating continuous decisions. For a patient i, let d_i denote the received resource (more resource means enrolling in more or better programs). We model the patient utility as $u_i(d_i) = r_i g_i d_i$, where r_i is the patient's risk score (a higher risk means greater needs for programs), g_i is a gain factor representing how much the patient would benefit from program enrollment. In addition, the decision faces a cost constraint: providing resource to each patient i requires a unit cost c_i, and there is a total budget of Q for all patients. The decision optimization problem is given in (9).

From the dataset, we use the algorithm generated risk scores as $\mathbf{r}$ and view these values as unknown at the time of enrollment decision. The gain factor $\mathbf{g}$ is not directly observed and we need to create reasonable estimates using the available information. We observe that the dataset contains a portion of patients that are enrolled in program, and these patients have avoidable costs indicating their actual cost saving from enrollment. For these patients, the avoidable costs can be used as g_i. For other patients with missing avoidable costs, we fill in the missing values via nearest neighbor matching with the enrolled patients, then use the generated values as their corresponding g_i.

5.2 Methods and Implementation

We compare our FDFL framework with two-stage methods to assess difference in decision quality, prediction performance, and fairness. For two-stage methods, we consider two variants of prediction, without and with prediction fairness regularization, followed by the same fairness optimization decision. Similarly, we implement FDFL to optimize decision fairness without and with prediction fairness considerations. The experiments use a random sample of $n = 5,000$ patients (11.36% are Black) from the entire dataset and consider a resource allocation problem with total budget $Q = 1,000$. Resource costs follow a normal distribution $c \sim \mathcal{N}(1, 0.5)$. To examine how different fairness preferences affect outcomes, we evaluate the models across multiple values of the fairness parameter: $\alpha \in \{0.5, 1.5, 2, \infty\}$.

We implement the following approaches: (1) **Vanilla two-stage**, (2) **Two-Stage with prediction accuracy disparity**, (3) **Vanilla FDFL**, (4) **FDFL with prediction accuracy disparity**. When prediction accuracy disparity is included, $\lambda = 1$, otherwise $\lambda = 0$. Note that all four approaches account for decision fairness through the α-fairness objective. We leveraged *PyEPO* [43]for implementing the training pipeline with both closed-form gradients and DBB (via finite difference) based approximate gradients in FDFL. We use two prediction models: a linear regression model and a neural network with two hidden layers (64 units each). For all methods, we used closed-form to solve the decision problem. For other fairness measure and utility functions without closed-form solution one can employ commercial solvers to determine optimal decisions.

5.3 Results

All methods are evaluated across prediction accuracy (MSE), decision quality (normalized regret computed as $\frac{L_{regret}(\hat{\mathbf{r}},\mathbf{r})}{|W(\mathbf{d}(\hat{\mathbf{r}}),\mathbf{r})|}$ [43]) and prediction accuracy disparity between racial groups. Table 2 presents a comprehensive comparison of regret

Table 2. Normalized Regret (Mean ± Std) Across Models and α Values

Method	$\alpha = 0.5$ LR	$\alpha = 0.5$ NN	$\alpha = 1.5$ LR	$\alpha = 1.5$ NN	$\alpha = 2.0$ LR	$\alpha = 2.0$ NN	$\alpha = \infty$ LR	$\alpha = \infty$ NN
2-Stage ($\lambda = 0$)	0.13 ± 0.03	**0.08 ± 0.02**	0.28 ± 0.04	0.41 ± 0.05	4.46 ± 0.59	10.59 ± 0.55	0.28 ± 0.01	0.41 ± 0.04
2-Stage ($\lambda = 1$)	0.11 ± 0.03	**0.08 ± 0.03**	0.31 ± 0.05	0.41 ± 0.04	6.74 ± 0.93	10.71 ± 0.80	0.31 ± 0.04	0.40 ± 0.03
FDFL-CF ($\lambda = 0$)	0.14 ± 0.04	0.09 ± 0.02	0.29 ± 0.06	**0.22 ± 0.05**	4.02 ± 1.71	**1.23 ± 0.45**	0.33 ± 0.03	**0.29 ± 0.04**
FDFL-CF ($\lambda = 1$)	0.14 ± 0.04	0.09 ± 0.02	0.31 ± 0.07	**0.24 ± 0.07**	5.01 ± 2.29	**1.21 ± 0.43**	0.33 ± 0.10	**0.34 ± 0.07**
FDFL-DBB ($\lambda = 0$)	0.11 ± 0.04	14.29 ± 10.98	0.52 ± 0.14	0.49 ± 0.10	17.79 ± 9.05	5.61 ± 0.50	0.41 ± 0.10	0.51 ± 0.12
FDFL-DBB ($\lambda = 1$)	0.11 ± 0.04	17.71 ± 11.02	0.47 ± 0.12	0.45 ± 0.08	13.70 ± 8.79	5.54 ± 0.87	0.47 ± 0.12	0.46 ± 0.06

performances across different methods, models, and α values controlling the fairness-efficiency tradeoff in the decision objective. The results reveal several patterns. At $\alpha = 0.5$, the decision objective emphasizes efficiency over fairness, so the advantages of FDFL versus two-stage methods are relatively small. The closed-form solution formulas in Proposition 4.1 indicate that the decision accuracy is aligned with the prediction accuracy, that is, accurate predictions guarantee accurate decisions. In this case, both two-stage methods and FDFL with closed-form gradients, which respectively pursue prediction accuracy and decision accuracy, attain comparable low regrets from both linear regression and neural network predictor models. The FDFL-DBB methods using approximate gradients lead to lower decision quality due to less accurate gradients. As we increase α to shift emphasis towards fairness, FDFL begins to show advantages over the two-stage approaches, particularly when training a neural network using closed-form gradients. This trend is unsurprising, as we observe from the closed-form solutions that prediction accuracy and decision accuracy become less aligned at larger α. As α increase, the performance gap widens, and we observe the largest FDFL advantage at $\alpha = 2$. We next provide more details about these results. In Appendix B, we present detailed results for the other α values.

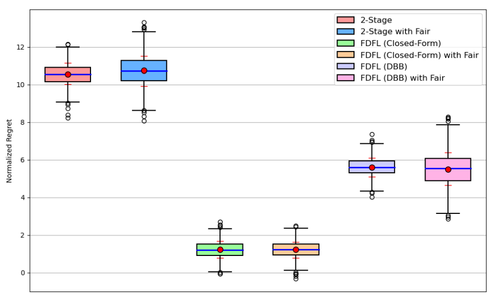

Fig. 3. Decision Accuracy via Normalized Regret for $\alpha = 2$

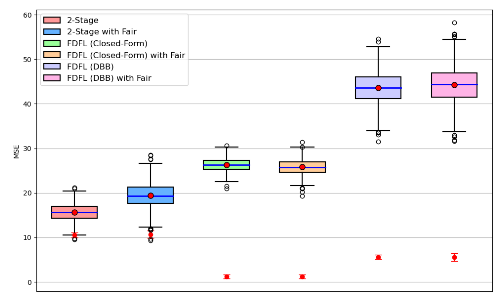

Fig. 4. Prediction Accuracy via MSE for $\alpha = 2$

Table 3. Neural Network Results (Mean ± Std) For $\alpha = 2$

Method	MSE	Regret	Accuracy Parity	Time/Epoch (seconds)
2-Stage ($\lambda = 0$)	**15.53 ± 1.86**	10.59 ± 0.55	**11.106 ± 6.060**	0.33 ± 0.03
2-Stage ($\lambda = 1$)	**19.30 ± 2.72**	10.71 ± 0.80	**1.205 ± 1.087**	0.31 ± 0.03
FDFL-CF ($\lambda = 0$)	26.26 ± 1.42	**1.23 ± 0.45**	17.040 ± 7.380	1.32 ± 0.12
FDFL-CF ($\lambda = 1$)	25.81 ± 1.70	**1.21 ± 0.43**	16.238 ± 6.881	1.29 ± 0.17
FDFL-DBB ($\lambda = 0$)	43.67 ± 3.69	5.61 ± 0.50	31.589 ± 15.180	21.49 ± 0.76
FDFL-DBB ($\lambda = 1$)	44.26 ± 4.20	5.54 ± 0.87	32.181 ± 15.147	21.45 ± 1.29

Table 3 summarizes all important evaluation metrics for the methods at $\alpha = 2$. In addition, Figures 3 and 4 visualize normalized regret and MSE values for clearer comparison of decision and prediction performances. All four FDFL variants attain better decision quality, as indicated by lower regrets, than two-stage methods. In contrast, two-stage methods are better in terms of prediction quality. These results validate that FDFL methods prioritize decision accuracy to optimize the end-to-end decision goals. Among FDFL methods, closed-form gradients outperform approximate gradients in all aspects, but FDFL-DBB using approximate gradients still demonstrate effectiveness over two-stage methods. Finally, we examine the effects of incorporating prediction fairness. In the two-stage approaches, adding a fairness penalty improves the fairness metric but often increases MSE. In FDFL, the prediction fairness regularization provides moderate improvement in MSE, regret and accuracy parity in the closed-form gradient based version, but minimal benefits for the DBB version. These results imply careful investigation of different ways to incorporate prediction fairness is needed in further FDFL studies.

6 Conclusion

This study introduces Fair Decision-Focused Learning (FDFL), a robust framework for end-to-end fairness optimization (E2EFO) in data-driven decision-making. By integrating prediction and optimization tasks, FDFL addresses the shortcomings of traditional two-stage methods, enabling direct alignment between prediction accuracy, decision quality, and fairness. Our empirical results demonstrate that FDFL achieves superior decision outcomes and fairness, particularly when the decision optimizes an α-fairness objective with higher α to emphasize fairness, while maintaining computational efficiency through closed-form solutions and gradient-based methods. Our findings highlight the potentials of FDFL as a versatile and impactful framework for promoting fairness over two-stage PTO methods in practical applications.

Our paper establish FDFL as a promising framework for integrating fairness into data-driven decision making. There are many open questions in theory, algorithm and practical aspects that are worthwhile to investigate. In this study, while the E2EFO formulation allows flexible utility and fairness definitions, our algorithm design and experiments focus on linear utility functions and α-fairness objective. This restricts the applicability to problems with more complex fairness measures or utility structures. Developing efficient FDFL algorithms to handle diverse fairness metrics, such as axiomatic fairness [27], would greatly expand the applicability of FDFL to address equity challenges in broader decision contexts. Our FDFL algorithms rely on gradient-based approaches, which are effective only when gradients can be computed reliably and efficiently. The closed-form gradient approach, though computationally efficient, is limited to specific problem classes. For general fairness optimization decisions, there are significant potentials for developing scalable methods that leverage problem-specific structures. Additionally, incorporating gradient-free methods could extend FDFL's applicability to non-differentiable or discrete optimization settings. Future works could explore these alternative techniques to propose more advanced FDFL algorithms. Lastly, theoretical understanding of FDFL, such as the interaction between prediction and decision fairness, the impact of different fairness requirements on the performance differences between FDFL and two-stage methods, remains largely unexplored. A deeper theoretical investigation could not only provide rigor for the empirical evidence of FDFL's effectiveness in fair decision-making, but also generate insights into end-to-end fairness in data-driven decisions in general.

References

[1] Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J Zico Kolter. 2019. Differentiable convex optimization layers. *Advances in neural information processing systems* 32 (2019).

[2] Ibrahim Alabdulmohsin. 2020. Fair classification via unconstrained optimization. *arXiv preprint* 2005.14621 (2020).

[3] Brandon Amos and J Zico Kolter. 2017. Optnet: Differentiable optimization as a layer in neural networks. In *International conference on machine learning*. PMLR, 136–145.

[4] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. 2021. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research* 50, 1 (2021), 3–44.

[5] Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. 2020. Learning with differentiable pertubed optimizers. *Advances in neural information processing systems* 33 (2020), 9508–9519.

[6] Dimitris Bertsimas and Nathan Kallus. 2020. From predictive to prescriptive analytics. *Management Science* 66, 3 (2020), 1025–1044.

[7] Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. 2017. Optimized pre-processing for discrimination prevention. In *Advances in Neural Information Processing Systems*. 3992–4001.

[8] Chris Cameron, Jason Hartford, Taylor Lundy, and Kevin Leyton-Brown. 2022. The perils of learning before optimizing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 3708–3715.

[9] Violet Xinying Chen and J. N. Hooker. 2023. A guide to formulating fairness in an optimization model. *Annals of Operations Research* 326, 1 (July 2023), 581–619. https://doi.org/10.1007/s10479-023-05264-

[10] Marco Chiarandini, Rolf Fagerberg, and Stefano Gualandi. 2019. Handling preferences in student-project allocation. *Annals of Operations Research* 275, 1 (2019), 39–78.

[11] Alex Chohlas-Wood, Madison Coots, Henry Zhu, Emma Brunskill, and Sharad Goel. 2024. Learning to Be Fair: A Consequentialist Approach to Equitable Decision Making. *Management Science* (2024).

[12] Sam Corbett-Davies, Johann D Gaebler, Hamed Nilforoshan, Ravi Shroff, and Sharad Goel. 2023. The measure and mismeasure of fairness. *The Journal of Machine Learning Research* 24, 1 (2023), 14730–14846.

[13] Hugh Dalton. 1920. The measurement of the inequality of incomes. *The economic journal* 30, 119 (1920), 348–361.

[14] My H Dinh, James Kotary, and Ferdinando Fioretto. 2024. Learning Fair Ranking Policies via Differentiable Optimization of Ordered Weighted Averages. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency*. 2508–2517.

[15] Michele Donini, Luca Oneto, Shai Ben-David, John S Shawe-Taylor, and Massimiliano Pontil. 2018. Empirical risk minimization under fairness constraints. In *Advances in Neural Information Processing Systems*. 2791–2801.

[16] Priya Donti, Brandon Amos, and J Zico Kolter. 2017. Task-based end-to-end model learning in stochastic optimization. *Advances in neural information processing systems* 30 (2017).

[17] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*. 214–226.

[18] O. Eisenhandler and M. Tzur. 2019. The humanitarian pickup and distribution problem. *Operations Research* 67 (2019), 10–32.

[19] Adam N. Elmachtoub and Paul Grigas. 2020. Smart "Predict, then Optimize". arXiv:1710.08005 [math.OC] https://arxiv.org/abs/1710.08005

[20] Adam N Elmachtoub, Henry Lam, Haofeng Zhang, and Yunfan Zhao. 2023. Estimate-Then-Optimize versus Integrated-Estimation-Optimization versus Sample Average Approximation: A Stochastic Dominance Perspective. *arXiv preprint arXiv:2304.06833* (2023).

[21] Jessie Finocchiaro, Roland Maio, Faidra Monachou, Gourab K Patro, Manish Raghavan, Ana-Andreea Stoica, and Stratis Tsirtsis. 2021. Bridging machine learning and mechanism design towards algorithmic fairness. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. 489–503.

[22] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*. 3315–3323.

[23] Michael Huang and Vishal Gupta. 2024. Decision-Focused Learning with Directional Gradients. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=g8kFlZDcaX

[24] Michael Huang and Vishal Gupta. 2024. Learning Best-in-Class Policies for the Predict-then-Optimize Framework. *arXiv preprint arXiv:2402.03256* (2024).

[25] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2016. Inherent trade-offs in the fair determination of risk scores. In *Proceedings of the 8th Innovations in Theoretical Computer Science Conference*. ACM, 43:1–43:23.

[26] James Kotary, Ferdinando Fioretto, Pascal Van Hentenryck, and Ziwei Zhu. 2022. End-to-end learning for fair ranking systems. In *Proceedings of the ACM Web Conference 2022*. 3520–3530.

[27] Tian Lan and Mung Chiang. 2011. An axiomatic theory of fairness in resource allocation. *George Washington University, http://www. seas. gwu. edu/tlan/papers/fairness. pdf, Tech. Rep* (2011).

[28] Lydia T Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. 2018. Delayed impact of fair machine learning. In *International Conference on Machine Learning*. PMLR, 3150–3158.

[29] Hanan Luss. 1999. On equitable resource allocation problems: A lexicographic minimax approach. *Operations Research* 47, 3 (1999), 361–378.

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

[31] Jeonghoon Mo and Jean Walrand. 2000. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on networking* 8, 5 (2000), 556–567.

[32] Mahdi Mostajabdaveh, Walter J Gutjahr, and Sibel Salman. 2019. Inequity-averse shelter location for disaster preparedness. *IISE Transactions* 51, 8 (2019), 809–829.

[33] Maxime Mulamba, Jayanta Mandi, Michelangelo Diligenti, Michele Lombardi, Victor Bucarey, and Tias Guns. 2020. Contrastive losses and solution caching for predict-and-optimize. *arXiv preprint arXiv:2011.05354* (2020).

[34] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. 2019. Dissecting racial bias in an algorithm used to manage the health of populations. *Science* 366, 6464 (2019), 447–453. https://doi.org/10.1126/science.aax2342 arXiv:https://www.science.org/doi/pdf/10.1126/science.aax2342

[35] Włodzimierz Ogryczak and Tomasz Śliwiński. 2002. On equitable approaches to resource allocation problems: The conditional minimax solutions. *Journal of Telecommunications and Information Technology* (2002), 40–48.

[36] Mahbod Olfat and Anil Aswani. 2018. Spectral algorithms for computing fair support vector machines. In *International Conference on Artificial Intelligence and Statistics*. 1933–1942.

[37] Marin Vlastelica Pogančić, Anselm Paulus, Vit Musil, Georg Martius, and Michal Rolinek. 2020. Differentiation of blackbox combinatorial solvers. In *International Conference on Learning Representations*.

[38] Meng Qi and Zuo-Jun Shen. 2022. Integrating prediction/estimation and optimization with applications in operations management. In *Tutorials in operations research: emerging and impactful topics in operations*. INFORMS, 36–58.

[39] John Rawls. 1971. *A Theory of Justice*. Harvard University Press.

[40] Subham Sekhar Sahoo, Anselm Paulus, Marin Vlastelica, Vít Musil, Volodymyr Kuleshov, and Georg Martius. 2022. Backpropagation through combinatorial algorithms: Identity with projection works. *arXiv preprint arXiv:2205.15213* (2022).

[41] Teresa Scantamburlo, Joachim Baumann, and Christoph Heitz. 2024. On prediction-modelers and decision-makers: why fairness requires more than a fair prediction model. *AI & SOCIETY* (2024), 1–17.

[42] Sanket Shah, Kai Wang, Bryan Wilder, Andrew Perrault, and Milind Tambe. 2022. Decision-focused learning without decision-making: Learning locally optimized decision losses. *Advances in Neural Information Processing Systems* 35 (2022), 1320–1332.

[43] Bo Tang and Elias B. Khalil. 2023. PyEPO: A PyTorch-based End-to-End Predict-then-Optimize Library for Linear and Integer Programming. arXiv:2206.14234 [math.OC] https://arxiv.org/abs/2206.14234

[44] Angelina Wang, Sayash Kapoor, Solon Barocas, and Arvind Narayanan. 2024. Against Predictive Optimization: On the Legitimacy of Decision-making Algorithms That Optimize Predictive Accuracy. *ACM J. Responsib. Comput.* 1, 1, Article 9 (3 2024), 45 pages. https://doi.org/10.1145/3636509

[45] Bryan Wilder, Bistra Dilkina, and Milind Tambe. 2019. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 1658–1665.

[46] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P Gummadi. 2019. Fairness constraints: A flexible approach for fair classification. *Journal of Machine Learning Research* 20, 75 (2019), 1–42.

[47] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *International Conference on Machine Learning*. 325–333.

[48] Arman Zharmagambetov, Brandon Amos, Aaron M Ferber, Taoan Huang, Bistra Dilkina, and Yuandong Tian. 2023. Landscape Surrogate: Learning Decision Losses for Mathematical Optimization Under Partial Information. In *Thirty-seventh Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=qyEm4tF2p1

A Statistics on Dataset

Table 4 and Table 5 presents some statistics of the variables used in our dataset. Recall that gain factors are generated using avoidable costs.

Table 4. Descriptive statistics of Dataset (mean)

	Count	Risk Score	Gain Factor	Avoidable Costs	Screening Eligible	Propensity Score
White	4432	4.31	58.82	2269.34	0.46	0.45
Black	568	5.45	77.36	3544.37	0.49	0.50

Table 5. Descriptive statistics of Dataset (std)

	Count	Risk Score	Gain Factor	Avoidable Costs	Screening Eligible	Propensity Score
White	4432	5.27	291.58	11638.21	0.50	0.37
Black	568	8.11	428.94	15655.29	0.50	0.38

B Results from Using Different Alpha Values and Models

This section presents additional experimental results examining how different choices of α in our α-fairness objective function and model architectures (linear regression and neural networks) affect the performance of our proposed decision-focused methods.

Table 6. Neural Network Results For $\alpha = 0.5$

Method	λ	MSE (Mean ± Std)	Regret (Mean ± Std)	Accuracy Parity (Mean ± Std)	Time/Epoch
2-Stage (MSE)	0	15.60 ± 2.65	0.08 ± 0.02	10.493 ± 6.137	0.36 ± 0.01
2-Stage (MSE)	1	20.34 ± 2.32	0.08 ± 0.03	2.435 ± 3.979	0.34 ± 0.02
FDFL (Closed-Form)	0	33.97 ± 4.15	0.09 ± 0.02	25.715 ± 12.674	1.35 ± 0.02
FDFL (Closed-Form)	1	33.94 ± 4.11	0.09 ± 0.02	25.627 ± 12.556	1.36 ± 0.02
FDFL (DBB)	0	50.79 ± 3.85	14.29 ± 10.98	40.282 ± 16.483	28.86 ± 1.32
FDFL (DBB)	1	50.58 ± 3.60	17.71 ± 11.02	40.305 ± 16.556	28.35 ± 1.35

Table 7. Linear Regression Results $\alpha = 0.5$

Method	λ	MSE (Mean ± Std)	Regret (Mean ± Std)	Accuracy Parity (Mean ± Std)	Time/Epoch
2-Stage	0	33.56 ± 3.37	0.13 ± 0.03	21.880 ± 11.837	0.39 ± 0.02
2-Stage	1	34.44 ± 3.86	0.11 ± 0.03	19.129 ± 11.301	0.38 ± 0.01
FDFL (Closed-Form)	0	47.03 ± 3.63	0.14 ± 0.04	41.287 ± 16.423	1.35 ± 0.02
FDFL (Closed-Form)	1	47.13 ± 3.74	0.14 ± 0.04	41.408 ± 16.376	1.35 ± 0.02
FDFL (DBB)	0	45.96 ± 3.64	0.11 ± 0.04	39.625 ± 15.860	42.04 ± 1.82
FDFL (DBB)	1	46.27 ± 3.52	0.11 ± 0.04	39.617 ± 15.971	45.35 ± 1.66

Table 8. Neural Network Results $\alpha = 1.5$

Method	λ	MSE (Mean ± Std)	Regret (Mean ± Std)	Accuracy Parity (Mean ± Std)	Time/Epoch
2-Stage	0	16.34 ± 1.78	0.41 ± 0.05	10.295 ± 6.413	0.38 ± 0.01
2-Stage	1	19.98 ± 2.42	0.41 ± 0.04	0.894 ± 0.768	0.39 ± 0.02
FDFL (Closed-Form)	0	44.95 ± 21.81	0.22 ± 0.05	38.849 ± 21.638	1.37 ± 0.03
FDFL (Closed-Form)	1	58.83 ± 27.64	0.24 ± 0.07	50.118 ± 28.338	1.37 ± 0.03
FDFL (DBB)	0	50.50 ± 3.91	0.49 ± 0.10	40.050 ± 16.674	34.12 ± 0.76
FDFL (DBB)	1	50.05 ± 3.97	0.45 ± 0.08	39.530 ± 16.621	34.23 ± 1.70

Table 9. Linear Regression Results $\alpha = 1.5$

Method	λ	MSE (Mean ± Std)	Regret (Mean ± Std)	Accuracy Parity (Mean ± Std)	Time/Epoch
2-Stage	0	28.20 ± 2.80	0.28 ± 0.04	16.042 ± 10.030	0.49 ± 0.01
2-Stage	1	30.30 ± 3.15	0.31 ± 0.05	13.347 ± 9.142	0.49 ± 0.01
FDFL (Closed-Form)	0	33.01 ± 5.18	0.29 ± 0.06	22.527 ± 12.773	1.41 ± 0.02
FDFL (Closed-Form)	1	36.12 ± 4.99	0.31 ± 0.07	26.106 ± 14.362	1.41 ± 0.02
FDFL (DBB)	0	43.34 ± 6.13	0.52 ± 0.14	36.789 ± 19.480	39.83 ± 0.81
FDFL (DBB)	1	42.52 ± 4.95	0.47 ± 0.12	35.591 ± 15.563	36.58 ± 0.89

Table 10. Linear Regression Results For $\alpha = 2$

Method	λ	MSE (Mean ± Std)	Regret (Mean ± Std)	Accuracy Parity (Mean ± Std)	Time/Epoch
2-Stage	0	28.42 ± 2.79	4.46 ± 0.59	16.666 ± 10.722	0.53 ± 0.02
2-Stage	1	30.55 ± 3.25	6.74 ± 0.93	12.907 ± 9.190	0.52 ± 0.01
FDFL (Closed-Form)	0	31.33 ± 3.82	4.02 ± 1.71	19.668 ± 10.714	1.37 ± 0.06
FDFL (Closed-Form)	1	33.36 ± 5.20	5.01 ± 2.29	25.320 ± 15.531	1.36 ± 0.05
FDFL (DBB)	0	42.06 ± 6.39	17.79 ± 9.05	33.164 ± 16.777	32.19 ± 2.57
FDFL (DBB)	1	40.28 ± 5.62	13.70 ± 8.79	30.514 ± 16.503	37.63 ± 0.62

Table 11. Neural Network Results For $\alpha = \infty$

Method	λ	MSE (Mean ± Std)	Regret (Mean ± Std)	Fairness (Mean ± Std)	Time/Epoch
2-Stage	0	16.06 ± 2.30	0.41 ± 0.04	10.179 ± 5.872	0.33 ± 0.02
2-Stage	1	19.71 ± 2.77	0.40 ± 0.03	1.789 ± 2.200	0.33 ± 0.03
FDFL (Closed-Form)	0	27.57 ± 3.54	0.29 ± 0.04	19.206 ± 7.407	1.27 ± 0.07
FDFL (Closed-Form)	1	23.98 ± 4.93	0.34 ± 0.07	17.163 ± 7.234	1.29 ± 0.04
FDFL (DBB)	0	49.72 ± 4.28	0.51 ± 0.12	39.517 ± 16.097	34.63 ± 0.47
FDFL (DBB)	1	50.30 ± 3.95	0.46 ± 0.06	39.928 ± 16.342	34.45 ± 0.65

C Alternative Methods in Fair Decision-Focused Learning

This section discusses additional methodological approaches for implementing fair decision-focused learning, building upon the gradient computation framework established in Equation (7), The SPO+ method [19] addresses the challenge of optimizing decision regret by introducing a convex surrogate loss that upper bounds the true regret. When the decision objective is linear, $W(\mathbf{d}; \mathbf{r}) = \mathbf{d}^T \mathbf{r}$, the SPO+ loss is defined as:

Table 12. Linear Regression Results For $\alpha = \infty$

Method	λ	MSE (Mean ± Std)	Regret (Mean ± Std)	Fairness (Mean ± Std)	Time/Epoch
2-Stage	0	28.38 ± 3.12	0.28 ± 0.01	16.862 ± 10.976	0.31 ± 0.02
2-Stage	1	30.33 ± 3.42	0.31 ± 0.04	12.996 ± 9.052	0.34 ± 0.03
FDFL (Closed-Form)	0	36.54 ± 5.24	0.33 ± 0.03	26.431 ± 12.845	1.42 ± 0.09
FDFL (Closed-Form)	1	34.78 ± 7.86	0.33 ± 0.10	25.798 ± 15.023	1.29 ± 0.09
FDFL (DBB)	0	42.91 ± 5.62	0.41 ± 0.10	35.555 ± 17.002	28.29 ± 3.62
FDFL (DBB)	1	43.06 ± 6.11	0.47 ± 0.12	37.481 ± 19.580	26.15 ± 0.80

$$L_{\text{SPO+}}(\hat{\mathbf{r}}, \mathbf{r}) = \max_{\mathbf{d} \in \mathcal{S}} \{(2\hat{\mathbf{r}} - \mathbf{r})^T \mathbf{d}\} + 2\hat{\mathbf{r}}^T \mathbf{d}^*(\mathbf{r}) - \mathbf{d}^*(\mathbf{r}), \tag{10}$$

The subgradient of this loss with respect to $\hat{\mathbf{r}}$ can be derived as:

$$\nabla_{\hat{\mathbf{r}}} L_{\text{SPO+}} \in 2(\mathbf{d}^*(\mathbf{r}) - \mathbf{d}^*(2\hat{\mathbf{r}} - \mathbf{r})). \tag{11}$$

This subgradient enables the use of standard stochastic gradient descent methods for training the predictor within the DFL framework.

Perturbation Gradient Loss (PG) [23] leverages Danskin's Theorem to derive informative gradients through zeroth-order approximations. The PG loss approximates the gradient of the decision objective by perturbing the predicted parameters and observing the resulting changes in the optimal decision:

$$\nabla_{\hat{\mathbf{r}}} L_{\text{PG}}(\hat{\mathbf{r}}, \mathbf{r}) \approx \frac{1}{\sigma}(\mathbf{d}^*(\hat{\mathbf{r}} + \sigma \mathbf{u}) - \mathbf{d}^*(\hat{\mathbf{r}})), \tag{12}$$

where $\sigma > 0$ controls the perturbation magnitude and $\mathbf{u}$ is a randomly sampled direction vector. This approximation provides a non-zero gradient that guides the predictor towards decisions that minimize regret.

The Differentiable Perturbed Optimizer (DPO) and Perturbed Fenchel-Young Loss (PYFL) [5] utilize Monte Carlo sampling with Gaussian noise to estimate the gradients. By introducing multiple perturbations to the predicted parameters, these methods approximate the expected optimal decisions:

$$\mathbb{E}_{\xi}[\mathbf{d}^*(\hat{\mathbf{r}} + \sigma\xi)] \approx \frac{1}{K} \sum_{k=1}^{K} \mathbf{d}^*(\hat{\mathbf{r}} + \sigma\xi_k), \tag{13}$$

where $\xi_k \sim \mathcal{N}(0, I)$ are independent Gaussian noise samples. This averaging process smoothens the gradient estimates, providing a reliable signal for training the predictor within the DFL framework.

Negative Identity Backpropagation (NID) [40] presents a hyperparameter-free alternative to DBB by treating the optimizer as a negative identity mapping during the backward pass. Instead of introducing an explicit perturbation parameter, NID assumes an identity relationship between the gradients, effectively bypassing the need for additional solver calls during backpropagation. Mathematically, the gradient is

$$\frac{\partial \mathbf{d}^*(\hat{\mathbf{r}})}{\partial \hat{\mathbf{r}}} = -I, \tag{14}$$

where I is the identity matrix. This simplification reduces computational overhead while maintaining gradient informativeness.

Noise Contrastive Estimation (NCE) [33] and its variant, Contrastive Maximum A Posteriori Estimation (CMAP), adopt a contrastive learning approach by treating suboptimal solutions as negative examples. The NCE loss function is designed to maximize the separation between the optimal decision and sampled suboptimal decisions:

$$L_{\text{NCE}} = -\log \frac{\exp(\mathbf{d}^*(\hat{\mathbf{r}})^T \mathbf{d}^*(\mathbf{r}))}{\exp(\mathbf{d}^*(\hat{\mathbf{r}})^T \mathbf{d}^*(\mathbf{r})) + \sum_{k=1}^{K} \exp(\mathbf{d}^*(\hat{\mathbf{r}})^T \mathbf{d}_k)}, \tag{15}$$

where $\mathbf{d}_k$ are sampled suboptimal solutions. CMAP simplifies this by considering only the best negative sample, thereby reducing computational complexity while retaining the essence of contrastive learning.

D Closed-form Solution Special Alphas

Here we provide some extra results of the closed form solution of the α-fairness maximization problem from (9), and provide omitted proofs. In addition to the general closed-form solution for $\alpha \neq 1$, the solutions and gradients for specific values of $\alpha = 0$, $\alpha = 1$, and $\alpha = \infty$ can be expressed as follows:

$\alpha = 0$ *(Utilitarian Fairness).* The optimal solution d_i^* is given by allocating the entire budget to the decision variable with the highest utility-to-cost ratio:

$$d_i^* = \begin{cases} \frac{Q}{c_i} & \text{if } i = \arg\max_j \frac{r_j g_j}{c_j} \\ 0 & \text{otherwise.} \end{cases}$$

$\alpha = 1$ *(Proportional Fairness).* The optimal solution d_i^* is given by:

$$d_i^* = \frac{Q}{nc_i}.$$

$\alpha = \infty$ *(Max-Min Fairness).* The optimal solution d_i^* is determined by equalizing the utilities per unit cost:

$$d_i^* = \frac{Qc_i}{(r_i g_i) \sum_{j=1}^{n} \frac{c_j^2}{r_j g_j}}.$$

D.1 Omitted Proofs of Closed Form Solution and Gradient

Below we sketch proof for the closed form solution and analytical form of gradients for the alpha fairness problem from (9).

Proof. To ensure consistency with standard optimization conventions, we convert this into a minimization problem:

$$\begin{aligned} \min_{\mathbf{d}} \quad & -\frac{1}{1-\alpha} \sum_{i=1}^{n} (u_i)^{1-\alpha} \\ \text{s.t.} \quad & d_i \geq 0, \quad \forall i = 1, \ldots, n, \\ & \sum_{i=1}^{n} c_i d_i \leq Q. \end{aligned}$$

We begin by formulating the Lagrangian for the minimization problem:

$$\mathcal{L}(\mathbf{d}, \lambda, \mu) = -\frac{1}{1-\alpha} \sum_{i=1}^{n} (u_i)^{1-\alpha} + \lambda \left(\sum_{i=1}^{n} c_i d_i - Q \right) - \sum_{i=1}^{n} \mu_i d_i,$$

where $\lambda \geq 0$ is the Lagrange multiplier for the inequality constraint $\sum_{i=1}^n c_i d_i \leq Q$, and $\mu_i \geq 0$ are the Lagrange multipliers for the non-negativity constraints $d_i \geq 0$.

Substituting $u_i = r_i g_i d_i$, the Lagrangian becomes:

$$\mathcal{L}(\mathbf{d}, \lambda, \mu) = -\frac{1}{1-\alpha} \sum_{i=1}^n (r_i g_i d_i)^{1-\alpha} + \lambda \left(\sum_{i=1}^n c_i d_i - Q \right) - \sum_{i=1}^n \mu_i d_i.$$

Assuming interior solutions where $d_i > 0$ and thus $\mu_i = 0$ (complementary slackness), we take the derivative of $\mathcal{L}$ with respect to d_i and set it to zero:

$$\frac{\partial \mathcal{L}}{\partial d_i} = -(r_i g_i)^{1-\alpha} d_i^{-\alpha} + \lambda c_i = 0.$$

Solving for d_i, we obtain:

$$d_i = \left(\frac{(r_i g_i)^{1-\alpha}}{\lambda c_i} \right)^{\frac{1}{\alpha}}.$$

Simplifying, we have:

$$d_i = \frac{(r_i g_i)^{\frac{1}{\alpha}-1}}{c_i^{\frac{1}{\alpha}} \lambda^{\frac{1}{\alpha}}}.$$

To determine λ, we use the constraint $\sum_{i=1}^n c_i d_i = Q$:

$$\sum_{i=1}^n c_i d_i = \sum_{i=1}^n c_i \cdot \frac{(r_i g_i)^{\frac{1}{\alpha}-1}}{c_i^{\frac{1}{\alpha}} \lambda^{\frac{1}{\alpha}}} = \frac{1}{\lambda^{\frac{1}{\alpha}}} \sum_{i=1}^n c_i^{1-\frac{1}{\alpha}} (r_i g_i)^{\frac{1}{\alpha}-1} = Q.$$

Solving for λ:

$$\lambda^{\frac{1}{\alpha}} = \frac{1}{Q} \sum_{i=1}^n c_i^{1-\frac{1}{\alpha}} (r_i g_i)^{\frac{1}{\alpha}-1},$$

$$\lambda = \left(\frac{1}{Q} \sum_{i=1}^n c_i^{1-\frac{1}{\alpha}} (r_i g_i)^{\frac{1}{\alpha}-1} \right)^{\alpha}.$$

Substituting λ back into the expression for d_i, we obtain:

$$d_i^* = \frac{(r_i g_i)^{\frac{1}{\alpha}-1}}{c_i^{\frac{1}{\alpha}}} \cdot \frac{Q}{\sum_{j=1}^n c_j^{-\frac{1}{\alpha}} (r_j g_j)^{\frac{1}{\alpha}-1}}.$$

Noting that g_i is a constant parameter and can be absorbed into r_i for simplification (assuming g_i are known and fixed), we redefine r_i to include g_i. Thus, the closed-form solution simplifies to:

$$d_i^* = \frac{c_i^{-\frac{1}{\alpha}} r_i^{\frac{1}{\alpha}-1} Q}{\sum_{j=1}^n c_j^{-\frac{1}{\alpha}} r_j^{\frac{1}{\alpha}-1}}.$$

Case $\alpha = 0$ (Utilitarian). When $\alpha = 0$, the objective function simplifies to maximizing the total utility:

$$W_0(\mathbf{u}) = \sum_{i=1}^{n} r_i g_i d_i.$$

This is a linear objective function subject to the budget constraint $\sum_{i=1}^{n} c_i d_i \leq Q$. To maximize W_0, allocate all resources to the decision variable with the highest utility-to-cost ratio $\frac{r_i g_i}{c_i}$. Define:

$$i^* = \arg\max_j \left(\frac{r_j g_j}{c_j}\right).$$

Thus, the optimal allocation is:

$$d_{i^*}^* = \frac{Q}{c_{i^*}}, \quad d_i^* = 0 \ \forall i \neq i^*.$$

This allocation satisfies the budget constraint:

$$\sum_{i=1}^{n} c_i d_i^* = c_{i^*} \cdot \frac{Q}{c_{i^*}} = Q.$$

Case $\alpha = 1$ (Proportional). When $\alpha = 1$, the objective function corresponds to maximizing the proportional fairness:

$$W_1(\mathbf{u}) = \sum_{i=1}^{n} \log(u_i) = \sum_{i=1}^{n} \log(r_i g_i d_i).$$

Formulate the Lagrangian:

$$\mathcal{L}(\mathbf{d}, \lambda) = \sum_{i=1}^{n} \log(r_i g_i d_i) + \lambda \left(Q - \sum_{i=1}^{n} c_i d_i\right).$$

Taking the derivative with respect to d_i and setting to zero:

$$\frac{\partial \mathcal{L}}{\partial d_i} = \frac{1}{d_i} - \lambda c_i = 0 \implies d_i = \frac{1}{\lambda c_i}.$$

Applying the budget constraint:

$$\sum_{i=1}^{n} c_i d_i = \sum_{i=1}^{n} c_i \cdot \frac{1}{\lambda c_i} = \frac{n}{\lambda} = Q \implies \lambda = \frac{n}{Q}.$$

Thus, the optimal allocation is:

$$d_i^* = \frac{1}{\lambda c_i} = \frac{Q}{n c_i}.$$

However, to generalize based on individual utilities, the allocation adjusts to:

$$d_i^* = \frac{Q}{\sum_{j=1}^{n} \frac{c_j}{r_j g_j}} \cdot \frac{1}{c_i}.$$

Case $\alpha = \infty$ (Maximin). For $\alpha = \infty$, the objective focuses on maximizing the minimum utility:

$$W_\infty(\mathbf{u}) = \min_i \{r_i g_i d_i\}.$$

To maximize the minimum utility, allocate resources such that all utilities are equal:

$$r_i g_i d_i = C \quad \forall i,$$

where C is a constant. Solving for d_i:

$$d_i = \frac{C}{r_i g_i}.$$

Substituting into the budget constraint:

$$\sum_{i=1}^{n} c_i d_i = \sum_{i=1}^{n} c_i \cdot \frac{C}{r_i g_i} = C \sum_{i=1}^{n} \frac{c_i}{r_i g_i} = Q \implies C = \frac{Q}{\sum_{j=1}^{n} \frac{c_j}{r_j g_j}}.$$

Thus, the optimal allocation is:

$$d_i^* = \frac{C}{r_i g_i} = \frac{Q c_i}{r_i g_i \sum_{j=1}^{n} \frac{c_j}{r_j g_j}}.$$

Multiplying numerator and denominator by c_i for consistency:

$$d_i^* = \frac{Q c_i^2}{r_i g_i \sum_{j=1}^{n} \frac{c_j^2}{r_j g_j}}.$$

Simplifying, we obtain:

$$d_i^* = \frac{Q c_i}{r_i g_i \sum_{j=1}^{n} \frac{c_j^2}{r_j g_j}}.$$

This concludes the proof of Proposition 4.1. □

Proof. Given the closed-form solution:

$$d_i^* = \frac{c_i^{-\frac{1}{\alpha}} r_i^{\frac{1}{\alpha}-1} Q}{S},$$

where $S = \sum_{j=1}^{n} c_j^{-\frac{1}{\alpha}} r_j^{\frac{1}{\alpha}-1}$.

To find $\frac{\partial d_i^*}{\partial r_i}$, we apply the quotient rule:

$$\frac{\partial d_i^*}{\partial r_i} = \frac{Q \cdot \frac{\partial}{\partial r_i}\left(c_i^{-\frac{1}{\alpha}} r_i^{\frac{1}{\alpha}-1}\right) \cdot S - Q \cdot c_i^{-\frac{1}{\alpha}} r_i^{\frac{1}{\alpha}-1} \cdot \frac{\partial S}{\partial r_i}}{S^2}.$$

Compute the derivatives:

$$\frac{\partial}{\partial r_i}\left(c_i^{-\frac{1}{\alpha}} r_i^{\frac{1}{\alpha}-1}\right) = c_i^{-\frac{1}{\alpha}} \left(\frac{1}{\alpha} - 1\right) r_i^{-2+\frac{1}{\alpha}},$$
$$\frac{\partial S}{\partial r_i} = c_i^{-\frac{1}{\alpha}} \left(\frac{1}{\alpha} - 1\right) r_i^{-2+\frac{1}{\alpha}}.$$

Substituting back into the expression for $\frac{\partial d_i^*}{\partial r_i}$:

$$\frac{\partial d_i^*}{\partial r_i} = \frac{Q \cdot c_i^{-\frac{1}{\alpha}} \left(\frac{1}{\alpha} - 1\right) r_i^{-2+\frac{1}{\alpha}} \cdot S - Q \cdot c_i^{-\frac{1}{\alpha}} r_i^{\frac{1}{\alpha}-1} \cdot c_i^{-\frac{1}{\alpha}} \left(\frac{1}{\alpha} - 1\right) r_i^{-2+\frac{1}{\alpha}}}{S^2}.$$

Factor out common terms:

$$\frac{\partial d_i^*}{\partial r_i} = \frac{Q \cdot c_i^{-\frac{1}{\alpha}} \left(\frac{1}{\alpha} - 1\right) r_i^{-2+\frac{1}{\alpha}} \left(S - c_i^{1-\frac{1}{\alpha}} r_i^{-1+\frac{1}{\alpha}}\right)}{S^2}.$$

Simplifying, we obtain:

$$\frac{\partial d_i^*}{\partial r_i} = \frac{Q\left(\left(-1+\frac{1}{\alpha}\right) c_i^{-\frac{1}{\alpha}} r_i^{-2+\frac{1}{\alpha}} \left(S - c_i^{1-\frac{1}{\alpha}} r_i^{-1+\frac{1}{\alpha}}\right)\right)}{S^2}.$$

We derive the optimal allocations and their gradients for each case using the method of Lagrangian multipliers.

Case $\alpha = 0$ (Utilitarian). The optimization problem simplifies to:

$$\min_{\mathbf{d}} \quad -\sum_{i=1}^{n} r_i g_i d_i \quad \text{s.t.} \quad \sum_{i=1}^{n} c_i d_i \leq Q, \quad d_i \geq 0.$$

Since the objective is linear, the optimal solution allocates the entire budget to the decision variable with the highest utility-to-cost ratio:

$$i^* = \arg\max_j \left(\frac{r_j g_j}{c_j}\right).$$

Thus,

$$d_{i^*}^* = \frac{Q}{c_{i^*}}, \quad d_i^* = 0 \quad \forall i \neq i^*.$$

Since d_i^* does not depend on r_k, the gradient is zero:

$$\frac{\partial d_i^*}{\partial r_k} = 0, \quad \forall i, k.$$

Case $\alpha = 1$ (Proportional). For $\alpha = 1$, the optimization problem becomes:

$$\min_{\mathbf{d}} \quad -\sum_{i=1}^{n} \ln(r_i g_i d_i) \quad \text{s.t.} \quad \sum_{i=1}^{n} c_i d_i \leq Q, \quad d_i \geq 0.$$

The Lagrangian is:

$$\mathcal{L} = -\sum_{i=1}^{n} \ln(r_i g_i d_i) + \lambda \left(\sum_{i=1}^{n} c_i d_i - Q\right).$$

Taking the derivative with respect to d_i and setting it to zero:

$$\frac{\partial \mathcal{L}}{\partial d_i} = -\frac{1}{d_i} + \lambda c_i = 0 \implies d_i^* = \frac{1}{\lambda c_i}.$$

Applying the budget constraint:

$$\sum_{i=1}^{n} c_i d_i^* = \sum_{i=1}^{n} c_i \cdot \frac{1}{\lambda c_i} = \frac{n}{\lambda} = Q \implies \lambda = \frac{n}{Q}.$$

Thus, the optimal allocation is:

$$d_i^* = \frac{Q}{n c_i}, \quad \forall i.$$

Since d_i^* is independent of r_k, the gradient is zero:

$$\frac{\partial d_i^*}{\partial r_k} = 0, \quad \forall i, k.$$

Case $\alpha = \infty$ (Maximin). For $\alpha = \infty$, the optimization problem emphasizes maximin fairness:

$$\min_{\mathbf{d}} \quad \min_i \{-\ln(r_i g_i d_i)\} \quad \text{s.t.} \quad \sum_{i=1}^{n} c_i d_i \leq Q, \quad d_i \geq 0.$$

This can be reformulated as:

$$\max_{\mathbf{d}} \quad \min_i \{r_i g_i d_i\} \quad \text{s.t.} \quad \sum_{i=1}^{n} c_i d_i \leq Q, \quad d_i \geq 0.$$

To maximize the minimum utility, set all utilities equal:

$$r_i g_i d_i = C \quad \forall i,$$

for some constant $C > 0$. Substituting into the budget constraint:

$$\sum_{i=1}^{n} c_i d_i = \sum_{i=1}^{n} c_i \cdot \frac{C}{r_i g_i} = Q \implies C = \frac{Q}{\sum_{j=1}^{n} \frac{c_j^2}{r_j g_j}} = \frac{Q}{S},$$

where $S = \sum_{j=1}^{n} \frac{c_j^2}{r_j g_j}$. Thus, the optimal allocation is:

$$d_i^* = \frac{C}{r_i g_i} = \frac{Q c_i}{r_i g_i S}.$$

To compute the gradient $\frac{\partial d_i^*}{\partial r_k}$, we differentiate d_i^* with respect to r_k:

$$d_i^* = \frac{Q c_i}{r_i g_i S}, \quad \text{where} \quad S = \sum_{j=1}^{n} \frac{c_j^2}{r_j g_j}.$$

Differentiating:

$$\frac{\partial d_i^*}{\partial r_k} = \begin{cases} -\frac{d_i^*}{r_i} - \frac{d_i^* c_i}{r_i g_i S}, & \text{if } k = i, \\ \frac{d_i^* c_k^2}{c_i r_k^2 g_k S}, & \text{if } k \neq i. \end{cases}$$

This completes the proof of Proposition 4.2. □