Querywise Fair Learning to Rank through Multi-Objective Optimization

Debabrata Mahapatra
debabrata@u.nus.edu
National University of Singapore
Singapore

Chaosheng Dong
chaosd@amazon.com
Amazon
Seattle, WA, USA

Michinari Momma
michi@amazon.com
Amazon
Seattle, WA, USA

ABSTRACT

In Learning-to-Rank (LTR) problems, the task of delivering relevant search results and allocating fair exposure to items of a protected group can conflict. Previous works in Fair LTR have attempted to resolve this by combining the objectives of relevant ranking and fair ranking into a single linear combination, but this approach is limited by the nonconvexity of the objective functions and can result in suboptimal relevance in ranking outputs. To address this, we propose a solution using Multi-Objective Optimization (MOO) algorithms. We extend these algorithms to querywise MOO to reduce the exposure disparity, not only on average but also at the query level. Interestingly, for moderate fairness requirements, it improves the relevance of ranking instead of deteriorating. We attribute this improvement to the benefits of multi-task learning and study the effect of fair ranking on the relevant ranking task. Moreover, we significantly improve the computational efficiency compared to previous methods by using the Gumbel max trick to sample the Plackett-Luce distribution. We evaluate our proposed methods on three real-world datasets and show their improvement in relevance ranking over state-of-the-art solutions.

CCS CONCEPTS

• **Computing methodologies → Multi-task learning**; • **Information systems → Learning to rank**; • **Mathematics of computing → Nonconvex optimization**.

KEYWORDS

Learning to Rank, Fair Ranking, Multi-Objective Optimization

ACM Reference Format:
Debabrata Mahapatra, Chaosheng Dong, and Michinari Momma. 2023. Querywise Fair Learning to Rank through Multi-Objective Optimization. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3580305.3599482

1 INTRODUCTION

Search ranking systems have become a fundamental aspect of modern society. Users and producers from virtually every economic strata use these systems to search and distribute information, media, food, products, services, employment, and even social connections. The success of the entities being searched is largely determined by their placement in the ranking, which highlights the significant economic and social influence that ranking algorithms hold.

The *Learning to Rank* (LTR) algorithms are trained on labels, such as purchase decisions, dwell time, click data etc., to assign a relevance score and rank the candidates/items. However, candidates from minority groups are more sensitive to relevance scores than those from dominant groups, e.g., in job search [13] and web search [37]. Additionally, the training data used for relevance scores is susceptible to position bias, which exacerbates the disparity in exposure and makes popular items even more popular.

To address this issue, the research in fair ranking has burgeoned recently, aiming for proper representations of different groups by providing them sufficient presence across all the ranking positions [5]. The fairness in exposure may be imposed itemwise, called as individual fairness, or groupwise, called as group fairness. In this research, we focus on group fairness. In fair LTR [37, 47, 44, 25, 18], in addition to ranking relevance, an additional objective is introduced to allocate fair exposure for the protected groups (based on gender, race, income level, etc.), who are underrepresented or consistently relegated to lower ranking positions.

Previous works in fair LTR have employed the *Linear Scalarization* (LS) approach to trade-off between the objectives of relevant ranking and fair ranking. However, it doesn't guarantee finding a model that produces the most relevant items while satisfying a specified level of fairness due to the nonconvex nature of the objective functions. Fairness is more important in some applications, such as job search, while relevance takes priority in others, such as e-commerce product search portals. Suboptimal ranking relevance could discourage platform providers, such as Google, Amazon, and LinkedIn, from adopting fair LTR techniques, as the relevance of search results is crucial for their business.

In this research, we address the issue of suboptimal ranking relevance in fair LTR by utilizing the principles of Multi-Objective Optimization (MOO). Specifically, we employ the ***W**eighted **C**hebyshev* (WC) Scalarization [41] and its recent advancements [27, 24, 23] to achieve the best ranking models for high levels of restriction on group exposure disparity. We aim to encourage the adoption of fair LTR methods by demonstrating that moderate usage of fairness tasks can enhance, not undermine, relevance in ranking. This is possible due to the inductive bias effect in Multi-Task Learning (MTL) [4, 33]. To achieve this, we propose the querywise WC method that minimize the exposure disparity at a query level. We extend it to advanced variants of WC, such as EPO Search [23, 24] and WC-MGDA [27], and evaluate its impact on relevance ranking tasks.

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08.
https://doi.org/10.1145/3580305.3599482

We implement the above mentioned methods efficiently to be applicable for large-scale real-world datasets. Our solution uses the Gumbel max trick [40] to sample permutations efficiently from the Plackett-Luce (PL) ranking policy [31, 21]. In fair LTR [37, 47, 44], objective functions are defined from the PL ranking policy [31, 21] and approximated by drawing permutation samples from it. However, previous methods, which used sampling without replacement, are not scalable to large datasets with many items per query. To further accelerate the training, we analytically derive the double derivatives of the PL log probabilities, which is significantly faster than using automatic differentiation of PyTorch [30].

In summary, we make the following contributions:

- We identify limitations of LS for combining relevance and fairness in ranking, and propose to use WC scalarization and its modern variants with theoretical justification.
- We introduce the Querywise Weighted Chebyshev (QWC) method and extend it to the modern variants of WC. We analyze its regularization and inductive bias effects that improve the relevance in ranking instead of deteriorating it for moderate fairness constraints.
- To accelerate the training of fair LTR models, we propose efficient sampling from PL distribution with the Gumbel max trick and faster double derivative computation of objectives.
- We demonstrate the superiority of the non-querywise MOO (in restrictive fairness conditions) and querywise MOO (in moderate fairness conditions) over LS on three real world datasets, and the scalability of our implementation on simulated datasets.

2 RELATED WORK

2.1 Fair Learning to Rank

Excellent reviews on the broad interdisciplinary subject of fairness in ranking (from social, philosophical and algorithmic aspects) can be found in [48], and on the specific topic of Fair LTR can be found in [49]. There are three approaches for Fair LTR: Pre-processing [38], where the input data to the ranking system is transformed; Post-processing [1, 6, 35, 45, 46, 36], where first an LTR model predicts the relevance scores and then the list is re-ranked based on a fair ranking policy; and In-processing [47, 37, 44, 18, 43], where an LTR model is trained to maximize ranking utility and minimize unfairness. Our work focuses on the in-processing approach.

The in-processing literature can be divided based on the ranking policy into two types: PL model [31, 21] based methods [37, 47, 44], and Doubly Stochastic Matrix (DSM) [25, 18] based methods.

In PL model based fair LTR, [37] applies policy gradients using the REINFORCE algorithm [43] to train the model. [44] extends this by using unbiased estimates of relevance labels and fairness loss in model training. [47] adopts the LTR approach of [3] considering the group discrepancy only in the top ranking position.

In DSM based fair LTR, [25] uses a game theoretic approach, where one player maximizes the ranking utility with fairness constraint and the other minimize. [18] uses a post-ranking strategy to achieve fairness. They highlight an important issue in the definition of fairness loss used in earlier methods that although improve the overall fairness, but overlook querywise fairness.

Despite the formulation of convex objectives in DSM-based methods, their reliance on inference-time re-ranking limits their practicality for large-scale search platforms. The linear programming problem to be solved at inference, with a size of $O(n^2)$, where n is the number of items to be ranked, can result in longer latency that negatively impacts revenue for search platform providers. This could deter the adoption of fair LTR, especially in scenarios where low latency is a critical factor. To address this, we adopt PL-based fair LTR algorithms that do not require inference-time optimization, enabling improved ranking relevance while preserving the benefits of low latency.

2.2 Multi-task Learning by Multi-Objective Optimization

LS was predominantly used in MTL literature[33]. To show MOO can improve MTL, [34] used the ***Multi-Gradient Descent Algorithm*** (MGDA) [9]. However, this method did not have any control over the trade-offs among the objectives. [19] introduced a preference based MOO algorithm to gain more control over the trade-off solutions. More recently, ***Exact Pareto Optimal*** (EPO) Search [24, 23] and WC-MGDA [27] extend the WC scalarization [41] to gain precise control over the multi-task trade-offs. Moreover, [22] systematically studied these multi-task learning by MOO algorithms for LTR. In this research, we extend the WC method and its modern variants by applying them separately on every query to improve the ranking relevance.

3 BACKGROUND

3.1 Fair Learning to Rank

We summarize the main notations in Table 1.

Dataset Description: The training dataset $\mathcal{D} = \left\{(\mathbf{x}_q, \mathbf{y}_q)\right\}_{q=1}^{N}$ contains information about N queries, where $\mathbf{x}_q \in \mathbb{R}^{n_q \times d}$ represents the d dimensional features for n_q items and $\mathbf{y}_q \in \mathbb{Y}^{n_q}$ is their relevance labels for the q^{th} query. Set $\mathbb{Y}$ is the range of relevance labels, e.g., binary $\{0, 1\}$, or ordinal ratings $\{0, 1, 2, 3, 4, 5\}$. The number of items n_q of query q is also called as its *Slate length.* We drop the subscript q and simply use the notations $\mathbf{x}$ and $\mathbf{y}$ for n items when the discussion is about a single query.

Metric for Relevance of a Ranking: Given n items, a ranking $\sigma : [n] \to [n]$ assigns the rank $\sigma(i)$ to item i. To determine the quality of this ranking σ w.r.t. the relevance labels $\mathbf{y}$, we use the *Normalized Discounted Cumulative Gain* (NDCG) [8] metric. Assuming the gain of item i to be its relevance label y_i, the relevance ranking metric is defined as

$$M(\mathbf{y}, \sigma) = \frac{\mathbf{y}_\sigma^T \boldsymbol{\Delta}}{\text{sort}(\mathbf{y})^T \boldsymbol{\Delta}}, \tag{1}$$

where $\mathbf{y}_\sigma$ is the permutation of $\mathbf{y}$ according to the ranking σ, $\text{sort}(\mathbf{y})$ is a sorting of the relevance labels in descending order, and $\boldsymbol{\Delta} \in \mathbb{R}^n$ is the discount vector that reduces the gain of higher ranked (less relevant) items, e.g., $\Delta_k = \frac{1}{\log(1+k)}$ is the discount at rank k.

Group Exposure Discrepancy for Unfairness of a Ranking: The exposure of an item at rank k is given by v_k that decreases with increasing rank, e.g., $v_k = \frac{1}{log(k+1)}$ or $\frac{1}{k^p}$ for $p \geq 1$. The exposure of a group $G \subset [n]$ of items enjoyed by ranking σ is denoted by $v_G(\sigma) = \sum_{i \in G} v_{\sigma(i)}$, and the discrepancy in exposure between groups G and G' is given by

$$D(\sigma) = \frac{v_G(\sigma)}{|G|} - \frac{v_{G'}(\sigma)}{|G'|}. \tag{2}$$

Another form of group discrepancies used in the literature [37, 44, 18] is only the numerator of (2), i.e., $v_G(\sigma)|G'| - v_{G'}|G|$. However, this form of group discrepancy assumes the slate lengths n_q of all the queries are the same, so that the scale of this metric remains same while averaging over the queries. Unfortunately, this assumption is not practical since the slate length varies in the real world datasets. An additional notion of an item's merit m_i is considered in the literature, where, instead of $|G|$, the group exposure is normalized by the group merit $\sum_{i \in G} m_i$ in (2). Although the merit is understood as relevance labels in some of the previous works [37, 18], others explicitly leave it out [49]. Because, as discussed in [48] the understanding of merit depends on subjective factors, such as worldviews and on one's conception of equal opportunity, that may not be reduced to relevance labels. Therefore, we stick to the normalized form (2) of group discrepancy due to its simplicity and scale invariance w.r.t. slate length.

Probabilistic Ranking: In traditional LTR [20], a deterministic scoring function $f_\theta : \mathbb{R}^d \to \mathbb{R}$ predicts the items' scores $\hat{y}_i = f_\theta(x_i)$ for $i \in [n]$, then a ranking $\sigma : [n] \to [n]$ is assigned such that higher scored items are ranked lower, i.e., $\sigma(i) < \sigma(j)$ if $\hat{y}_i > \hat{y}_j$. However, in deterministic ranking, it may be implausible to achieve both higher NDCG and lower discrepancy in group exposure, since there are only a few positions with high exposure. Therefore, in fair LTR, a stochastic ranking system is used to measure the expected group discrepancy over many probable rankings for the same query. To sample a ranking $\sigma \in \mathcal{P}$, where $\mathcal{P}$ is the set of all permutations, a probabilistic policy π is constructed from the predicted score vector $\hat{\mathbf{y}}$. In particular, the Plackett-Luce (PL) distribution [31, 21] is used as the policy, wherein the probability mass of a ranking defined as

$$\pi(\sigma|\hat{\mathbf{y}}) = \prod_{k=1}^{n} \frac{\exp\left(\hat{y}_{\sigma^{-1}(k)}\right)}{\sum_{l=k}^{n} \exp\left(\hat{y}_{\sigma^{-1}(l)}\right)}, \quad \forall \sigma \in \mathcal{P}, \tag{PL}$$

which assumes the ranks are assigned by sampling n items without replacement from the softmax probability of their scores. Note, we index the items by i, j, but their ranks by k, l.

Problem Statement: The goal of fair LTR is to learn a scoring function f_θ whose (PL) policy maximizes the the expected NDCG (1), also called as *Ranking Utility* $U(\theta)$, while constraining the expected discrepancy in group exposures (2), also called as *Fairness violation* $F(\theta)$, within $\epsilon > 0$:

$$\max_\theta \; U(\theta), \quad \text{s.t.} \quad |F(\theta)| \leq \epsilon, \tag{3}$$

$$\text{where} \quad U(\theta) = \mathbb{E}_q[U(\theta|q)] = \mathbb{E}_q\left[\mathbb{E}_{\sigma\sim\pi(\cdot|\hat{\mathbf{y}}_q)}\left[M(\mathbf{y}_q, \sigma)\right]\right], \tag{4}$$

$$F(\theta) = \mathbb{E}_q[F(\theta|q)] = \mathbb{E}_q\left[\mathbb{E}_{\sigma\sim\pi(\cdot|\hat{\mathbf{y}}_q)}\left[D(\mathbf{m}_q, \sigma)\right]\right], \tag{5}$$

and the scores $\hat{\mathbf{y}}_q = f_\theta(\mathbf{x}_q)$. In practice, the outer expectations of (4) and (5) are reduced to empirical expectation over the queries available in a given dataset. Similarly, the inner expectations are

Table 1: Main notations

Notation	Description
n, n_q	Number of items for a generic, specific query q
d	Number of features of an item
$\mathbf{x}_q$	$\in \mathbb{R}^{n_q \times d}$, Features of items in query q
$\mathbb{Y}$	Range of relevance labels, e.g., $\{0, 1\}$
$\mathbf{y}_q$	$\in \mathbb{Y}^{n_q}$, Relevance labels of items for query q
f_θ, Θ	Scoring function, and its parameter space
$\hat{\mathbf{y}}_q$	$\in \mathbb{R}^{n_q}$, Predicted scores of items of query q
$\hat{Y}$	Predicted scores all query-items in a dataset
$\mathbf{x}, \mathbf{y}, \hat{\mathbf{y}}$	Features, labels, scores of items of a generic query
$\sigma, \mathcal{P}, \mathcal{S}$	A ranking on $[n]$, set of all rankings, a sample set
$\mathbf{\Delta}, \mathbf{v}$	Discount and bias vector of n ranked positions
M, D	Ranking metric, group exposure discrepancy
π	Ranking policy
U, F_{abs}	Ranking utility, absolute fairness loss
$\overline{U}, \overline{F}_{abs}$	Empirical mean of U and F_{abs} over all queries
g	Scalarization function for multiple objectives
$\lambda, \epsilon; \mathbf{r}$	Fairness weight, constraint; weights of cost vector
$\boldsymbol{\alpha}, s$	Objective coefficients, their smoothing factor

reduced to Monte Carlo estimates by sampling several rankings from the (PL) distribution, because for queries with large slate length n_q, it is infeasible to compute the true expectation over all the $n_q!$ rankings. [18] highlighted that although the overall fairness violation is constrained in (3), the querywise violations are not constrained. Because, the positive and negative expected group discrepancies $F(\theta|q)$ of different queries cancel each other to an overall lower violation. Therefore, to account for the querywise fairness violations, instead of taking the absolute value of overall expected group discrepancies as done in the literature [44, 37], we modify the formulation in (5) to take the absolute value in a querywise manner, i.e.,

$$F_{\text{abs}}(\theta) = \mathbb{E}_q\left[F_{\text{abs}}(\theta|q)\right] = \mathbb{E}_q\left[\left|\mathbb{E}_{\sigma\sim\pi(\cdot|\hat{\mathbf{y}}_q)}\left[D(\mathbf{m}_q, \sigma)\right]\right|\right], \tag{6}$$

and replace the constraint in (3) to $F_{\text{abs}}(\theta) < \epsilon$. As a result of this modification, the positive group discrepancy of one query cannot compensate for the negative value of another query.

***Linear Scalarization (LS):*.** The constraint optimization problem (3) is difficult for a scoring function with high dimensional parameter space, such as Support Vector machines, gradient boosted machines, Deep Neural Networks (DNN) etc. Therefore, it is relaxed to the LS form[37, 47, 44, 25, 18] using a penalty weight λ for the fairness violation:

$$\theta^*_\lambda = \arg\min_\theta \; 1 - U(\theta) + \lambda F_{abs}(\theta) = C_\lambda(\theta), \tag{7}$$

where the Lagrangian multiplier $\lambda > 0$ sets the priority for fairness task for the overall training cost C_λ. Although the link between λ and constraint upperbound ϵ can be recovered after optimization as $\epsilon_\lambda = F_{abs}(\theta^*_\lambda)$, the direct specification of ϵ is lost in (7).

Scoring Function: We use *Gradient Boosted Machine* (GBM) [14] to model the scoring function as many state-of-the-art production models in the industry use it. However, our development is general and can be easily extended to other models such as DNN. A GBM model is updated through the objective gradients w.r.t. the output scores $\hat{\mathbf{y}}$. As a result, we can treat the objectives U and F_{abs} as functions of the concatenated scores $\hat{Y} = \{\hat{\mathbf{y}}_\mathbf{q}\}_{q=1}^N$. This makes the querywise analysis in Section 4.4 easier. A GBM model consists of T decision trees:

$$f_{\boldsymbol{\theta}}^T(x) = \tau_{\theta^0}(x) - \sum_{t=1}^{T-1} \eta_t \tau_{\theta^t}(\mathbf{x}), \tag{8}$$

where η_t is the learning rate, τ_{θ^t} is the t^{th} tree parameterized by θ^t, and the full model parameter is $\boldsymbol{\theta} = \{\theta^t\}_{t=0}^{T-1}$. The training consists of sequentially learning these T trees. On the t^{th} iteration, instead of updating the parameters by the usual gradient descent rule $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta_t \nabla_{\boldsymbol{\theta}} C_\lambda$, where $\nabla_{\boldsymbol{\theta}} C_\lambda$ is the gradient of the overall cost in (7), in a GBM model, the tree τ_{θ^t} is learnt from the following training data:

$$\mathcal{D}_{\tau_{\theta^t}} = \left\{ \left(\mathbf{x}_q, \nabla_{\hat{\mathbf{y}}_q} C_\lambda \right) \right\}_{q=1}^m, \tag{9}$$

where the labels are gradients w.r.t. the latest scores $\hat{\mathbf{y}}_q = f_{\boldsymbol{\theta}}^t(\mathbf{x}_q)$. In other words, the update happens in the function space of trees, i.e., $f_{\boldsymbol{\theta}}^{t+1} = f_{\boldsymbol{\theta}}^t - \eta_t \tau_{\theta^t}$, instead of the parameter space. Consequently, we can analyze the ranking utility (4) and fairness violation (6) either as functions of the scores, $U(\hat{Y})$ and $F(\hat{Y})$, or as functions of the parameters, $U(\boldsymbol{\theta})$ and $F(\boldsymbol{\theta})$, depending on the context. Note, this interpretation of the objective functions is not limited only to GBM, as it can be generalized to any smooth parametric model through first order analysis of the objectives with a small learning rate η.

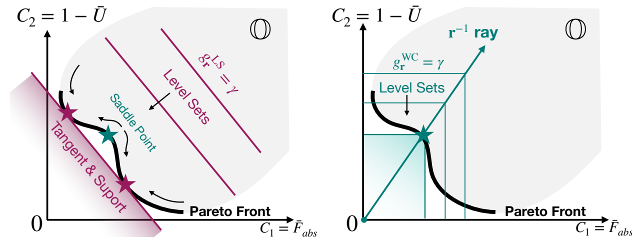
3.2 Multi-Objective Optimization

We consider the vector valued cost function $\mathbf{C} := [F_{abs}, 1 - U]^T$, and denote a point in the objective space as $\mathbf{c} \in \mathbb{R}^2$. Note that both cost functions are non-negative. Therefore, the set of all attainable cost vectors (range of $\mathbf{C}$), denoted as $\mathbb{O}$, is a subset of the positive quadrant $\mathbb{R}_+^2 := \{\mathbf{c} \in \mathbb{R}^2 \mid c_i \geq 0,\ \forall\, i \in [2]\}$.

Partial Ordering: The positive quadrant cone $\mathbb{R}_+^2$ is used to define a partial ordering relation on $\mathbb{R}^2$. For any two points $c^1, c^2 \in \mathbb{R}^2$, we write $c^1 \succcurlyeq c^2$, if c^1 lies in the positive cone pivoted at c^2, i.e., $c^1 \in \left\{ c^2 + c \mid c \in \mathbb{R}_+^2 \right\}$. In other words, $c^1 \succcurlyeq c^2 \iff c^1 - c^2 \in \mathbb{R}_+^K$, making $c_k^1 \geq c_k^2$ for every $k \in [K]$. Strict inequality $\mathbf{c}^1 \succ \mathbf{c}^2$ is written when, at least one $j \in [2]$, $c_j^1 > c_j^2$.

Pareto Frontier: A point $\mathbf{c} \in \mathbb{O}$ is said to be minimal if there exists no other point $\mathbf{c}' \in \mathbb{O}$ such that $\mathbf{c} \succ \mathbf{c}'$. The *Pareto Frontier* (PF) of $\mathbf{C}$ is defined as the set of all minimal points in $\mathbb{O}$, which could be a 1-dimensional manifold if connected [10, 23].

In order to make fair LTR adoptable to a wide range of applications having different levels of fairness requirements, the training algorithm should be able to attain the minimal cost vectors, $\mathbf{c}^*$s in $\mathbb{R}^2$, corresponding to a wide range of fairness constraints ϵs. Because, at a minimal point with $c_1^* = \epsilon$, the model achieves the best possible ranking utility (least c_2^*) without violating the fairness requirement. In other words, optimization algorithm should be able to approximate the PF.



(a) Linear Scalarization **(b) Chebyshev Scalarization**

Figure 1: Illustration of two types of preference specifications depicted in the objective space. Fig 1a shows how Linear scalarization can have non-unique Pareto optimal points for the same preference. Among the 3 Pareto optimal cost vectors, whose corresponding solutions θ^*s or $\hat{Y}^*$s are stationary points of $g_\mathbf{r}^{\text{LS}}$ defined in (7) or (12) respectively, the green solution is neither a local minimum nor a maximum. It is a saddle point for $g_\mathbf{r}^{\text{LS}}$, therefore cannot be attained. Fig 1b shows how Chebyshev scalarization can attain the green optimum by minimizing $g_\mathbf{r}^{\text{WC}}$ (13).

Multi-Gradient Combination: The MOO methods we consider in this work, viz., WC [41], EPO Search [24, 23] and WC-MGDA [27], are designed to approximate the PF even for non-convex MOO problems. Unlike LS, these methods adaptively combine the gradients of both objectives:

$$\nabla_{\hat{\mathbf{y}}_q} C_\lambda = \alpha_1 \nabla_{\hat{\mathbf{y}}_q} F_{abs} - \alpha_2 \nabla_{\hat{\mathbf{y}}_q} U, \tag{10}$$

where the combination coefficients α_1 and α_2 are computed in each iteration with the knowledge of λ, the cost vector and its gradients.

Coefficient Smoothing: [22] observed that in MOO algorithms exhibit non-smoothness in the costs trajectories if the values of combination coefficients α_1 and α_2 vary significantly in consecutive iterations. Therefore, to reduce the oscillations in the cost trajectories, a coefficient smoothing is applied as $\alpha_i^t \leftarrow s\alpha_i^{t-1} + (1-s)\alpha_i^t$ for $i \in [2]$, where where $s \in [0,1]$ is the smoothing factor and treated as a hyper-parameter while training the models.

4 FAIR LEARNING TO RANK WITH MOO

4.1 Shortcomings of Linear Scalarization

Consider a generalization of LS (10) as scalar valued function g:

$$g_\mathbf{r}^{\text{LS}}(\hat{Y}) = r_1 \frac{1}{N} \sum_{q=1}^N F_{abs}(\hat{\mathbf{y}}_q) + r_2 \frac{1}{N} \sum_{q=1}^N (1 - U(\hat{\mathbf{y}}_q)), \tag{11}$$

$$:= r_1 \bar{F}_{abs}(\hat{Y}) + r_2 (1 - \bar{U}(\hat{Y})). \tag{12}$$

where setting the preference vector to $\mathbf{r} = [\lambda, 1]$ converts the constrained fair LTR problem (3) to (7) in simple manner. However, with this simplicity one loses the direct control over the constraint upperbound ϵ. For a particular λ the corresponding upper bound can be estimated only after the optimization as $\epsilon_\lambda = \bar{F}_{abs}^* := \bar{F}_{abs}(\hat{Y}_\lambda^*)$, where the optimal scores $\hat{Y}_\lambda^*$ is obtained from the trained GBM model, but not before optimization. Because, the location of optimal cost vector $\mathbf{c}_\mathbf{r}^* = [\bar{F}_{abs}^*, 1 - \bar{U}^*]$ corresponding to $\mathbf{r}$ in the objective space depends on the structure of PF, as illustrated in Fig

1a; $\mathbf{c}_\mathbf{r}^*$ lies on the line with normal direction $\mathbf{r}$ that is both a tangent and a support to the PF. Therefore, for non-convex MOO, LS does not guarantee to reach every Pareto optimal solution, because not all tangents to the PF are also supports (see chapter 4 of [2] for detail).

4.2 Weighted Chebyshev Scalarization

In WC, the vector valued cost is scalarized to

$$g_\mathbf{r}^{\mathrm{WC}}(\hat{Y}) = \max\left\{r_1\overline{F}_{abs}(\hat{Y}),\ r_2(1-\overline{U}(\hat{Y}))\right\}. \quad (13)$$

In general, the minimal costs of WC w.r.t. $\mathbf{r}$ satisfies $r_1\overline{F}_{abs}(\hat{Y}_\mathbf{r}^*) = r_2(1-\overline{U}(\hat{Y}_\mathbf{r}^*))$, which can be deduced by analyzing the level sets of WC scalarization. Its $\gamma-$level set $\mathcal{L}_\gamma = \{\mathbf{c} \in \mathbb{R}^2 \mid \max_{j\in[2]} r_j c_j \le \gamma\}$ can also be written as $\{\gamma\mathbf{r}^{-1}\} - \mathbb{R}_+^2$, i.e., the negative quadrant pivoted at $\gamma\mathbf{r}^{-1}$. Therefore, if $\mathcal{L}_\gamma \cap \mathbb{O}$ is a singleton set then the corresponding solution is $\hat{Y}_\mathbf{r}^*$. In other words, the optimal vector $\mathbf{c}_\mathbf{r}^*$ lies on the $\mathbf{r}^{-1} := [r_1^{-1}, r_2^{-1}]$ ray, as illustrated in Figure 1b. This property of WC gives a direct relation between the fairness weight λ and the fairness violation upper bound ϵ.

Proposition 4.1. *Let $\gamma = \min \overline{F}_{abs}(\hat{Y})$ be the minimum value of absolute fairness violation when optimized without the ranking utility. Then, irrespective of the structure of the PF, the optimal solution of WC scalarization* (13) *w.r.t.* $\mathbf{r} = [\lambda, 1]$*, where $\lambda \le \frac{1}{\gamma}$, guarantees that $\overline{F}_{abs}^* = \epsilon_\lambda \le \frac{1}{\lambda}$.*

In addition to the above guarantee, WC also guarantees reachability to every solution in the PF, even for non-convex MOO [26]. Note, any value of $\epsilon \in [\gamma, \frac{1}{\gamma}]$ guarantees the existence of a Pareto optimal model on the PF. Let $\mathbf{c}^t = [\overline{F}_{abs}, (1-\overline{U})]$ at iteration t. Then the multi-gradient combination coefficients (10) of WC are

$$\alpha_j^t = \begin{cases} r_j, & \text{if } i = \arg\max_{i\in\{1,2\}} r_i c_i^t \\ 0, & \text{otherwise} \end{cases}, \quad \text{for } j = 1, 2. \quad (14)$$

Variants of WC Scalarization: Recent extensions to the WC method, such as EPO Search [24, 23] and WC-MGDA [27] also reach the preference specific Pareto optimal solution of $g_\mathbf{r}^{\mathrm{WC}}$ (13). To find the combination coefficients for the gradients, EPO Search solves a *Quadratic Programming* (QP) problem whereas WC-MGDA solves a *Second Order Cone Programming* (SOCP) problem in each iteration. We provide a brief explanation of EPO Search in the following and WC-MGDA in Appendix C.

Exact Pareto Optimal Search: In this method [23], the combination coefficients are decided based on an anchor direction $\mathbf{a}$ to control the first order movement in the objective space. Let $G = [\nabla\overline{F}_{abs}, -\nabla\overline{U}]$ and $\mathbf{c}^t = [\overline{F}_{abs}, (1-\overline{U})]$ at iteration t, then the coefficient is computed as

$$\boldsymbol{\alpha}^t = \underset{\alpha_i\ge 0, i\in[2]}{\arg\min}\ \|G^TG\boldsymbol{\alpha} - \mathbf{a}^t\|_2^2, \quad \text{s.t. } \alpha_1+\alpha_2 = 1, \quad (15a)$$

$$\text{and} \quad \mathbf{a}^t = \begin{cases} \mathbf{c}^t - \overrightarrow{\mathbf{r}^{-1}}\langle\mathbf{c}^t, \overrightarrow{\mathbf{r}^{-1}}\rangle, & \text{if cosine}(\mathbf{c}^t, \mathbf{r}^{-1}) \le \mu, \\ \mathbf{c}^t, & \text{otherwise} \end{cases} \quad (15b)$$

where $\overrightarrow{\mathbf{r}^{-1}}$ is the ℓ_2 normalized vector, and μ is a hyperparameter close to 1. Simply put, when the cost vector is far from the preference ray (small cosine similarity), the anchor direction $\mathbf{a}^t$ is the orthogonal error from $\mathbf{c}$ to $\mathbf{r}^{-1}$. On the other hand, when the cost reaches close to $\mathbf{r}^{-1}$, the anchor is simply the cost vector. The search direction $G\boldsymbol{\alpha}^t$ results in a first order movement along $G^TG\boldsymbol{\alpha}^t$ in the objective space as $\mathbf{c}^t - \eta G^TG\boldsymbol{\alpha}^t$ for a small learning rate η. As a result, when $\mathbf{c}^t$ is far from $\mathbf{r}^{-1}$, the next cost vector $\mathbf{c}^{t+1}$ moves closer to the $\mathbf{r}^{-1}$ ray, and when $\mathbf{c}^t$ is near to $\mathbf{r}^{-1}$, it moves closer to the PF (see [23] for details).

4.3 Querywise Weighted Chebyshev

If fairness objective is considered as a regularization in the model training for the LTR task, then the WC formulation (13) offers better regularization than LS (7), because WC searches for only those models whose cost vectors ideally (practically) lie on (near) the $\mathbf{r}^{-1}$ ray. However, WC achieves this on average: the mean of all querywise cost vectors lies on $\mathbf{r}^{-1}$.

We introduce an extension of the WC method that promotes the cost vectors of individual queries to lie on $\mathbf{r}^{-1}$, which can further regularize the model. We propose the ***Q****uerywise* ***W****eighted* ***C****hebyshev* (QWC) scalarization, defined as

$$g_\mathbf{r}^{QWC}(\hat{Y}) := \frac{1}{N}\sum_{q=1}^{N} \max\left\{r_1F_{abs}(\hat{\mathbf{y}}_q),\ r_2(1-U(\hat{\mathbf{y}}_q))\right\}, \quad (16)$$

where the max operator is taken inside the empirical expectation over the queries. Unlike the multi-gradient combination coefficients of WC in (14), in QWC, it is done querywise, where $\alpha_{q,1}^t$ and $\alpha_{q,2}^t$ are set using the querywise cost vectors $\mathbf{c}_q^t = [F_{abs}(\hat{\mathbf{y}}_q), 1-U(\hat{\mathbf{y}}_q)]$ for all $q \in [N]$.

Variants of QWC Scalarization: Similar to the variants of WC, we extend QWC to *Querywise EPO Search* and *Querywise WC-MGDA*. In querywise EPO Search, instead of solving one QP (15), N QP problems are solved in each iteration using the querywise gradients $G_q = [\nabla F_{abs}(\hat{\mathbf{y}}_q), -\nabla U(\hat{\mathbf{y}}_q)]$ to obtain the $\boldsymbol{\alpha}_q^t$ for $q \in [N]$. Similarly, in Querywise WC-MGDA, N SOCP problems are solved in each iteration. However, these QP and SOCP based querywise methods cannot be scaled to real world datasets having a large number of queries. Therefore, we develop a matrix inversion based EPO Search, and extend it to the querywise approach, where solving QPs can be avoided. This makes the computation of querywise coefficients $\boldsymbol{\alpha}_q^t$s parallelizable, and hence scalable to large datasets.

Matrix Inversion based EPO Search: To make the querywise EPO Search scalable, we propose a simplification of the EPO Search algorithm. For the non-querywise approach, instead of solving the QP in (15), we obtain the coefficients by

$$\boldsymbol{\alpha}^t = (G^TG)^{-1}\mathbf{a}^t, \quad (17)$$

and then normalize the $\boldsymbol{\alpha}^t$. This formulation assumes that the constraint $\alpha_1+\alpha_2 = 1$, which was inherited from the MGDA algorithm [9], is replaced by the constraint $\|\boldsymbol{\alpha}\|_2 = 1$. For querywise EPO Search, we parallelize matrix inversion of $G_q^TG_q$ for $q \in [N]$ to efficiently compute the coefficients as $\boldsymbol{\alpha}_q^t = (G_q^TG_q)^{-1}\mathbf{a}_q^t$, where the querywise anchor direction $\mathbf{a}_q^t$ is computed similar to (15b) using $\mathbf{c}_q^t$. We avoid the corner case of having a negative a coefficient in $\boldsymbol{\alpha}_q^t$, by replacing them with the coefficients of the QWC for the corresponding query. However, we found that the anchor directions and gradients are robust to such occurrence due to the normalized formulations of both the ranking utility and fairness loss.

4.4 Querywise vs. Non-Querywise WC

To analyze their differences, we first formalize the range $\mathbb{O}_q \subset \mathbb{R}^2$ of querywise cost $\mathbf{C}(\cdot|q) = [F_{abs}(\cdot|q), 1-U(\cdot|q)]^T$. Given a query q, the ranking utility and fairness violation without the Monte Carlo approximation can be written as

$$U(\hat{\mathbf{y}}_q) = \langle \pi(\cdot|\hat{\mathbf{y}}_q), M(\mathbf{y}_q, \cdot)\rangle \tag{18}$$

$$F(\hat{\mathbf{y}}_q) = \langle \pi(\cdot|\hat{\mathbf{y}}_q), D(\mathbf{m}_q, \cdot)\rangle, \tag{19}$$

where $\pi(\cdot|\hat{\mathbf{y}}_q) \in \mathbb{R}^{n_q!}$ is a vector in the $n_q! - 1$ dimensional probability simplex. Recall, $\pi(\cdot|\hat{\mathbf{y}}_q)$ is a probability mass function over all possible permutations/rankings of n_q items. Similarly, $M_q := M(\mathbf{y}_q, \cdot)$ and $D_q := D(\mathbf{m}_q, \cdot)$ are vectors in $\mathbb{R}^{n_q!}$, which are the ranking metrics and group exposure discrepancies respectively for all possible rankings of n_q items. The range of PL function given by

$$\pi(\cdot|\mathbb{R}^{n_q}) := \left\{\pi(\cdot|\hat{\mathbf{y}}_q) \in \mathbb{R}^{n_q!} \;\middle|\; \hat{\mathbf{y}}_q \in \mathbb{R}^{n_q}\right\} \tag{20}$$

is an $n_q - 1$ dimensional hypersurface [42] in the probability simplex. If $\mathbf{C}(\cdot|\mathbf{q})$ is treated as a function of the scores, then the querywise range $\mathbb{O}_q$ can be realized by first projecting $\pi(\cdot|\mathbb{R}^{n_q})$ onto the 2d hyperplane of ranking metric vector M_q and group discrepancy vector D_q, then applying the necessary transforms, such as orthogonalizing $M_q D_q$, converting to $1 - U(\hat{\mathbf{y}}_q)$ and $|F(\hat{\mathbf{y}}_q)|$ for all $\hat{\mathbf{y}}_q \in \mathbb{R}^{n_q}$. However, if $\mathbf{C}(\cdot|q)$ is treated as a function of the model parameters θ, then $\mathbb{O}_q$ requires the projection of $\pi(\cdot|\hat{\mathbb{Y}}_q)$ onto the $M_q D_q$ hyperplane, where $\hat{\mathbb{Y}}_q := \{f_\theta(\mathbf{x}_q) \in \mathbb{R}^{n_q} \mid \theta \in \Theta\} \subset \mathbb{R}^{n_q}$ is the range of scoring function when treated as a function of $\theta \in \Theta$.

Update Directions in Objective Space: The difference between the querywise and non-querywise WC methods can be realized by juxtaposing the ranges $\mathbb{O}_q$ for $q \in [N]$ on to a common $\mathbb{R}^2$ objective space, and analyzing the first order movements of the querywise cost vectors for a particular model parameter; this is depicted in Figure 2. In WC, the update direction of the mean cost vector $\bar{\mathbf{c}}$ is inherited to all the queries. However, in QWC the update direction is decided for individual queries.

Regularization in Parameter Space: We denote the range of overall non-querywise cost $\overline{\mathbf{C}} = [\overline{F}_{abs}, 1 - \overline{U}]^T$ by $\overline{\mathbb{O}}$, which is the pointwise mean of all $\mathbb{O}_q$ for $q \in [N]$. Given a preference vector $\mathbf{r}$, we analyze how the two methods constrain the parameter search space Θ by defining three sets:

$$\Theta^{\mathbf{r}} := \overline{\mathbf{C}}^{-1}(\overline{\mathbb{O}} \cap \overrightarrow{R}), \quad \text{where ray } \overrightarrow{R} := \{s\mathbf{r}^{-1} \mid s > 0\}, \tag{21}$$

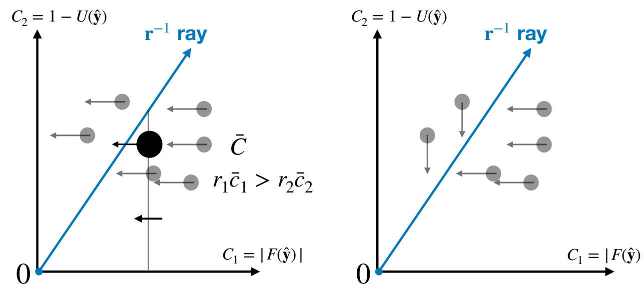
$$\Theta^{\mathbf{r}}_q := \mathbf{C}_q^{-1}(\mathbb{O}_q \cap \overrightarrow{R}), \quad \text{where } \mathbf{C}_q := \mathbf{C}(\cdot|q), \tag{22}$$

$$\Theta^{\mathbf{r}}_{\mathbb{Q}} := \cap_{q\in\mathbb{Q}}\, \Theta^{\mathbf{r}}_q, \quad \text{where } \mathbb{Q} \text{ is set of all queries.} \tag{23}$$

$\Theta^{\mathbf{r}}$ and $\Theta^{\mathbf{r}}_{\mathbb{Q}}$ are restrictions on the optimal model by WC and QWC.

Proposition 4.2. *The restriction on the optimal model offered by QWC is stricter than that of WC, i.e.,* $\Theta^{\mathbf{r}}_{\mathbb{Q}} \subset \Theta^{\mathbf{r}}$.

Note, the LTR problem is an ill-posed problem. Because, different permutations of a list of items can have the same NDCG value due to course grained relevance labels, e.g., binary or five star ratings.



(a) WC **(b) QWC**

Figure 2: Illustration of the movement directions of the queries in the objective space in one iteration. In WC, all the queries minimize the same objective (C_1 in this case) as decided by the relative values of the mean cost vector. In QWC, the objectives for optimization are chosen individually by the queries.

Proposition 4.2 formalizes how QWC tackles the ill-posedness with more regularization than WC by comparing $\Theta^{\mathbf{r}}_{\mathbb{Q}}$ and $\Theta^{\mathbf{r}}$. An empty set $\Theta^{\mathbf{r}}_{\mathbb{Q}}$ is a possibility, wherein the minimal solution may not exist.

Inductive Bias: In QWC, the influence of fairness task on the ranking task, i.e., whether reducing $|F|$ increases U or not, is captured by the angle between the two vectors in $\mathbb{R}^{n_q!}$, group discrepancies $D(\cdot|q)$ and ranking metrics $M(\cdot|q)$. Since U is maximized and F_{abs} is minimized, when D_q is orthogonal to M_q, the positive influence of the fairness task is maximum. Note, the ranking metric vector $M(\cdot|q)$ (NDCGs) lies in the positive orthant while the group discrepancy vector $D(\mathbf{m}_q, \cdot)$ does *not* lie in it, which facilitates a higher angle between the vectors. However, such treatment is not apparent in the WC method as the update happens for the mean cost.

5 SCALABLE IMPLEMENTATION

5.1 Gradient Approximation

Similar to [37], we approximate the gradient using the log-derivative trick, originally proposed in the REINFORCE algorithm [43], over a subset $\mathcal{S} \subset \mathcal{P}$ of permutations sampled from the (PL) distribution:

$$\frac{\partial U(\hat{\mathbf{y}})}{\partial \hat{y}_i} = \sum_{\sigma\in\mathcal{P}} \frac{\partial \pi(\sigma|\hat{\mathbf{y}})}{\partial \hat{y}_i} M(y, \sigma) = \mathbb{E}_\sigma\left[\frac{\partial \log(\pi(\sigma|\hat{\mathbf{y}}))}{\partial \hat{y}_i} M(\mathbf{y}, \sigma)\right] \tag{24}$$

$$\Rightarrow \widehat{\frac{\partial U}{\partial \hat{y}_i}} = \frac{1}{|\mathcal{S}|} \sum_{\sigma\in\mathcal{S}} \frac{\partial \log(\pi(\sigma|\hat{\mathbf{y}}))}{\partial \hat{y}_i} M(\mathbf{y}, \sigma). \tag{25}$$

The gradient of absolute fairness violation loss F_{abs} (6) for a query (inner expectation in (6)), requires the sign of non-absolute fairness violation F (inner expectation in (5)):

$$\frac{\partial F_{abs}(\hat{\mathbf{y}})}{\partial \hat{y}_i} = \text{sign}(F(\hat{\mathbf{y}}))\ \mathbb{E}_\sigma\left[\frac{\partial \log(\pi(\sigma|\hat{\mathbf{y}}))}{\partial \hat{y}_i} D(\mathbf{m}, \sigma)\right] \tag{26}$$

$$\Longrightarrow \widehat{\frac{\partial F_{abs}}{\partial \hat{y}_i}} = \text{sign}\left(\widehat{F(\hat{\mathbf{y}})}\right) \frac{1}{|\mathcal{S}|} \sum_{\sigma\in\mathcal{S}} \frac{\partial \log(\pi(\sigma|\hat{\mathbf{y}}))}{\partial \hat{y}_i} D(\mathbf{m}, \sigma), \tag{27}$$

Table 2: Descriptions of the datasets: number queries (train/test/valid), slate lengths (min/median/max), number of features, features used for grouping, and grouping method.

Dataset	Queries	Slate length	Features	Grouping Attribute	Groups
German	500/500/500	20/20/20	58	Purpose of loan applicant	radio/television
MSLR-WEB30K	18.9K/6.3K/6.3K	1/243/1251	136	Document body length	30 percentile threshold
E-Commerce	735K/155K/187K	1/79/309	60	Product price	40 percentile threshold

where $\widehat{F(\hat{\mathbf{y}})} = \frac{1}{|\mathcal{S}|}\sum_{\sigma\in\mathcal{S}} D(\mathbf{m},\sigma)$. When training with multiple queries, this sign should be computed for every query separately.

5.2 Hessian Diagonal Approximation

In addition to the gradient information, morden GBM based packages, such as LightGBM [16] and XGBoost [7], require the second order derivatives w.r.t the scores for efficient training. We approximate the double derivatives by differentiating the estimates of gradients in (25) and (27) as

$$\frac{\partial}{\partial \hat{y}_i}\frac{\widehat{\partial U}}{\partial \hat{y}_i} = \frac{1}{|\mathcal{S}|}\sum_{\sigma\in\mathcal{S}}\frac{\partial^2 \log(\pi(\sigma|\hat{\mathbf{y}}))}{\partial \hat{y}_i^2}M(\mathbf{y},\sigma), \quad \text{and} \tag{28}$$

$$\frac{\partial}{\partial \hat{y}_i}\frac{\widehat{\partial F_{abs}}}{\partial \hat{y}_i} = \text{sign}\left(\widehat{F(\hat{\mathbf{y}})}\right)\frac{1}{|\mathcal{S}|}\sum_{\sigma\in\mathcal{S}}\frac{\partial^2 \log(\pi(\sigma|\hat{\mathbf{y}}))}{\partial \hat{y}_i^2}D(\mathbf{m},\sigma) \tag{29}$$

respectively. Another method is to apply the log-derivative trick twice and then approximate with a sample set. However, we found this method does not improve the training (see Appendix B). Instead of using the modern autodiff packages we analytically derived the double derivatives of the log-probabilities, which makes the computation significantly faster. Similar to the gradient combination in (10), we combine (28) and (29) with the α coefficients.

5.3 Sampling with Gumbel Max Trick

The role of sampling is significant in fair LTR since the estimation of objectives, gradients and hessian-diagonals require many ranking samples. All PL policy based previous works in fair LTR literature [37, 44] have used the definition of (PL), sampling without replacement, to sample a permutations. However, this is not scalable to practical scenarios, since the for loop cannot be parallelized over n.

We avoid the trap of sampling without replacement by the Gumbel max trick [40]. A sample ranking can be simply obtained by adding Gumbel random noises to the scores, and then sorting the noisy scores [17]. In particular, if $\zeta_i \sim \text{Gumbel}(0,1) := e^{-e^{-\zeta}}$ for $i \in [n]$ random noises distributed according to the Gumbel distribution, then (PL) probability of $\sigma = \text{argsort}(\hat{\mathbf{y}} + \boldsymbol{\zeta})$ is the same as the order statistical probability of $\hat{\mathbf{y}} + \boldsymbol{\zeta}$, i.e.,

$$\pi(\sigma|\hat{\mathbf{y}}) = P\left(\hat{y}_{\sigma^{-1}(1)} + \zeta_{\sigma^{-1}(1)} > \cdots > \hat{y}_{\sigma^{-1}(n)} + \zeta_{\sigma^{-1}(n)}\right). \tag{30}$$

With this trick we can efficiently draw many samples, resulting in a larger $\mathcal{S}$ to improve the approximations of objectives, gradients and hessian-diagonals in a scalable manner.

6 EXPERIMENTS

We compare the querywise and non-querywise MOO methods proposed in Section 4 against the traditional approach of Linear Scalarization (LS) for training Fair LTR models. Although previous works, such as [37, 44, 47], differ in their exact formulation of fairness loss and pre-processing techniques for data debiasing, all of them essentially used LS to combine objectives. Our implementation uses the LightGBM package [15] for the GBM model and PyTorch [30] for derivatives computation.

6.1 Setup

Datasets: We used three datasets: German Credit [12], Microsoft Learning to Rank Dataset (MSLR-WEB30k) [32], and another E-Commerce dataset similar to [39, 29, 28]. An overview of these datasets are given in Table 2.

- The German credit Dataset is originally a binary classification dataset consisting of 1000 loan applicants, where the labels are their creditworthiness. Similar to [37], we convert it to an LTR dataset by constructing 500 queries from train, test, and validation sets, respectively, where each query consists of 20 randomly selected applicants such that the ratio of non-creditworthy individuals to creditworthy individuals is 9:1. The grouping is done by a binary feature indicating whether the purpose of the loan applicant is radio or television. The ratio of applicants in two groups is around 8:2.
- The MSLR-WEB30k dataset contains a large number of queries from Bing with manually judged relevance labels for retrieved webpages. Unlike the previous works in fair LTR [44, 37], we do not pre-process the data to have fixed slate lengths for every query, because it does not reflect the real world scenario. Instead we use the original dataset to showcase the scalability of our methods. We group the documents based on body length of the webpages and threshold at 30$^{\text{th}}$ percentile of the overall dataset. We selected the threshold such that not many queries will be left out of having both the groups: only 467 out of ∼19K queries in the training dataset did not have both groups.
- The E-commerce dataset was collected in 2021. Each query is associated with a set of products impressed by customers in a search session. The query-product dependent features are constructed from information such as product description, their textual matches with the query, and customer's purchase decision. We sampled ∼735K queries for training, ∼155K for testing and ∼187K for validation. The grouping was done based on the product price, so that the disparity in exposure of higher priced and lower priced products can be minimized.

Competing Algorithms: In non-querywise MOO, we tested four methods on all three datasets: WC [41], WC-MGDA [27], the QP based EPO Search [23] (EPO-QP), and our matrix inversion version based of the EPO Search (EPO).

Table 3: Hyperparameters

Hyperparameters	German	MSLR-WEB30K	E-Commerce
max leaf nodes	50	50	50
# of iterations	500	300	150
learning rate	0.1	0.1	0.1
Fairness weight λ	[1e-5, 0.1, 1, 5, 10, 15, $\cdots$, 80]		
Smoothing factor s	[0.1, 0.3, 0.5, 0.7, 0.9]		

In querywise MOO, we tested only QWC, and our matrix inversion based **Q**uerywise EPO Search (QEPO), in all three datasets. The QP based QEPO-QP, and SOCP based **Q**uerywise WC-MGDA (QWC-MGDA) are not scalable due to lack of libraries that support parallelization of thousands of separate QP or SOCP problems. Therefore, we tested these two methods only on the small German Credit dataset.

Hyperparameters: For fair comparison, we kept the GBDT model parameters, such as learning rate, number of maximum leaf nodes, and number of training iterations, consistent for all the algorithms for a given dataset as detailed in Table 3. The variations in hyperparameters such as fairness weight λ and coefficient smoothing s were the same across all MOO methods.

6.2 Evaluation

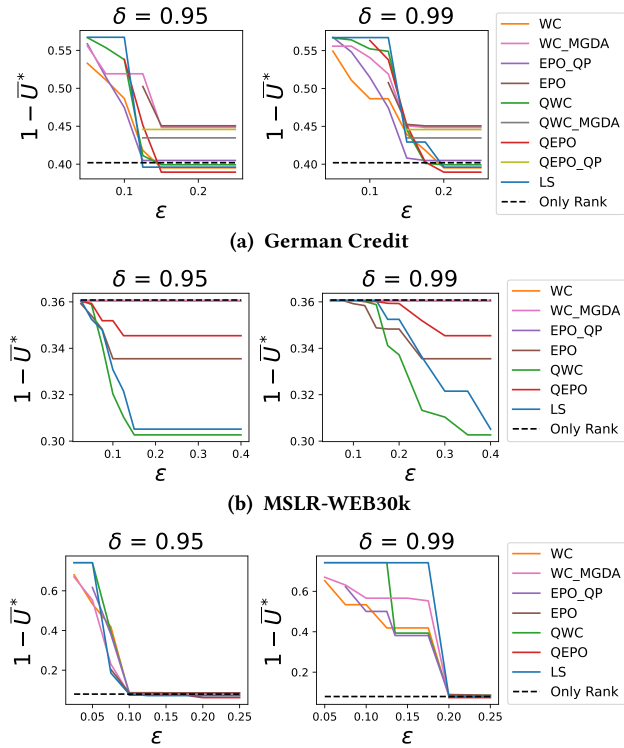
Instead of strictly constraining the fairness violation for every query, we evaluate the methods for constraining the violation for most of the queries while maximizing the ranking cost:

$$\max_{\theta} \overline{U}(\theta) \quad \text{s.t.} \quad P(F_{abs}(\theta|q) \leq \epsilon) \geq \delta, \tag{31}$$

where $\overline{U}(\theta) = \mathbb{E}_q[U(\theta|q)]$ is computed empirically. The probabilistic constraints are imposed empirically for preset values of ϵ and δ. First, for a trained model θ^*, we obtain the δ^{th} quantile value of $F_{abs}(\theta^*|q)$ over the queries in the test dataset, and denote it by $F^{\delta}_{abs}(\theta^*)$. Then, among all the models trained with the λs, we report the model that has best ranking utility and satisfies ϵ restriction:

$$\overline{U}^* = \max_{\lambda \in \Lambda | F^{\delta}_{abs}(\theta^*_\lambda) \leq \epsilon} \overline{U}(\theta^*_\lambda), \tag{32}$$

where Λ is the grid of λ values in Table 3 used for training. We plot the ranking cost $1 - \overline{U}^*$ against a preset grid of upper bounds for two confidence values $\delta = 0.95$ and 0.99. Results for different MOO methods are plotted in Figure 3. When $\lambda = 0$, i.e., the model is trained only for the relevance task, the result is plotted as a horizontal black dashed line, labeled as 'Only Rank'. **Training results** are given in **Appendix D**.



(a) German Credit

(b) MSLR-WEB30k

(c) E-commerce

Figure 3: Pareto Fronts obtained by different MOO methods on 3 datasets, where the axes are average ranking cost $1 - \overline{U}^*$ (32) and upper bound ϵ of probabilistic fairness violation (31).

6.3 Does WC Outperform LS?

In Figure 3, we first observe in all three datasets, none of the algorithms generate a convex optimal frontier. This is an empirical validation that the Fair LTR problem formulated using the Plackett-Luce policy is a non-convex problem. Therefore, in all three datasets, a major portion of the PF is not reached by LS.

For German Credit and E-Commerce datasets, the non-querywise MOO methods WC, WC-MGDA and EPO-QP significantly outperform the LS method in the restrictive fairness regions (lower ϵ values). However, for the less restricted fairness regions (higher ϵ), LS has better performance. For MSLR datasets, the ranking utility $\overline{U}^*$ of non-querywise MOO methods, except our matrix inversion based EPO, did not increase with higher ϵ values. In fact, they performed similarly to the 'Only Rank' method ($\lambda = 0$), whose ranking utility did not improve during training. The similarity between WC and Only Rank can be attributed to the single objective selection of WC method. For MSLR dataset, this objective happens to be only the ranking objective. We reason the similarity between WC and its QP based variants is due to the ℓ_1 constraint (15a), which promotes sparsity [11] in the coefficients, thereby virtually choosing only one objective. Whereas, our matrix inversion based EPO does not have the ℓ_1 constraint, therefore produces non-sparse coefficients to combine the gradients, and improves the ranking utility.

6.4 Does Querywise WC Improve Relevance?

In all three datasets we notice, for the restrictive fairness regions (low ϵ), the non-querywise MOO methods have better relevance than the querywise methods. This is due to the heavily regularized setup of querywise formulation, where the preference specific ideal solution may not exist, as explained in Section 4.4. However, when the fairness restrictions are slacked (high ϵ), QWC has better ranking utility than the non-querywise methods. Unlike the non-querywise methods, the QP based algorithms, QEPO-QP and QWC-MGDA in the German dataset, do not have similar performance as QWC. Whereas the proposed matrix inversion based QEPO has

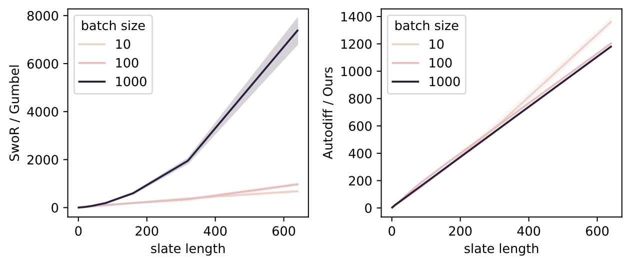
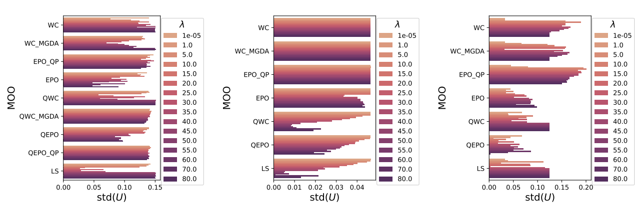


Figure 5: Computational Improvements (ratio of time taken) for, (left) sampling from PL distribution by sampling without replacement compared to Gumbel max trick, and (right) compute the double derivatives using automatic differentiation compared to our implementation.

Table 4: Best Ranking Utilities $\overline{U}^*$, when trained with non-querywise MOO, querywise MOO, LS and only ranking task.

	German	MSLR-WEB30K	E-Commerce
WC	0.6048	0.6393	0.9204
WC-MGDA	0.5514	0.6393	0.9204
EPO-QP	0.5951	0.6396	0.9146
EPO	0.5550	0.6645	0.9141
QWC	0.6008	**0.6973**	0.9196
QWC-MGDA	0.5653	–	–
QEPO-QP	0.5542	–	–
QEPO	**0.6105**	0.6546	**0.9381**
LS	0.6039	0.6948	0.9279
Only Rank	0.5981	0.6392	0.9204



(a) German **(b) MSLR Credit** **(c) E-Commerce**

Figure 4: Standard Deviations of the ranking utility

similar performance to QWC. The best ranking utilities in moderate fairness restriction regions of Figure 3 are reported in Table 4. The querywise methods, i.e., QEPO for German and E-Commerce, and QWC for MSLR-WEB30K, have the highest improvement from the single task learning (Only Rank) result as compared to others.

An interesting observation is, even other MOO methods, e.g., LS and WC, have better ranking utility than the Only Rank method. This result advocates the use of fairness restriction in moderation to improve the relevance of the LTR model.

In Figure 4, we compare the standard deviation of the ranking utilities over the queries in the test dataset, when the model was trained with different values of fairness weight λ. We see that in MSLR-WEB30k and E-Commerce data, the querywise methods have lesser standard deviations as compared to the non-querywise methods. In German data the comparison is not conclusive. The QP based querywise methods have higher standard deviation as compared to QWC and the proposed matrix inversion based QEPO.

6.5 How Efficient is Our Implementation?

We measure computational efficiency in two aspects. First, we answer how much faster is the Gumbel max trick as compared to the sampling without replacement method for sampling permutations from PL distribution. Second, how much faster is our analytical formulation of double-derivatives as compared to the automatic differentiation? We consider different batch sizes (number of queries), slate lengths (number of items) and random scores, and plot the results in Figure 5. Each configuration is run for 5 times on a NVIDIA Tesla V100 GPU machine parallelizing wherever possible. We see significant improvements in both aspects.

In theory, the improvement in Gumbel max trick compared to sampling without replacement should be linear, since the former avoids the loop over the slate length that cannot be avoided in the later. But for higher batch sizes (e.g. 1000 in left Figure 5), we attribute the deviation from linear improvement to the time required for excessive memory allocation in sampling w/o replacement.

We used the automatic differentiation of PyTorch [30] to compute the hessian diagonals. It computes each element of the hessian diagonal by a hessian vector product that involves two forward/backward passes. This requires a for loop over the slate length to compute all the double derivatives. Whereas our analytical formulation vectorizes the computations and avoids the for loop. Therefore, there is a linear improvement. Autodiff underperforms due to the intricacies of creating and maintaining a computational graph.

7 CONCLUSION

We demonstrated that multi-objective optimization algorithms such as Weighted Chebyshev scalarization based methods achieve better relevance in ranking as compared to the linear scalarization used in the literature. We developed querywise methods and analyzed their differences from the non-querywise methods. Through the querywise methods, we discovered a key insight: the seemingly competing task of fairness, when imposed in moderation, benefits the relevance task, instead of deteriorating it. Moreover, our efficient implementation of querywise and non-querywise MOO methods can train fair LTR models on large-scale real world datasets.

Our research opens several avenues for future exploration, such as examining the bias in the gradient estimates for the absolute fairness loss, exploring the conditions for the non-existence of a solution for QWC, and developing a splitting strategy of the queries (non-uniform mini-batching of the training dataset) such that the solution exists. We will make the source code public available.

8 ACKNOWLEDGEMENT

This research was conceived and conducted in an internship program at Amazon and later the first author received support for attending the conference from a Singapore National Research Foundation (NRF) Fellowship (Grant Number: A-0008064-00-00).

REFERENCES

[1] Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. 2018. Equity of attention: amortizing individual fairness in rankings. *CoRR*, abs/1805.01788. arXiv: 1805.01788.

[2] Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.

[3] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning* (ICML '07). Association for Computing Machinery, Corvalis, Oregon, USA, 129–136.

[4] Rich Caruana. 1993. Multitask learning: a knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on International Conference on Machine Learning* (ICML'93). Morgan Kaufmann Publishers Inc., Amherst, MA, USA, 41–48.

[5] Carlos Castillo. 2019. Fairness and transparency in ranking. *SIGIR Forum*, 52, 2, (Jan. 2019), 64–71.

[6] L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. 2017. Ranking with fairness constraints. *CoRR*, abs/1704.06840. arXiv: 1704.06840.

[7] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754. arXiv: 1603.02754.

[8] Wei Chen, Tie-yan Liu, Yanyan Lan, Zhi-ming Ma, and Hang Li. 2009. Ranking measures and loss functions in learning to rank. In *Advances in Neural Information Processing Systems*. Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, (Eds.) Vol. 22. Curran Associates, Inc.

[9] Jean-Antoine Désidéri. 2009. Multiple-Gradient Descent Algorithm (MGDA). Research Report. INRIA, (June 2009), 17.

[10] Chaosheng Dong and Bo Zeng. 2020. Expert learning through generalized inverse multiobjective optimization: models, insights, and algorithms. In *International Conference on Machine Learning*.

[11] David L Donoho and Michael Elad. 2003. Optimally sparse representation in general (nonorthogonal) dictionaries via $\mathcal{L}_1$ minimization. *Proceedings of the National Academy of Sciences*, 100, 5, 2197–2202.

[12] Dheeru Dua and Casey Graff. 2017. UCI machine learning repository. (2017).

[13] Shady Elbassuoni, Sihem Amer-Yahia, Christine El Atie, Ahmad Ghizzawi, and Bilel Oualha. 2019. Exploring fairness of ranking in online job marketplaces. In *Advances in Database Technology - 22nd International Conference on Extending Database Technology, EDBT 2019, Lisbon, Portugal, March 26-29, 2019*. Melanie Herschel, Helena Galhardas, Berthold Reinwald, Irini Fundulaki, Carsten Binnig, and Zoi Kaoudi, (Eds.) OpenProceedings.org, 646–649.

[14] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

[15] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: a highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (NIPS'17). Curran Associates Inc., Long Beach, California, USA, 3149–3157.

[16] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: a highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*. I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, (Eds.) Vol. 30. Curran Associates, Inc.

[17] Wouter Kool, Herke van Hoof, and Max Welling. 2020. Ancestral gumbel-top-k sampling for sampling without replacement. *Journal of Machine Learning Research*, 21, 47, 1–36.

[18] 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* (WWW '22). Association for Computing Machinery, Virtual Event, Lyon, France, 3520–3530.

[19] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. 2019. Pareto multi-task learning. In *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc.

[20] Tie-Yan Liu. 2009. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3, 3, (Mar. 2009), 225–331.

[21] R. Duncan Luce. 1959. *Individual choice behavior; a theoretical analysis.* Wiley, New York.

[22] Debabrata Mahapatra, Chaosheng Dong, Yetian Chen, and Michinari Momma. 2023. Multi-label learning to rank through multi-objective optimization. In *Proceedings of the 29th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[23] Debabrata Mahapatra and Vaibhav Rajan. 2021. Exact pareto optimal search for multi-task learning: touring the pareto front. (2021). arXiv: 2108.00597.

[24] Debabrata Mahapatra and Vaibhav Rajan. 2020. Multi-task learning with user preferences: gradient descent with controlled ascent in pareto optimization. In *Proceedings of the 37th International Conference on Machine Learning*.

[25] Omid Memarrast, Ashkan Rezaei, Rizal Fathony, and Brian Ziebart. 2021. Fairness for robust learning to rank. *arXiv preprint arXiv:2112.06288*.

[26] Kaisa Miettinen. 1998. *Nonlinear multiobjective optimization. International series in operations research and management science*. Vol. 12. Kluwer.

[27] Michinari Momma, Chaosheng Dong, and Jia Liu. 2022. A multi-objective / multi-task learning framework induced by pareto stationarity. In *Proceedings of the 39th International Conference on Machine Learning*.

[28] Michinari Momma, Ali Bagheri Garakani, Nanxun Ma, and Yi Sun. 2020. Multi-objective relevance ranking via constrained optimization. In *The Web Conference 2020*.

[29] Michinari Momma, Alireza Bagheri Garakani, and Yi Sun. 2019. Multi-objective relevance ranking. In *eCOM@ SIGIR*.

[30] Adam Paszke et al. 2019. Pytorch: an imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*. H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, (Eds.) Curran Associates, Inc., 8024–8035.

[31] R. L. Plackett. 1975. The analysis of permutations. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 24, 2, 193–202.

[32] Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 datasets. *CoRR*, abs/1306.2597.

[33] Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098. arXiv: 1706.05098.

[34] Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc.

[35] Ashudeep Singh and Thorsten Joachims. 2017. Equality of Opportunity in Rankings. In *Workshop on Prioritizing Online Content (WPOC) at NIPs*.

[36] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD '18). Association for Computing Machinery, London, United Kingdom, 2219–2228.

[37] 2019. *Policy learning for fairness in ranking. Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA, 11 pages.

[38] Ryosuke Sonoda. 2021. A pre-processing method for fairness in ranking. *CoRR*, abs/2110.15503. arXiv: 2110.15503.

[39] Daria Sorokina and Erick Cantu-Paz. 2016. Amazon search: the joy of ranking products. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR '16). Association for Computing Machinery, Pisa, Italy, 459–460.

[40] 1954. Statistical theory of extreme values and some practical applications. lectures by emit j. gumbel. national bureau of standards, washington, 1954. 51 pp. diagrams. 40 cents. *The Journal of the Royal Aeronautical Society*, 58, 527, 792–793.

[41] Ralph E. Steuer. 1989. The Tchebycheff procedure of interactive multiple objective programming. In *Multiple Criteria Decision Making and Risk Analysis Using Microcomputers*. Birsen Karpak and Stanley Zionts, (Eds.) Springer.

[42] Bernd Sturmfels and Volkmar Welker. 2012. Commutative algebra of statistical ranking. *Journal of Algebra*, 361, 264–286.

[43] Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8, 3–4, (May 1992), 229–256.

[44] Himank Yadav, Zhengxiao Du, and Thorsten Joachims. 2021. Policy-gradient training of fair and unbiased ranking functions. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1044–1053.

[45] Ke Yang and Julia Stoyanovich. 2017. Measuring fairness in ranked outputs. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management* (SSDBM '17) Article 22. Association for Computing Machinery, Chicago, IL, USA, 6 pages.

[46] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. 2017. Fa*ir: a fair top-k ranking algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (CIKM '17). Association for Computing Machinery, Singapore, Singapore, 1569–1578.

[47] Meike Zehlike and Carlos Castillo. 2020. Reducing disparate exposure in ranking: a learning to rank approach. In *Proceedings of The Web Conference 2020* (WWW '20). Association for Computing Machinery, Taipei, Taiwan, 2849–2855.

[48] Meike Zehlike, Ke Yang, and Julia Stoyanovich. 2022. Fairness in ranking, part i: score-based ranking. *ACM Comput. Surv.*, 55, 6, Article 118, (Dec. 2022), 36 pages.

[49] Meike Zehlike, Ke Yang, and Julia Stoyanovich. 2022. Fairness in ranking, part ii: learning-to-rank and recommender systems. *ACM Comput. Surv.*, 55, 6, Article 117, (Dec. 2022), 41 pages.

A PROOFS

Proposition 4.1. *Let $\gamma = \min \overline{F}_{abs}(\hat{Y})$ be the minimum value of absolute fairness violation when optimized without the ranking utility. Then, irrespective of the structure of the PF, the optimal solution of WC scalarization* (13) *w.r.t. $\mathbf{r} = [\lambda, 1]$, where $\lambda \le \frac{1}{\gamma}$, guarantees that $\overline{F}^*_{abs} = \epsilon_\lambda \le \frac{1}{\lambda}$.*

Proof. The condition $\lambda \le \frac{1}{\gamma}$ ensures the existence of a solution with $r_1 \overline{F}^*_{abs} = r_2(1 - U^*) \implies \lambda \overline{U}^*_{abs} = (1 - \overline{U}^*)$. We use the normalized property of NDCG to avail $\overline{U}^* \le 1$, since it is the mean NDCG. Therefore, $1 - \overline{U}^* \le 1$, thus proving the proposition. □

Proposition 4.2. *The restriction on the optimal model offered by QWC is stricter than that of WC, i.e., $\Theta^{\mathbf{r}}_{\mathbb{Q}} \subset \Theta^{\mathbf{r}}$.*

Proof. Let $\theta \in \Theta^{\mathbf{r}}_{\mathbb{Q}}$, and $\hat{Y}^\theta = \{\hat{\mathbf{y}}^\theta_q\}$ be the scores when the model parameter is θ. Then $r_1 F_{abs}(\hat{\mathbf{y}}^\theta_q) = r_2 U(\hat{\mathbf{y}}^\theta_q)$ for all $q \in [N]$. Taking the mean across all queries on both LHS and RHS, we get

$$r_1 \frac{1}{N} \sum_{q=1}^{N} F_{abs}(\hat{\mathbf{y}}^\theta_q) = r_2 \frac{1}{N} \sum_{q=1}^{N} U(\hat{\mathbf{y}}^\theta_q) \tag{33}$$

$$\implies r_1 \overline{F}_{abs}(\hat{Y}^\theta) = r_2 \overline{U}(\hat{Y}^\theta) \tag{34}$$

Therefore, $\theta \in \Theta^{\mathbf{r}}$, thus proving the proposition. □

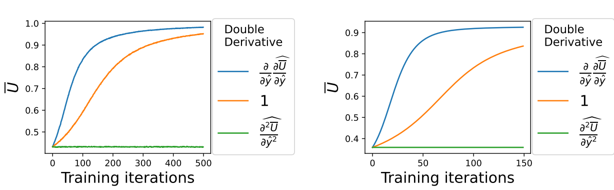
B DOUBLE DERIVATIVE

Alternative to (28) and (29), one may apply the log-derivative trick twice to obtain the true double derivatives similar to the gradients (24) and (26), and then approximate with a sample set. This would result in

$$\widehat{\frac{\partial^2 U}{\partial \hat{y}_i^2}} = \frac{1}{|\mathcal{S}|} \sum_{\sigma \in \mathcal{S}} \left(\left(\frac{\partial l\pi(\sigma|\hat{\mathbf{y}})}{\partial \hat{y}_i} \right)^2 + \frac{\partial^2 l\pi(\sigma|\hat{\mathbf{y}})}{\partial \hat{y}_i^2} \right) M(\mathbf{y}, \sigma), \tag{35}$$

$$\widehat{\frac{\partial^2 F_{abs}}{\partial \hat{y}_i^2}} = \frac{1}{|\mathcal{S}|} \sum_{\sigma \in \mathcal{S}} \left(\left(\frac{\partial l\pi(\sigma|\hat{\mathbf{y}})}{\partial \hat{y}_i} \right)^2 + \frac{\partial^2 l\pi(\sigma|\hat{\mathbf{y}})}{\partial \hat{y}_i^2} \right) \overline{D}(\mathbf{m}, \sigma), \tag{36}$$

where $l\pi$ is short for $\log(\pi)$ and $\overline{D} = \mathrm{sign}\left(\widehat{F(\hat{\mathbf{y}})}\right) D$. Also, to showcase acceleration in training, we also compare with optimization that does not use double derivatives, instead uses quasi-Newton method with unit weight on the second order term. In Figure 6, we show the training evolution when the model is trained only for the ranking task on German Credit and E-Commerce datasets. Clearly, the double derivatives of (28) accelerate the training, and the double derivatives of (35) do not improve the ranking utility during training.



(a) German Credit **(b) E-Commerce**

Figure 6: Training results of double derivative methods. 1 **stands for quasi-Newton with unit weight for 2nd order term.**

C WC-MGDA EXTENSION OF WC AND QWC

The WC-MGDA [27] leverages the Multi-Gradient Descent Algorithm [9] to find search directions according to the Weighted Chebychev scalarization. It solves a Second Order Cone Programming (SOCP) problem in every iteration to find the combination coefficients for the gradients w.r.t. a preference vector $\mathbf{r}$:

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^2_+, \gamma} \ \boldsymbol{\alpha}^T \left(\mathbf{r} \odot (\mathbf{c}^t - \mathbf{b}) \right) - u\gamma \tag{37a}$$

$$\text{s.t.} \quad \alpha_1 + \alpha_2 = 1 \tag{37b}$$

$$\|G_{\mathbf{r}} \boldsymbol{\alpha}\| \le \gamma, \tag{37c}$$

where $\mathbf{c}^t = [\overline{F}_{abs}(\theta^t), 1 - \overline{U}(\theta^t)]$ is the cost vector at t^{th} iteration, $\mathbf{b}$ is a reference point from which the search is carried out along the $\mathbf{r}^{-1}$ ray in the objective space, $G_{\mathbf{r}} = \mathrm{diag}(\sqrt{\mathbf{r}})\sqrt{G^T G}\mathrm{diag}(\sqrt{\mathbf{r}})$ where $G = [\nabla_\theta F_{abs}, -\nabla_\theta U]$ contains the gradients at t^{th} iteration. WC-MGDA jointly solves WC and MGDA to ensure achieving both preference alignment and Pareto Optimality. While the WC problem tries to find solutions by minimizing weighted ℓ_∞, the norm minimization ensures Pareto Optimality.

In the querywise setup, the SOCP problem in (37) is solved for every query. In particular, it is solved using the querywise cost $\mathbf{c}^t = [F_{abs}(\theta^t|q), 1 - U(\theta^t|q)]$, and gradients $G = [\nabla_\theta F_{abs}(\cdot|q), -\nabla_\theta U(\cdot|q)]$. Much like the Quadratic Programming (QP) based EPO Search, extending the SOCP based WC-MGDA to querywise setup is computationally expensive. This is mainly due to lack of software packages using which several QP/SOCP problems can be solved parallelly.

D TRAINING RESULTS

D.1 E-Commerce Dataset

Training results on E-Commerce dataset are shown in Figure 7a.

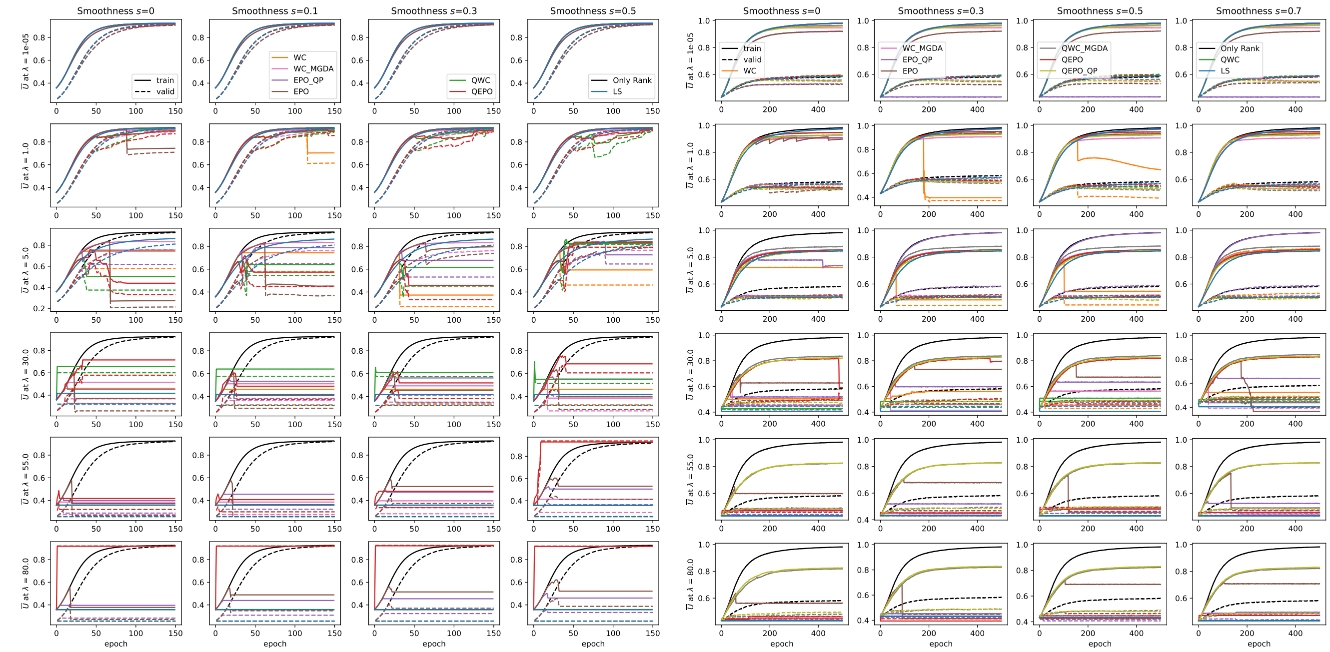
First, we notice the conflict between the two tasks faced by all the MOO methods. As the λ increases, the saturated ranking utility decreases, the gap widens from single task learning, i.e., Only Rank.

Second, we see that coefficient smoothness does improve the results for all the WC based methods. The smoothness is not applicable to LS since its coefficients remain static.

Lastly, we observe an interesting phenomenon at $\lambda = 80$. Our matrix inversion based querywise EPO search method outperforms the Only Rank method despite such a high λ. We attribute this to the inductive bias effect discussed in Section (4.4).

D.2 German Credit Dataset

The training results for German Credit datasets are shown in Figure 7b. The results are similar to that of E-Commerce datasets. As the fairness weight λ increases, the saturated ranking utility decreases, i.e., the gap between the single task learning based Only Rank and other MOO methods widens. The effect of smoothing is unclear. For $s = 0.3$ and 0.5, the performance of WC seems to be worse as compared to $s = 0.0$, but it improves for $s = 0.7$, e.g. at $\lambda = 1.$ and 5. For the matrix inversion based EPO, increasing smoothness seems to worsen the performance, except for $\lambda = 80$, where higher s has better ranking utility. An interesting observation is that, although the QWC and the matrix inversion based QEPO does not surpass



(a) E-Commerce dataset **(b) German Credit dataset**

Figure 7: Training results of different MOO methods for various settings of λ and smoothness factor s.

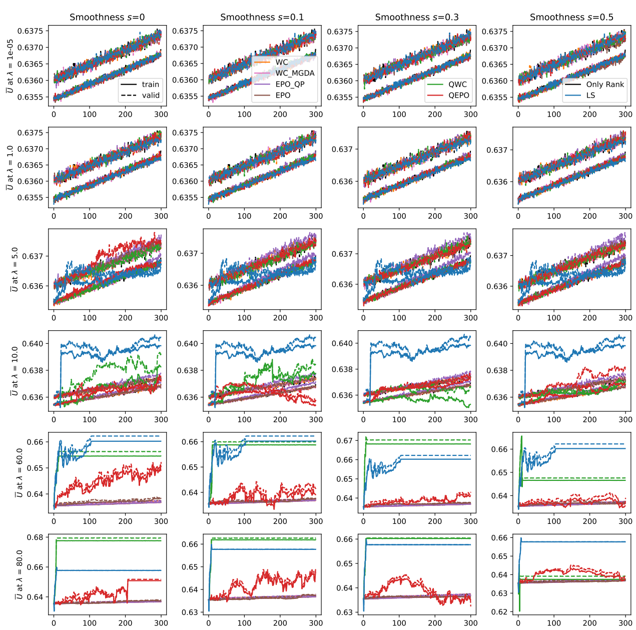


Figure 8: Training results of different MOO methods on MSLR-WEB30k dataset for various settings of λ and smoothness factor s.

the results of other MOO methods in the training datasets in Figure 7b, it does surpass significantly in the test dataset, as shown in Table 3 of the main paper. This can be attributed to the beneficial effects, such as regularization and inductive bias, of the querywise approach. The QP and SOCP based QEPO Search and QWC-MGDA have similar performance in the training dataset, but did not have better ranking utility in the test dataset.

D.3 MSLR-WEB30K Dataset:

The training results for MSLR-WEB30K datasets are shown in Figure 8.

For the MSLR-WEB30K dataset, the single task learning, i.e., Only Rank, is not able to improve the Plackett-Luce based ranking utility. This is reflected also in the MOO methods at lower values of fairness weights, i.e. $\lambda \leq 5$. We tested with different values of learning rates and tree parameters, such as maximum depth and number of iterations, to see if the training results improved. For all these experiments with lower value of λ, the pattern was similar to that in Figure 8. However, with increasing λ the improvement in ranking utility is clearly visible, especially in LS, QWC and the matrix inversion based EPO. At $\lambda = 80$, the behavior is similar to that in the E-Commerce dataset; there is a sharp increase in the ranking utility despite the high value of λ, and this increment is highest for the querywise WC method. This result of QWC is also reflected in the test dataset as reported in Table 3 of the main paper. The smoothing factor did not seem to improve the results as it did in the E-Commerce dataset.