5-2024

Fairness and Bias of Machine Learning in Search and Ranking

Yuan Wang

SANTA CLARA UNIVERSITY

Department of Computer Science & Engineering

Date: May 2024

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER

DR. YI FANG BY

Yuan Wang

ENTITLED

Fairness and Bias of Machine Learning in Search and Ranking

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE

OF

DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE & ENGINEERING

Dr. Yi Fang
Dr. Yi Fang (May 6, 2024 11:29 PDT)

Thesis Advisor
Dr. Yi Fang

Dr. David Anastasiu (May 3, 2024 16:31 PDT)

Thesis Reader
Dr. David Anastasiu

Silvia Figueira
Silvia Figueira (May 9, 2024 14:55 PDT)

Department Chair
Dr. Silvia Figueira

Sean Choi (May 3, 2024 16:29 PDT)

Thesis Reader
Dr. Sean Choi

Dr. Haibing Lu (May 3, 2024 16:20 PDT)

Thesis Reader
Dr. Haibing Lu

Dr. Zhiqiang Tao
Dr. Zhiqiang Tao (May 6, 2024 14:46 EDT)

Thesis Reader
Dr. Zhiqiang Tao

Fairness and Bias of Machine Learning in Search and Ranking

by

Yuan Wang

Dissertation

Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Computer Science & Engineering
in the School of Engineering at Santa Clara University, 2024

Santa Clara, California

Dedicated to my family...

Acknowledgements

First and foremost, I extend my deepest gratitude to my advisor, Professor Yi Fang, whose unwavering guidance and support have been instrumental to my doctoral journey. Professor Fang not only believed in my potential but also supported me with patience, kindness, and an unmatched dedication to excellence. His mentorship transcended academic instruction, offering personal support and invaluable life lessons that have shaped me both as a scholar and as an individual. He teaches me to keep trying, to be curious, and to work hard. I want to keep doing these things in my job too. Learning from him has been a really special chance for me, and I'm so thankful for it.

I would like to thank my doctoral committee consisting of Prof. Zhiqiang Tao, Prof. David Anastasiu, Prof. Sean Choi, and Prof. Haibing Lu for their time and suggestions to make my thesis better.

I would like to thank my lab mates Travis Ebesu, Xuyang Wu, Zhiyuan Peng, and Suthee Chaidaroon, who supported me through different parts of this journey.

Lastly, I would like to thank my family. My parents and my brother have given me endless love and support throughout this entire journey. I also want to thank my fiancée, Yijia, for her love and patience.

Fairness and Bias of Machine Learning in Search and Ranking

Yuan Wang
Department of Computer Science & Engineering
Santa Clara University
Santa Clara, California
2024

ABSTRACT

Recent advancements in Information Retrieval (IR) and machine learning have significantly improved ranking and search system performance. However, these data-driven approaches often suffer from inherent biases present in training datasets, leading to unfair treatment of certain demographic groups and contributing to systematic discrimination based on race, gender, or geographic location. This research aims to address the fairness and bias issue in ranking and search systems by proposing innovative frameworks that mitigate data bias and ensure equitable representation and exposure across diverse groups.

We introduce two novel frameworks: the Meta-learning based Fair Ranking (MFR) model and the Meta Curriculum-based Fair Ranking (MCFR) framework, both designed to alleviate dataset bias through automatically-weighted loss functions and curriculum learning strategies, respectively. These approaches utilize meta-learning to adjust ranking loss, focusing particularly on improving the fairness metrics for minority groups while maintaining competitive ranking performance. Additionally, we conduct an empirical evaluation of Large Language Models (LLMs) in text-ranking tasks, revealing

biases in handling queries and documents related to binary protected attributes. Our analysis offers a benchmark for assessing LLMs' fairness and highlights the necessity for equitable representation in search outcomes.

Furthermore, we explore the challenge of data selection bias in multi-stage recommendation systems, particularly in online advertising contexts like Pinterest's multi-cascade ads ranking system. Through comprehensive experiments, we assess various state-of-the-art methods, and our findings demonstrate the effectiveness of a modified version of unsupervised domain adaptation (MUDA) in mitigating selection bias.

Collectively, our work contributes to the development of fairer ranking and search systems. By addressing bias at its source and employing meta-learning and curriculum learning techniques, we pave the way for more equitable and transparent IR systems that serve diverse user bases without discrimination.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The quest for fairness in information retrieval (IR) systems is gaining unprecedented attention, as the digital era demands equity across all platforms and services. Central to this pursuit is the challenge of mitigating systematic biases within data-driven ranking models. These biases, often a reflection of historical discrimination, manifest as unfair treatment towards underrepresented groups, leading to disparate exposure and unequal opportunities in various real-world applications such as expert search and job recommendations. The essence of fairness in IR extends beyond mere algorithmic adjustments; it is about ensuring that all demographic groups have equal visibility and representation in the outcomes of search and recommendation systems.

To address the inherent biases in datasets used for training machine learning models, we developed a few novel frameworks. These frameworks, including Meta-learning based Fair Ranking (MFR) and Meta Curriculum-based Fair Ranking (MCFR), represent significant strides towards achieving equitable treatment across protected attributes. By re-weighting ranking losses and incorporating curriculum learning into meta dataset construction, these models aim to balance exposure between advantaged and disadvantaged groups, offering a more nuanced approach to fairness that transcends traditional mitigation strategies.

The integration of Large Language Models (LLMs) in ranking tasks further complicates the landscape of fairness in IR. Despite their superior performance in understanding and processing natural language, LLMs are not immune to fairness concerns. The empirical scrutiny of these models against fairness benchmarks reveals a pressing need to evaluate and fine-tune them with a focus on equity, ensuring that their deployment does not perpetuate existing biases.

Lastly, the realm of online advertising, particularly in multi-stage recommendation systems like those used in ad retrieval, underscores the pervasive challenge of selection bias. While this area might seem tangential, it shares the core issue of bias mitigation with broader IR systems. Efficiently managing the diversity and quality of ads in the upper funnel stages without succumbing to biases is crucial for maintaining the integrity and fairness of digital advertising ecosystems.

In summary, the motivation for this thesis stems from the urgent need to address and

rectify fairness issues in IR systems. Through a comprehensive exploration of innovative frameworks, meticulous evaluation of LLMs, and consideration of selection bias in online advertising, this work aims to contribute meaningful solutions to the overarching challenge of ensuring fairness in the digital information landscape.

1.2 Overview

This thesis presents a comprehensive exploration of fairness in ranking and search systems, addressing the multifaceted challenge of bias in information retrieval (IR) through a series of innovative approaches and methodologies. Across four distinct but interconnected studies, we delve into the complexities of data bias, selection bias, and the ethical implications of Large Language Models (LLMs) in text ranking, providing a holistic examination of fairness in the digital information landscape.

We firstly introduces the Meta-learning based Fair Ranking (MFR) model, an advanced framework designed to mitigate data bias by re-weighting ranking losses through a bilevel optimization process. This model not only enhances fairness metrics but also maintains competitive ranking performance, offering a scalable solution for equitable IR systems.

Building on this foundation, we proposes the Meta Curriculum-based Fair Ranking (MCFR) framework, which further addresses data bias by integrating in-processing and pre-processing techniques with curriculum learning. MCFR demonstrates remarkable

versatility and effectiveness in improving fairness metrics across various ranking loss functions, showcasing its potential as a generic framework for fair ranking.

We then focus the evaluation of fairness in LLMs for text ranking, establishing a benchmark that incorporates listwise and pairwise evaluation methods focused on binary protected attributes. Through extensive experimentation, we reveal inherent fairness issues in LLMs and propose a fine-tuning strategy using Low-Rank Adaptation (LoRA) to mitigate these issues, marking a significant step towards more equitable LLM-based ranking systems.

Finally, we tackles the selection bias in multi-cascade advertisement recommendation systems, surveying state-of-the-art modeling strategies and introducing a Modified Unsupervised Domain Adaptation (MUDA) approach. MUDA outperforms both contemporary models and the existing production model in online settings, highlighting its effectiveness in addressing selection bias and enhancing the fairness and efficiency of recommendation systems.

1.3 Contributions

The contribution of thesis can be summarized as follows:

- We introduce the Meta-learning based Fair Ranking (MFR) model, a novel approach that addresses data bias in ranking systems by automatically adjusting ranking losses. The MFR model, framed as a bilevel optimization problem and

solved through an innovative gradients-through-gradients technique, demonstrates its robustness and effectiveness in real-world datasets. Our results highlight MFR's capacity to achieve competitive ranking performance while significantly enhancing fairness metrics, marking a critical advancement in the pursuit of fair information retrieval systems.

- We present the Meta Curriculum-based Fair Ranking (MCFR) framework, an innovative approach that mitigates data bias by blending in-processing and pre-processing techniques with curriculum learning. MCFR, formulated as a bilevel optimization problem solved via gradients-through-gradients, proves versatile across various ranking loss functions and fairness metrics. Our empirical studies across public datasets affirm MCFR's effectiveness in matching existing ranking performances while significantly advancing fairness metrics. Notably, MCFR enhances fairness more efficiently, requiring less data and achieving fast convergence, positioning it as a highly adaptable and impactful framework in promoting fairness in ranking systems.

- We create a benchmark for evaluating the fairness of Large Language Models (LLMs) in text ranking, focusing on binary protected attributes through listwise and pairwise methods. Our extensive experiments on real-world datasets reveal fairness issues in LLMs, prompting us to propose a fine-tuning strategy using Low-Rank Adaptation (LoRA) specifically designed to address these concerns. This dual approach of identifying and mitigating fairness problems marks a significant advancement in improving LLMs' performance in ranking tasks.

- We address the selection bias in advertisement recommendation systems by characterizing the issue and evaluating various modeling strategies. Our exploration leads to the development of a Modified Unsupervised Domain Adaptation (MUDA) approach, which stands out for its superior performance in online settings, outperforming both contemporary models and the existing production model. This study advances the mitigation of selection bias, showcasing MUDA's effectiveness in enhancing recommendation fairness and efficiency.

1.4 Outline

This thesis is structured as follows, Chapter 2 reviews the existing literature on fairness in information retrieval, highlighting the significance of addressing biases in ranking models and the evolving strategies to mitigate these challenges. Chapter 3 details the Meta-learning based Fair Ranking (MFR) model, focusing on its innovative approach to enhance fairness by adjusting training losses for improved minority group exposure and its validation through real-world datasets. Chapter 4 discusses the Meta Curriculum-based Fair Ranking (MCFR) framework, which integrates meta-learning with curriculum learning to counteract data bias and showcases its effectiveness over traditional fairness models. Chapter 5 explores the fairness of Large Language Models in ranking tasks, presenting an empirical study on biases and introducing a mitigation strategy via LoRA fine-tuning to promote equitable outcomes. Chapter 6 investigates selection bias in Pinterest's advertising system and proposes the Modified Unsupervised Domain

Adaptation (MUDA) model, demonstrating its capacity to improve recommendation performance and advertising efficiency. Chapter 7 concludes the thesis by summarizing the key contributions, reflecting on the impact of this work on fairness in search and ranking, and suggesting future research directions to further advance the field.

Chapter 2

Related Work

2.1 Fairness on Ranking

Zehlike et al.[92] categorized fair ranking models into score-based and supervised learning models. Score-based models modify score outcomes or distributions for enhanced fairness. Notable contributions include works by Yang et al.[86, 87], Celis et al.[18], Stoyanovich et al.[72], Kleinberg et al. [47], and Asudeh et al.[5].

Supervised fairness models in ranking span pre-processing, in-processing, and post-processing approaches. Pre-processing models, exemplified by Lahoti et al.[49], work on deriving fair training data. In-processing models, such as Zehlike et al.'s DELTR[89], address fairness during training, focusing on exposure bias. Similarly, Beutel et al.[9] introduced a pairwise ranking loss function with fairness regularizer, while Ma et al.[52]

tackled fairness in query generation. Haak et al.[39] aimed at search query bias identification, and Chu et al.[23] highlighted biases in neural architecture search evaluations. Importantly, Chen et al. [20] proposed a meta-learning-based debiasing framework for recommendations. Post-processing models, conversely, refine model outputs post-training for fairness. Among these, Zehlike et al.'s works [90, 91] like FA*IR ensure representation of protected groups and offer continuous fairness interpolation. Additionally, Biega et al. [10] developed an algorithm optimizing the equity of user attention through relevance loss function.

2.2 Meta-Learning on Fairness

Meta-learning is a field of study that aims to improve the learning ability of models by adapting to new tasks or environments, and it could be divided into two main categories: model-based [30, 3] and learning algorithm-based [4]. In addition to tasks such as few-shot learning [43], continual learning [60], and hyperparameter optimization [32], fairness is an important field.

Zhao et al.[98] presented the Follow the Fair Meta Leader (FFML) that learns an online fair classification model's primal, delivering both accuracy and fairness. In a subsequent work, Zhao et al.[97] emphasized the Primal-Dual Fair Meta-learning, targeting the optimal initialization of the base model's weights to rapidly adjust to new fairness tasks. They further advanced their research in [96], creating a few-shot discrimination

prevention model for unbiased multi-class classification, rooted in the MAML framework. Concurrently, Slack et al.[71] introduced Fair-MAML, designed to derive fair models from minimal data for emerging tasks. This model, like Zhao's, is built upon the MAML framework but incorporates fairness regularization and a specific fairness hyperparameter. On recommender systems, Chen et al.[20] applied meta-learning principles on the AutoDebias framework.

2.3 Fairness in LLMs

Research on fairness in LLMs has gained considerable traction, driven by the realization that biases present in pretraining corpora can lead LLMs to generate content that is not only harmful but also offensive, often resulting in discrimination against marginalized groups. This heightened awareness has spurred increased research efforts aimed at understanding the origins of bias and addressing the detrimental aspects of LLMs [68, 13]. Initiatives like Reinforcement Learning from Human Feedback [58] and Reinforcement Learning for AI Fairness [6] seek to mitigate the reinforcement of existing stereotypes and the generation of demeaning content.

Beyond existing literature, FaiRLLM [95] critically evaluates RecLLM's fairness, highlighting biases in ChatGPT recommendations by user attributes. Concurrently, efforts to refine LLM fairness assessments are gaining traction within the NLP community [22, 66]. Studies like [12] and [1] expose biases in GPT-3's content generation, with the

latter noting a violent bias against Muslims. Benchmarks such as BBQ [61], CrowS-Pairs [54], RealToxicityPrompts [34], and holistic evaluations [50] further this analysis across various LLMs. DecodingTrust [77] extends this to a detailed fairness exploration in ChatGPT and GPT-4.

2.4 Selection Bias

Research on selection bias in recommendation systems is increasing, exploring methods to reduce bias and enhance system performance. One of the approaches is through re-sampling techniques. This includes methods such as undersampling [63, 42] and SMOTE (Synthetic Minority Over-sampling Technique) [19, 14] which aims to balance out the distribution of data across different classes. Another popular approach is the use of cost-sensitive learning methods, which assign different costs to different types of errors in order to balance the trade-off between different types of bias. For example, the method of adversarial learning [94, 24] aims to minimize bias by adding an adversarial term to the loss function that encourages the model to produce fair predictions. Another area of research focuses on the use of debiasing techniques in the representation learning process, such as Fair Representation Learning [93] which learns representations that are invariant to certain sensitive attributes. There are also other recent studies that address selection bias by using counterfactual data augmentation (CFDA) [81], which creates new, hypothetical data points to increase the diversity of the training set. This can be done by generating synthetic data points that are similar to the original data points,

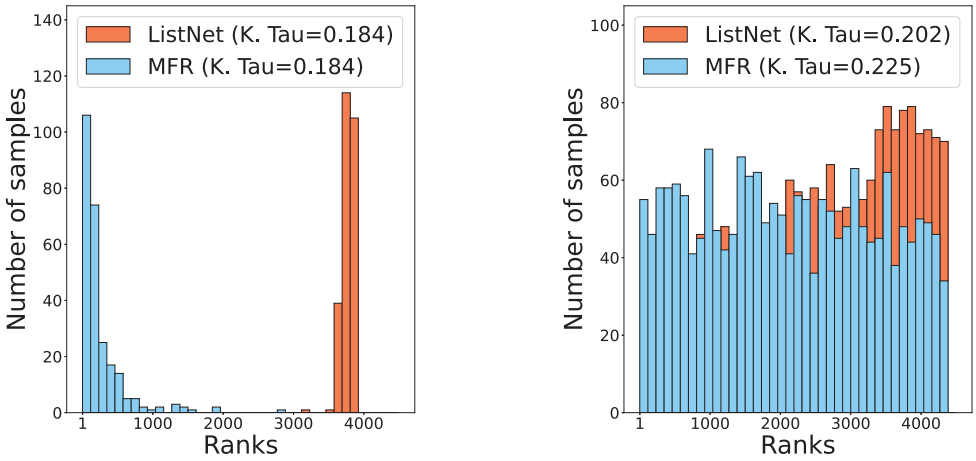
but with different sensitive attributes. In addition, meta-learning [21, 78] have been applied to debiasing recommendation systems. For multi-stage cascade systems, Qin et al. [64] proposed the RankFlow to solve the selection bias in the joint-training system, but it could be expensive to deploy in the production system. Our work aims to solve the selection bias issue for independent-training models in the cascade system.

Chapter 3

A Meta-learning Approach to Fair Ranking

3.1 Introduction

Recently, the fairness in information retrieval (IR) system has attracted more and more attention [92, 86, 87]. The ranking models aim to give the relevant scores for the items under the query, and the top items with the highest scores will be delivered to the users. These ranking models are generally data-driven, which means the models will observe particular patterns in the training dataset and make predictions based on them. However, when the subject of the ranking problem is about the expert search or the job recommendation, the *systematic biases* from the dataset – usually stemming from a biased data distribution – will introduce *unfairness* in the trained model. For example,



(A) Law Students (race)

(B) Law Students (gender)

FIGURE 3.1: Illustration of the predicted rankings distribution of the protected groups (*female students, African American students*) on the two different datasets. We report Kendall's Tau as the ranking metric. The proposed MFR model ranks the items from the protected groups higher compared to ListNet [17], which indicates that the MFR improves the protected attribute's exposure with unbiased ranking performance.

the traditional LTR model such as ListNet [17] will "discriminately" assign lower weights to the minority group due to the data bias (see Fig. 3.1). As addressed by Friedman [33], the historic discrimination to the socially underrepresented group in the dataset will make its way into the model as the pattern will be observed during the training process. The unfairness problem could be summarized as the disparate exposure [89] as the disadvantaged protected group is not treated as equally as the advantaged group in the dataset. This disparate exposure could lead to a negative impact on many real-world ranking problems, such as the unequal opportunity in the job market for the underrepresented group.

To solve the unfairness problem, tremendous research efforts have been made in designing fairness-aware algorithms, among which, the fairness ranking models can be categorized as the score-based and supervised ones. For score-based models, there are

the Rank-aware proportional representation [86], the Constrained ranking maximization [18], etc. Some score-based models aim to correct the bias in the training data, and the others aim to adjust the prediction scores for better fairness. There are also supervised models, such as DELTR [89], FA*IR [90], etc, which could learn a fair model from the biased dataset. In general, the ranking models focus on different mitigation points such as the post-, in-, and pre-processing of the model training. Although the in-processing models have achieved good performance on the fairness metric, there is still the limitation as the model is learned from the biased dataset. Thus, the meta-learning could benefit the aforementioned problem by training a meta-learner on a meta-dataset. The meta-dataset is collected uniformly without any bias, which would train a fair meta-learner so that the ranking model could learn from it. For general fairness problems such as training the classification model on a biased dataset, researchers have applied the Model-Agnostic Meta-Learning (MAML) [31]. For example, the Meta-Weight-Net [69] proposed to explicitly learn a weighting function from the meta-dataset which is updated simultaneously with the classifier. However, meta-learning is still under-explored for the fairness-aware ranking problems.

In this study, we propose a meta-learning framework to formulate the fairness-aware ranking task as a bilevel optimization problem, where the upper-level is the meta-trainer and the lower-level is the ranking model. That is, we can train a meta-learner on the meta-dataset which could help the ranking model to learn fairly on the biased dataset. The meta-dataset is a small unbiased dataset, which is collected by uniformly sampling from the training dataset under all queries for both the protected group and

the unprotected group. In detail, at each training iteration, we use the ranking model and the ranking loss function to compute the loss values for each data sample from the training dataset. Then we train a multi-layer neural network as the weighting function to re-weight the loss values, and the weighting function is optimized by the weighted loss values on the meta-dataset. Since the weighting function which is the meta-learner is subject to the ranking models, our goal is to optimize the loss' weights (given by the meta-learner) to achieve fairness on the training dataset. Intuitively, we can see the loss' weight as the hyperparameter which could be learned, and we train a meta-learner to tune the hyperparameter on the meta-dataset. Such the training process could also be referred to as the bilevel optimization as the learned parameters of the ranking model depend on the parameters of the meta-learner. To the best of our knowledge, we propose the first meta-learning approach to fair ranking. In summary, this work makes the following contributions:

- We propose a general meta-learning framework for the fairness ranking called Meta-learning based Fair Ranking (MFR) that addresses the data bias by automatically re-weighting the ranking losses.
- We formulate the MFR as a bilevel optimization problem and solve it using gradients through gradients.
- Experiments on the real-world datasets demonstrate that the proposed method achieves a comparable ranking performance and significantly improves the fairness metric compared with state-of-the-art methods.

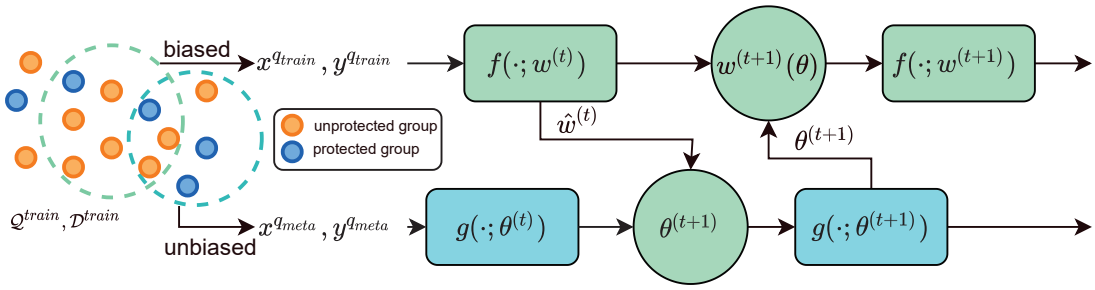


FIGURE 3.2: MFR learning algorithm flowchart (steps 4 and 6 in Algorithm 1). Note that $f(\cdot; w)$ is the ranking model, $g(\cdot; \theta)$ is the meta-learner, b is the batch size for the training dataset, d is the batch size for the meta-dataset, and α and β are the learning rates. At each iteration, we firstly update θ in the meta-learner using Eq. (8) with the meta-dataset, and then we update w in the ranking model using Eq. (9) with the training dataset.

3.2 Meta-learning Based Fair Ranking

We aim to train a fairness-aware ranking model that could achieve good performance on both utility and fairness metrics. To do that, we tune the ranking model's loss weights values to make the model emphasize more on the protected group than the unprotected one during the ranking inference. Instead of using the fixed weights, we utilize a meta-dataset which is sampled from the original training dataset with an unbiased distribution and smaller size to train a meta-learner as a weighting function. The meta-learner could guide the ranking model to learn fairly.

Given the training dataset with a set of queries $\mathcal{Q}^{train}$ with $|\mathcal{Q}^{train}|= m$ and a set of items $\mathcal{D}^{train}$ with $|\mathcal{D}^{train}|= n$. Each query q from $\mathcal{Q}^{train}$ is associated with a list of item candidates $d^{(q)}$ from $\mathcal{D}^{train}$, and each item is represented as a feature vector $x_i^{(q)}$. For each query q, the feature vector $x^{(q)}$ is associated with the relevance score $y^{(q)}$. Let $f(x^{(q)}; w)$ be the ranking model and w represent all the learnable parameters in f. Then

the output of the ranking model could be denoted as $\hat{y}^{(q)} = f(x^{(q)}; w)$. Generally, we learn the optimized parameters w^* by $\min_w \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(y_i^{(q)}, \hat{y}_i^{(q)})$ and $\mathcal{L}$ could be used as any ranking loss functions. However, equally treating $\mathcal{L}$ to each sample could lead the ranking model f unfair to minority groups since the heavy data bias issue in the training dataset. To address this challenge, we introduce a meta-learner $g(\cdot;\theta)$, parameterized by θ, to adaptively tune loss weights for each sample to achieve a fair exposure over diversity. Thus, we rewrite the training loss as the following:

$$\mathcal{L}^{train}(w;\theta) = \frac{1}{m}\sum_{i=1}^{m} \phi_i \mathcal{L}_i(w) = \frac{1}{m}\sum_{i=1}^{m} \prec_i \mathcal{L}(y_i^{(q)}, \hat{y}_i^{(q)}), \tag{3.1}$$

where $\hat{y}_i^{(q)} = f(x_i^{(q)}; w)$ represents the model output, and $\phi_i \in [0, 1]$ represents the i-th sample's loss weight given by the proposed meta-learner $g(\cdot;\theta)$. Notably, $\mathcal{L}^{train}(w;\theta)$ governed by the meta-learner's output weights is conditioning on a fixed θ and used for updating the ranking model's parameter w. For convenience, we denote $\mathcal{L}_i(w)$ as the original loss value of the i-th training data sample output from the ranking loss $\mathcal{L}$. Following [69], we develop the meta-learner g as a multi-layer neural network, which takes as input a loss value, and instantiate g as

$$\phi_i = g(\mathcal{L}_i(w);\theta) = g\Big(\mathcal{L}_i(y^{(q)}, f(x^{(q)}; w));\theta\Big), \tag{3.2}$$

where i could be a sample from either the training dataset or the meta-dataset. We set the last-layer's activation function in g as a `sigmoid` so that the range of the output

Algorithm 1: The MFR Learning Algorithm

Input: Training dataset $\mathcal{Q}^{train}$,$\mathcal{D}^{train}$, meta-dataset $\mathcal{Q}^{meta}$,$\mathcal{D}^{meta}$, batch size b, d, max iterations T.

Output: Classifier network parameter $w^{(T)}$

1: Initialize ranking model's parameter $w^{(0)}$ and the meta-learner's parameter $\theta^{(0)}$.
2: **for** $t = 0$ **to** $T - 1$ **do**
3: $\{x^{q_{meta}}, y^{q_{meta}}\} \leftarrow$ SampleMiniBatch($\mathcal{Q}^{meta}, \mathcal{D}^{meta}, d$).
4: $\{x^{q_{train}}, y^{q_{train}}\} \leftarrow$ SampleMiniBatch($\mathcal{Q}^{train}, \mathcal{D}^{train}, b$).
5: Update $\hat{w^{(t)}}(\theta)$ by Eq. (3.4) with $\{x^{q_{train}}, y^{q_{train}}\}$.
6: Update $\theta^{(t+1)}$ by Eq. (3.9) with $\{x^{q_{meta}}, y^{q_{meta}}\}$.
7: Update $w^{(t+1)}$ by Eq. (3.10) with $\{x^{q_{train}}, y^{q_{train}}\}$.
8: **end for**

lies between 0 and 1. Eventually, we define a meta training loss function as

$$\mathcal{L}^{meta}(w(\theta)) = \frac{1}{s}\sum_{i=1}^{s}\mathcal{L}_i(w(\theta)). \tag{3.3}$$

Here we update the parameters of the ranking network by doing the gradient decent on a batch of a training data with the loss function in Eq. (3.1), and we can define $w(\theta)$ as:

$$\hat{w}^{(t)}(\theta) = w^{(t)} - \alpha\frac{1}{b}\sum_{i=1}^{b} g(\mathcal{L}_i^{train}(w^{(t)});\theta)\nabla_w\mathcal{L}_i^{train}(w) \tag{3.4}$$

To train the meta-learner, we need to sample a small meta-dataset with $\mathcal{Q}^{meta}$ and $\mathcal{D}^{meta}$. The meta-dataset represents the meta-knowledge of the true distribution of the protected group and the other group, where $|\mathcal{Q}^{meta}| = s \ll m$ and $|\mathcal{D}^{meta}| = r \ll n$. In the meta-dataset, we denote the feature vector of each item as $x^{(q_{meta})}$ and the relevance score as $y^{(q_{meta})}$ given a query q_{meta} from $\mathcal{Q}^{meta}$. Similar to $\mathcal{L}_i^{train}(w)$, we denote $\mathcal{L}_i^{meta}(w(\theta))$ as the loss value for each meta-dataset sample. The goal of the meta-learner $g(\cdot;\theta)$ is to leverage the unbiased meta-dataset to learn how to re-weight the loss values to train the

model $f(\cdot; w)$ on the biased dataset. Since w is a function of θ, we naturally formulate the proposed MFR as a bilevel optimization problem and give the objective function as

$$\begin{aligned} &\min_{\theta} \mathcal{L}^{meta}(w^*(\theta)) \\ &\text{s.t. } w^*(\theta) = \arg\min_{w} \mathcal{L}^{train}(w; \theta). \end{aligned} \tag{3.5}$$

Loss Functions. The proposed MFR jointly considers utility and fairness metrics by developing a listwise ranking loss with an exposure term following the DELTR loss [89], given by

$$\mathcal{L}(y^{(q)}, \hat{y}^{(q)}) = \ell(y^{(q)}, \hat{y}^{(q)}) + \gamma U(\hat{y}^{(q)}), \tag{3.6}$$

where $U(\hat{y}^{(q)})$ is a listwise fairness measurement, $\ell(y^{(q)}, \hat{y}^{(q)})$ is a listwise loss based on Cross Entropy [17], and $\gamma > 0$ is a balancing parameter. To obtain optimal parameters w^* and θ^*, we minimize the training loss by

$$w^*(\theta) = \arg\min_{w} \mathcal{L}^{train}(w; \theta) = \frac{1}{m} \sum_{i=1}^{m} \phi_i \mathcal{L}_i^{train}(w), \tag{3.7}$$

and the loss for the meta-learner by

$$\theta^* = \arg\min_{\theta} \mathcal{L}^{meta}(w^*(\theta)) = \frac{1}{s} \sum_{i=1}^{s} \mathcal{L}_i^{meta}(w^*(\theta)). \tag{3.8}$$

Parameters Update. At each step t, we compute the weighted loss values with θ^t and w^t, and update θ with the loss of the ranking model on the meta-dataset as the

following:

$$\theta^{(t+1)} = \theta^{(t)} - \beta \frac{1}{d} \sum_{i=1}^{d} \nabla_{\theta} \mathcal{L}_i^{meta}(w^{(t)}(\theta)), \tag{3.9}$$

where β is the learning rate, d is the batch size of the meta-dataset. After we have the $\theta^{(t+1)}$, we update w as the following:

$$w^{(t+1)}(\theta) = w^{(t)} - \alpha \frac{1}{b} \sum_{i=1}^{b} \phi_i \nabla_w \mathcal{L}_i^{train}(w), \tag{3.10}$$

where α is the learning rate and b is the batch size of the training dataset. We adopt an alternating optimization strategy [69, 75, 88] to implement Eq. (3.9) and Eq. (3.10) instead of using nested optimization loops. The whole training process is summarized in Algorithm 1.

Although we consider the DELTR loss as the objective function of the ranking model, we could also use other fair ranking losses here. Besides the disparate exposure, there are other biases in the common ranking dataset such as selection bias and position bias. The model aims to provide a general meta-learning framework that can handle any fair ranking problems.

3.3 Experiments

In the experiments, we train and evaluate the model on the three real-world datasets used in DELTR [89]. We study both the ranking and fairness metrics of our approach compared to other baseline models. The baseline models include the following: (i)

ListNet [17]; (ii) Lambdamart [16]; (iii) the DELTR model with γ_{small} and γ_{large} which is the same setting as in [89]; (iv) the FA*IR [90] pre-processing approach that creates the fair dataset and trains on it; (v) the FA*IR post-processing approach that reorders the prediction results to ensure the fairness; (vi) MFR with different γ on a different dataset; (vii) MFR with the ListNet loss (MFR-ListNet). The code is available at https://github.com/ywang4/A-Meta-learning-Approach-to-Fair-Ranking.

For a fair comparison, we follow the same settings[1] as described in DELTR [89] to split the dataset and generate the item features. We use the following datasets: (i) W3C Experts (gender); (ii) Engineering Students (high school); (iii) Engineering Students (Gender); (iv) Law Students (gender); (v) Law Students (race). In the W3C Experts dataset, the task is the expert search originated from TREC 2005 Enterprise Track [26]. The protected attribute is female, and there are 200 items per query with an average of 21.5 items from the protected group. In the Engineering Students dataset, the task is the academic performance prediction, and the dataset contains anonymized historical information of college students. For the high school dataset, the protected attribute is public high school, and there are 480.6 items per query with 167.6 items from the protected group on average. For the gender dataset, the protected attribute is female, and there are 480.6 items per query with 97.6 items from the protected group on average. In the Law Students dataset, the task is also the academic performance prediction. For the gender dataset, the protected attribute is female, and there is a total of 21791 items with 9537 items from the protected group. For the race dataset, the protected

[1] https://github.com/MilkaLichtblau/DELTR-Experiments

attribute is black, and there is a total of 19567 items with 1282 from the protected group. The queries are technical topics for the W3C dataset and academic years for the other datasets. For a fair comparison, we adapt the same evaluation metrics as [89]. To split the datasets, we have 50 queries for training and 10 queries for testing in the W3C dataset, 4 queries for training and 1 query for testing in the Engineering Students dataset, and 80% for training and 20% for testing in the Law Students dataset. We use Precision@10 (P@10) for the W3C dataset and Kendall's Tau for other datasets to evaluate the ranking performance. To measure fairness, we compute the exposure ratio between the protected and the non-protected group. Thus, in the fairness metric, values greater than 1.0 indicate greater visibility for the protected group and vice versa. As described in Sec. 3.2, the meta-dataset is required for our approach. Since the protected attribute in all datasets is binary, we perform random uniform sampling to collect the meta-dataset. Specifically, we randomly sample the same amount of data for the items from each query for each protected group and non-protected group.

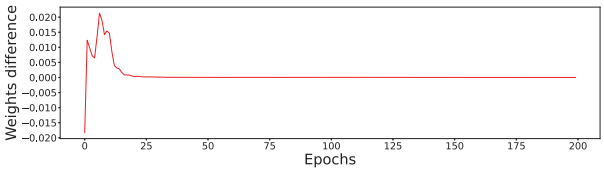
Settings. In general, for the weighting function, we set the update frequency of the parameter $\theta$to be per 2 steps, the optimizer to be SGD, the momentum to be 0.98, the learning rate to be 0.02, the hidden layer dimension to be 30, and the number of hidden layers to be 3. For the ranking model, we set the learning rate for all datasets to be 0.005 except for W3C data to be 0.0005, the optimizer to be SGD, the momentum to be 0.95, and the weight decay to be 0.005. The values of γ and training epoch vary for different datasets: W3C dataset uses $\gamma = 500$ and 100 epochs, Engineering Students (high school) uses $\gamma = 5000$ and 500 epochs, Engineering Students (Gender) uses $\gamma =$

	W3C Experts (gender)		Engineering Students (high school type)		Engineering Students (gender)		Law Students (gender)		Law Students (race)	
	P@10	**Fairness**	**K. Tau**	**Fairness**	**K. Tau**	**Fairness**	**K. Tau**	**Fairness**	**K. Tau**	**Fairness**
ListNet [17]	0.178	0.759	0.390	1.070	0.384	0.858	0.202	0.931	**0.184**	0.853
Lambdamart [16]	0.095	0.738	0.355	1.002	0.326	0.907	0.199	0.979	0.156	0.847
DELTR γ_{small} [89]	0.178	0.785	0.390	1.075	0.384	0.860	0.201	0.958	0.173	0.874
DELTR γ_{large} [89]	**0.180**	0.827	**0.391**	1.075	0.370	0.976	0.188	0.993	0.130	1.014
FA*IR post [90]	0.178	0.824	0.390	1.070	0.384	0.886	0.182	0.965	0.140	0.944
FA*IR pre [90]	**0.180**	0.770	0.374	1.020	0.360	0.942	0.203	0.931	0.161	0.895
MFR-ListNet	0.115	0.775	0.385	0.990	**0.385**	0.855	**0.225**	0.901	0.182	0.848
MFR	0.126	**0.830**	**0.391**	**1.086**	0.352	**1.052**	**0.225**	**1.015**	**0.184**	**1.654**

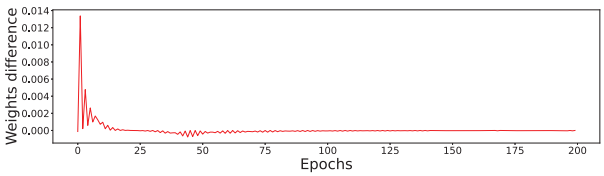
TABLE 3.1: Experimental results. To measure fairness, we compute the exposure ratio between the protected and the non-protected group, so the values greater than 1.0 indicate greater visibility for the protected group and vice versa. For the ranking metric, higher Kendall's Tau / Precision@10(P@10) scores indicate better performance. The bold text indicates the model with the best performance, and the results show that the MFR model is better on the fairness metrics with comparable performance on the ranking metrics against other state-of-the-art models.

500 and 100 epochs, Law Students(gender) uses $\gamma = 1200$ and 3000 epochs, and Law Students (race) uses $\gamma = 50000$ and 100 epochs.

Results Analysis. As shown in Tab. 3.1, our approach performs better in terms of the fairness metrics on all datasets than both the DELTR γ_{small} and DELTR γ_{large}. The DELTR γ_{small} and DELTR γ_{large} models use different scales of γ values to weight the exposure measure in the loss function. With the meta learner, we can achieve higher fairness metrics by re-weighting the loss distribution during the training process. The intuition behind the observation is that the imbalanced pattern among the training data is observed and corrected by the meta learner. For the ranking metrics, we have similar or better results on all datasets except the W3C dataset. Since ListNet and Lambdamart do not consider any fairness measure during the training, the results are as expected that the fairness metrics are worse than the fairness ranking models. In addition, we train the MFR-ListNet that has the standard listwise ranking loss in the framework. The evaluation results show the worse performance on both the ranking



(A) Engineering Students (high school)



(B) Law Students (gender)

FIGURE 3.3: The plot of the variation of learned weight over the two training datasets. The weight difference is computed as $\phi^t_{\text{diff}} = \frac{1}{m}\sum_{i=1}^{m} \phi^t_i - \phi^{t-1}_i$, and we plot the ϕ^t_{diff} over the training epochs. As shown in the plot, the weighting function is converging as the different values of weights between each epoch are decreasing to 0.0.

and fairness metrics. As listwise loss does not consider the exposure measure, the meta-dataset that has a different data distribution as the training dataset has a negative effect on the meta-learner during the re-weighting process. Thus, we conclude that the meta-learning approach could help the model to further improve the fairness metrics compare to the model with only the DELTR loss function.

In Fig. 3.1, we plot the histogram of ranks on the protected attributes from the different models. From the plot, we can see the distribution of the predicted ranks shifts from right to left, which indicates the MFR model generally ranks the items from the protected group higher compared to ListNet. Note that at the plot, 1 means the top rank, so when more data samples fall in the bins at the left, the items receive higher ranks. The plot also agrees with the evaluation results. As we see that there is a large difference in Fig. 3.1b, the fairness metric of MFR on Law Students (race) dataset is about two times than that of ListNet.

In Fig. 3.3, we plot the variation of the learned weight for the training data. The plots show that the weighting function is converging as the different values of weights between each epoch are decreasing to 0. As suggested in Meta-Weight-Net [69], we use the multi-layer neural network as the weighting function because the multi-layer neural network is known as a universal approximator for the most continuous functions. The convergence shown in the plots indicates the successful learning process on the weighting function.

3.4 Conclusion

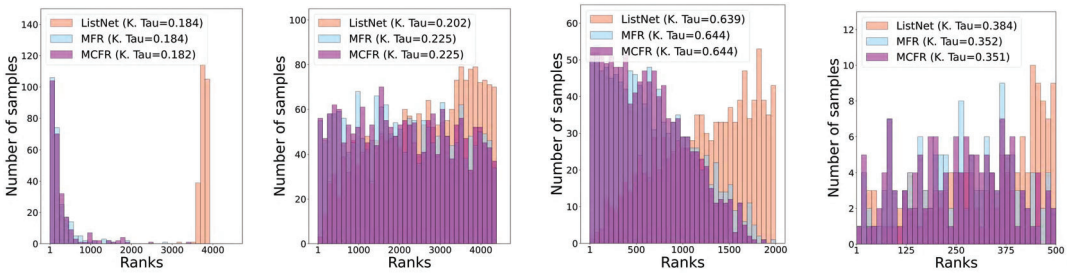
In this work, we have proposed a Meta-learning based Fair Ranking (MFR) model to improve the minority group's exposure. Our experiments on the real-world datasets demonstrate that our approach could achieve better fairness metrics compared to the fair ranking model without the meta-learning part.

Chapter 4

A Unified Meta-learning Framework for Fair Ranking with Curriculum Learning

4.1 Introduction

Fairness in search engines is an important topic, which focuses on training an unbiased ranking model towards protected attributes. Typically, when a user query is given, the ranking model predicts relevant scores among candidate items and returns items with the highest scores to users. The data-driven ranking model is usually trained with large datasets, and thus the ranker will learn user/item patterns from the training dataset and make predictions based on them. However, in many cases, the systematic biases



(A) African American (B) Female (C) African American (D) Female

FIGURE 4.1: Illustration of the predicted rankings distribution of two protected attributes on four datasets – *(a) Law Student (gender) [82], (b) Law Student (race) [82], (c) COMPAS [7], and (d) Engineering Student [89].* We report Kendall's Tau [48] as the ranking performance. MCFR and MFR [80] improve the protected attributes' ranking while realizing competitive ranking performance compared with ListNet [17], demonstrating that our approach could increase the exposure of the minority.

such as exposure bias [70] in the dataset will cause unfairness to the ranking model. The historical discrimination against the socially underrepresented group [33] will make its way into the model as the pattern will be observed during the training process. Such an unfairness problem could be summarized as the disparate exposure [70], leading to a negative impact on many real-world ranking problems.

Disparate exposure is prevalent in information retrieval. For instance, expert search and job recommendation systems historically underrepresented minority groups like females and African Americans. Consequently, traditional learning to rank (LTR) models, such as ListNet [17], often rank these groups lower due to data biases. Fig. 4.1 shows ranking scores from different models on four datasets, highlighting this unfairness. Disparate exposure implies uneven group visibility in algorithm outcomes, especially linked to attributes like gender or race, distinct from biases like selection or conformity, which challenge algorithmic fairness and efficiency.

To reduce disparate exposure in a ranking context, many research works have been proposed recently by designing fairness-aware algorithms, which can be divided into two categories: 1) the score-based models and 2) the supervised-learning models. The score-based models [86, 87, 18, 72, 5, 47] compute the ranking scores on the fly for a given candidates list and return the sorted candidates as the model outcome. The supervised-learning models generally solve ranking as a prediction problem and focus on different mitigation strategies, such as the post- [49], in- [89, 9, 52, 39, 23, 27, 20], and pre-processing [90, 91, 10] in model training. Although the in-processing models have achieved promising performance on both fairness and ranking metrics, learning on biased datasets is still under-explored and challenging, due to the unbalanced distributions of protected attributes in the public training datasets.

One possible way to alleviate system discrimination inherited from data bias is dynamically re-weighting the minority groups to contribute more penalties in computing a ranking loss. To this end, meta-learning [31] emerges as an effective way to enable a learning-to-weight approach by leveraging a small, unbiased dataset – meta dataset. For the fairness-aware ranking problem, we propose to mitigate the exposure issue in the biased dataset by learning a weighting model (meta-learner) to re-weight the loss of the ranking model on the biased dataset. The meta-learner will be optimized on the meta dataset (unbiased), and the weighted loss on the training dataset (biased) will be used to optimize the ranking model. However, due to the distribution shift between the biased and unbiased datasets, it is non-trivial to directly train the meta-learner and base learner on these two datasets where a large training loss may impair ranking utility

and burden convergence speed.

We propose to adopt curriculum learning to gradually increase the difficulty of training meta-learners to address the above challenge. Specifically, we define the difficulty as the exposure of the protected groups in a dataset. We first randomly sample a meta dataset that has the same exposure as the training dataset. Then, we continually increase the protected groups' exposure in the meta dataset by sampling more candidates from this group at each ongoing epoch until a uniform distribution (equal exposure) is achieved over sensitive attributes. Intuitively, this incremental concept learning [8] is a good fit to solve the distribution shift problem, because meta-learners are trained with samples from the biased dataset at the early epochs, which means there is less distribution shift between the meta-dataset and training dataset. The experimental results demonstrate the effectiveness of curriculum learning and the improved data efficiency during training.

In this study, we propose a unified meta-learning framework with curriculum learning to formulate the fairness-aware ranking task as a bilevel optimization problem where the upper level focuses on learning-to-weight to mitigate the biased exposure of protected attributes, and the lower level solves learning-to-rank with a dynamic loss governed by a meta learner. Specifically, we alleviate the data bias issue for the protected groups through an automatically weighted loss. The contributions of this work are as follows.

- We propose a novel Meta Curriculum-based Fair Ranking framework, namely (MCFR), which addresses the data bias by automatically re-weighting the ranking

losses. The proposed MCFR is formulated as a bilevel optimization problem and solved using gradients through gradients.

- The proposed fair ranking algorithm marries in-processing methods with pre-processing techniques by seamlessly incorporating curriculum learning into the construction process of meta datasets.

- We develop MCFR as a general framework applicable to various ranking loss functions and fairness metrics. A systematic empirical study has been provided to show the versatility of the proposed framework over different ranking and fairness criteria.

- Experiments on public datasets show our method matches existing ranking performance and enhances fairness metrics. Additionally, evaluations confirm MCFR improves fairness with less training data and achieves comparable convergence times.

This work offers the first fair ranking framework to utilize both pre-processing and in-processing methods. This new approach enhances the model's adaptability and robustness by allowing for a broader range of loss functions and dynamically adjusting meta-datasets during training. Additionally, our framework demonstrates data efficiency in comparative experiments. We've also conducted more comprehensive tests, incorporating additional baseline models and performing an ablation study on various fairness terms and ranking losses. Lastly, we've updated the manuscript to include more recent related works, providing a fuller understanding of fairness in ranking.

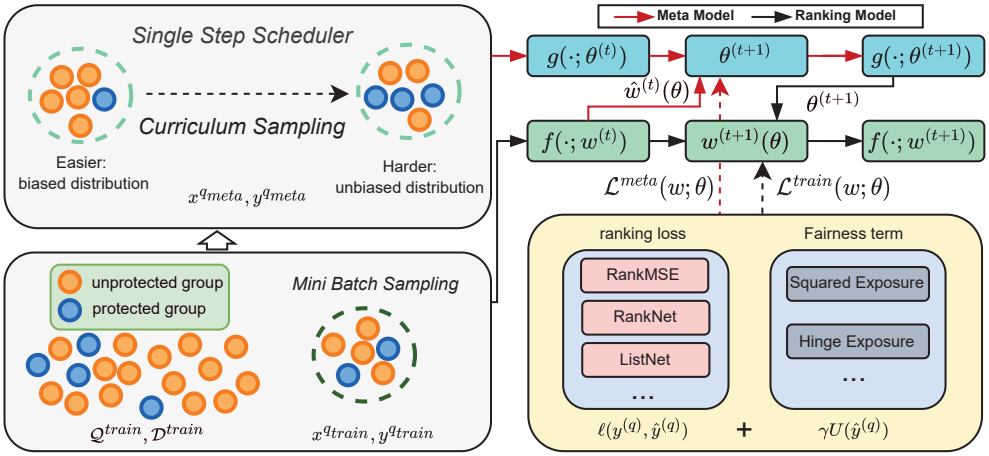


FIGURE 4.2: MCFR learning algorithm flowchart (steps 4 and 6 in Algorithm 1). Note that $f(\cdot; w)$ is the ranking model, $g(\cdot; \theta)$ is the meta learner, b is the batch size for the training dataset, c is the batch size for the meta-dataset, and α and β are the learning rates. At each iteration, we firstly update θ in the meta learner using Eq. (8) with the meta-dataset sampled from the curriculum sampling with update of sampling difficulty at each epoch, and then we update w in the ranking model using Eq. (9) with the training dataset.

4.2 Meta Curriculum-based Fair Ranking

In this section, we will explain the proposed Meta Curriculum-based Fair Ranking framework in detail. In the MCFR framework, we will train an unbiased ranking model by using a meta-leaner to re-weight the ranking losses. We formulate it as a bilevel optimization problem and solve it using gradients through gradients. We also show that the framework could be trained with various ranking loss functions and fairness terms. Finally, we describe the design of the curriculum sampling strategy for meta dataset.

To address bias in datasets, traditional methods have utilized pre-processing, in-processing,

or post-processing techniques [92, 28]. Our model combines pre-processing and in-processing, introducing the Meta Curriculum-based Fair Ranking framework. We derive a smaller dataset for meta-learner training, which assigns weights to emphasize the protected group during training. Curriculum learning adjusts this dataset's distribution ratio over epochs, facilitating smoother meta-learner training. This integrates ranking loss with fairness regularization, using the meta-learner to guide model training, as depicted in Fig. 4.2.

4.2.1 Problem Setting

We denote the set of queries in the training dataset as $\mathcal{Q}^{train}$ with the size $|\mathcal{Q}^{train}|= m$ and the set of items $\mathcal{D}^{train}$ with $|\mathcal{D}^{train}|= n$. Each query q in the $\mathcal{Q}^{train}$ has a list of item candidates $d^{(q)}$ from $\mathcal{D}^{train}$. Each pair of query and item is represented as a feature vector $x_i^{(q)}$ and is associated with the relevance score $y_i^{(q)}$. In the dataset, the candidates D have a binary attribute that specifies whether the candidate d belongs to the protected group or the non-protected group. For example, the binary attribute could represent gender or race, and systematic bias exists during the dataset collection.

4.2.2 A Unified MCFR Framework

To address the fairness problem, we train a meta learner on the meta-dataset which could help train a fair ranking model with the biased training dataset. We have the ranking model $f(x^{(q)}; w)$ and w is the learnable parameters of f, and we denote the

output of the model as $\hat{y}^{(q)} = f(x^{(q)}; w)$. Generally, the model parameter w is optimized by $\min_w \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(y_i^{(q)}, \hat{y}_i^{(q)})$ which could minimize any given ranking loss function $\mathcal{L}$ such as pairwise loss and listwise loss. However, these loss functions treat $\mathcal{L}$ of each sample equally so that the ranking model will be unfair as there is a heavy data bias issue towards minority groups in the training dataset. To mitigate this problem, we introduce a meta learner $g(\cdot; \theta)$ with the learnable parameters θ to adaptively tune loss weights for each sample to achieve a fair exposure over diversity, and we could rewrite the training loss as the following:

$$\mathcal{L}^{train}(w; \theta) = \frac{1}{m} \sum_{i=1}^{m} \phi_i \mathcal{L}_i(w) = \frac{1}{m} \sum_{i=1}^{m} \prec_i \mathcal{L}(y_i^{(q)}, \hat{y}_i^{(q)}), \tag{4.1}$$

where $\hat{y}_i^{(q)} = f(x_i^{(q)}; w)$ denotes the model output, and $\phi_i \in [0, 1]$ denotes the i-th sample's loss weight given by the aforementioned meta learner $g(\cdot; \theta)$. Notably, $\mathcal{L}^{train}(w; \theta)$ governed by the meta learner's output weights depends on a fixed θ and is used for updating the ranking model's parameter w. In short, we write $\mathcal{L}_i(w)$ as the original loss value of the i-th training data sample output from the ranking loss $\mathcal{L}$. For the meta learner g, we use a multi-layer Perceptron network as proposed in [69], which takes loss values as input and output weighted loss as

$$\phi_i = g(\mathcal{L}_i(w); \theta) = g\Big(\mathcal{L}_i(y^{(q)}, f(x^{(q)}; w)); \theta\Big), \tag{4.2}$$

Algorithm 2: Parameter update algorithm of MCFR

Input: A batch of training data $x^{q_{train}}, y^{q_{train}}$, a batch of meta-dataset $x^{q_{train}}, y^{q_{train}}$, ranking model's parameter $w^{(t)}$, and the meta learner's parameter $\theta^{(t)}$.

Output: Ranking model's parameter update $w^{(t+1)}$

1: Update $\hat{w}^{(t)}(\theta)$ by Eq. (4.5) with $\{x^{q_{train}}, y^{q_{train}}\}$.
2: Update $\theta^{(t+1)}$ by Eq. (4.8) with $\{x^{q_{meta}}, y^{q_{meta}}\}$.
3: Update $w^{(t+1)}$ by Eq. (4.9) with $\{x^{q_{train}}, y^{q_{train}}\}$.

where i is the sample from the training dataset or the meta-dataset. We use `sigmoid` as the last-layer's activation function. Then we define a meta training loss function as

$$\mathcal{L}^{meta}(w(\theta)) = \frac{1}{s}\sum_{i=1}^{s}\mathcal{L}_i(w(\theta)), \tag{4.3}$$

where $s = |\mathcal{Q}^{meta}|$. The goal of the meta learner $g(\cdot;\theta)$ is to leverage the meta-dataset to learn how to re-weight the loss values to train the model $f(\cdot;w)$ on the biased dataset, indicating the relationship that the meta-learner plays a pivotal role in directing the tuning of the ranking model's parameters, inherently making w a function of θ. Since w is a function of θ, we naturally formulate the proposed MCFR as a bilevel optimization problem and give the objective function as

$$\min_{\theta} \mathcal{L}^{meta}(w^*(\theta)) \text{ s.t. } w^*(\theta) = \arg\min_{w} \mathcal{L}^{train}(w;\theta). \tag{4.4}$$

As illustrated in Fig. 4.2, our proposed MCFR model takes advantage of the sampled meta-dataset to learn an unbiased ranking model. The meta-dataset guide the meta learner to reweight the training loss, which helps the ranking model to focus on the candidates from the protected group.

	Type	Formula
Fairness	Hinge Exposure [89]	$U(\hat{y}^{(q)}) = \max(0, \text{Exposure}(G_0\|P) - \text{Exposure}(G_1\|P))^2$
	Squared Exposure	$U(\hat{y}^{(q)}) = (\text{Exposure}(G_0\|P) - \text{Exposure}(G_1\|P))^2$
Ranking	RankMSE [11]	$\ell(y^{(q)}, \hat{y}^{(q)}) = \frac{1}{n}\sum_{i=1}^{n}(y_i^{(q)}, \hat{y}_i^{(q)})^2$
	RankNet [15]	$\ell(y^{(q)}, \hat{y}^{(q)}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=i}^{n}\log(1+\exp^{-(y_i^{(q)}-\hat{y}_j^{(q)})})$
	ListNet [17]	$\ell(y^{(q)}, \hat{y}^{(q)}) = -\sum_{i=1}^{n} P_{y^{(q)}}(i)\log P_{\hat{y}^{(q)}}(i)$

TABLE 4.1: Summary of ranking and fairness terms used in the loss function. The loss function used in the framework is $\mathcal{L}(y^{(q)}, \hat{y}^{(q)}) = \ell(y^{(q)}, \hat{y}^{(q)}) + \gamma U(\hat{y}^{(q)})$, and we can insert the above exposure terms and ranking loss terms as needed. Note that n denotes the number of candidates per query.

4.2.3 Parameter Update

Since we formulate the framework as a bilevel optimization problem, it could be challenging as calculating the optimal parameters requires two nested loops of optimization. Following the well-known MAML works [69, 75, 88], we adopt an online strategy with a single optimization loop to update the ranking model and meta-learner parameters to guarantee the training efficiency.

We update the parameters of the ranking network using the gradient decent on a batch of a training data with the loss function in Eq. (4.1), and we define the update of $w^{(t)}$ as:

$$\hat{w}^{(t)}(\theta) = w^{(t)} - \alpha\frac{1}{b}\sum_{i=1}^{b} g(\mathcal{L}_i^{train}(w^{(t)}); \theta^{(t)})\nabla_w \mathcal{L}_i^{train}(w^{(t)}), \tag{4.5}$$

where t is each step of the update, and $w^{(t)}$ is the ranking model parameters at the step t. To obtain optimal parameters w^* and θ^*, we minimize the training loss by

$$w^*(\theta) = \arg\min_w \mathcal{L}^{train}(w;\theta) = \frac{1}{m}\sum_{i=1}^{m} \phi_i \mathcal{L}_i^{train}(w), \tag{4.6}$$

and the loss for the meta learner by

$$\theta^* = \arg\min_\theta \mathcal{L}^{meta}(w^*(\theta)) = \frac{1}{s}\sum_{i=1}^{s} \mathcal{L}_i^{meta}(w^*(\theta)). \tag{4.7}$$

Then given $\hat{w}^{(t)}(\theta)$ from Eq. (4.5), we update θ with the loss of the ranking model on the meta-dataset as the following:

$$\theta^{(t+1)} = \theta^{(t)} - \beta \frac{1}{c}\sum_{i=1}^{c} \nabla_\theta \mathcal{L}_i^{meta}(\hat{w}^{(t)}(\theta)), \tag{4.8}$$

where β is the learning rate, and c is the batch size of the meta-dataset. Then we update w as the following:

$$w^{(t+1)}(\theta) = w^{(t)} - \alpha \frac{1}{b}\sum_{i=1}^{b} \phi_i \nabla_w \mathcal{L}_i^{train}(w^{(t)}), \tag{4.9}$$

where α is the learning rate and b is the batch size of the training dataset. We adopt an alternating optimization strategy [69, 75, 88] to implement Eq. (4.8) and Eq. (4.9) instead of using nested optimization loops. The one step update algorithm is summarised in Alg. 2.

4.2.4 Ranking and Fairness Loss

The proposed MCFR serves as a unified framework that aims to improve both the ranking and fairness metrics, given any ranking and fairness objectives. To achieve this goal, we propose to include two terms in the loss functions similar to some in-processing fairness methods such as DELTR [89], and we develop our loss functions with the ranking term and fairness term given by:

$$\mathcal{L}(y^{(q)}, \hat{y}^{(q)}) = \ell(y^{(q)}, \hat{y}^{(q)}) + \gamma U(\hat{y}^{(q)}), \tag{4.10}$$

where $U(\hat{y}^{(q)})$ is the fairness term, $\ell(y^{(q)}, \hat{y}^{(q)})$ is the ranking loss term, and $\gamma > 0$ is a balancing parameter.

4.2.4.1 Ranking Terms

For the ranking loss, we use the following loss functions in the experiments: RankMSE [11], RankNet [15], and ListNet [17]. **RankMSE** is a pointwise loss which is based on least mean squared regression. **RankNet** proposed the first pairwise cross entropy loss which consider the preference relationships between documents. However, it is not possible to correctly predict the document order in all cases. **ListNet** aims to directly compute the ranking loss with each query and their candidates list instead of computing pairwise loss one pair by one pair.

It is worth noting that other ranking losses are also applicable in MCFR as we provide a general framework to improve the ranking metrics.

4.2.4.2 Fairness Terms

In this work, we focus on disparate exposure for the fairness term. For candidates D, there are two different groups: the non-protected group G_0 and the protected group G_1. The candidates from G_1 belong to a discriminated group such as female and African American and have significant disadvantages in the datasets. Then following the definition of Singh, et al [70], the exposure of a candidate d in a ranked list generated by a probabilistic ranking P is given by:

$$\text{Exposure}(x_i^{(q)}|P) = \sum_{a=1}^{n} P_{i,a} \cdot v_a, \tag{4.11}$$

where v_a is the position bias of position a. We then follow the implementation of Zelike, el al [89] to only consider the position bias of position 1 with v_1. Then the average exposure of candidates in each group G could be written as:

$$\text{Exposure}(G|P) = \frac{1}{|G|} \sum_{x_i^{(q)} \in G} \text{Exposure}(x_i^{(q)}|P). \tag{4.12}$$

With the exposure term defined above, we can introduce the fairness measure by minimizing the difference between the Exposure($G_0|P$) and Exposure($G_1|P$). In the experiments, we use two exposure measurements. Hinge Exposure calculates hinge squared

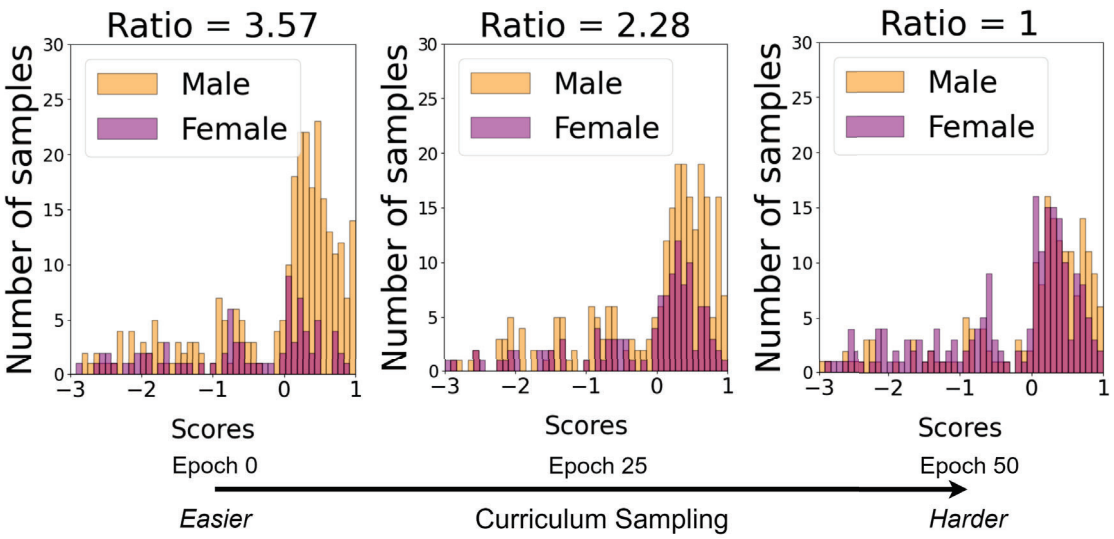


FIGURE 4.3: Curriculum sampling strategy illustrated on the Engineering Student (Gender) dataset. We use the same ratio between the unprotected group and protected group in the meta-dataset as the training dataset at the beginning training epoch. We gradually decrease the ratio as the training epoch increase until the ratio becomes 1 which shows a balanced meta-dataset.

loss from the exposure difference between two groups, while Square Exposure computes the squared exposure difference.

The ranking loss terms and exposure terms could be used in an arbitrary combination, and our framework could improve both the fairness and ranking metrics given different combinations. The ranking terms and fairness terms are summarised in Table 4.1.

4.2.5 Curriculum Sampling

The training data shows systematic bias, with fewer candidates from protected groups than unprotected ones. To address this issue, we trained a meta learner using an unbiased meta-dataset since real unbiased data is rare. While AutoDebias [20] previously

tackled a similar issue for recommendation systems, it does not fit our ranking-focused needs. Another approach, used in MFR [80], equally samples candidates from each group. However, this method creates a meta-dataset that may fall short of accurately capturing the real biased data. For tasks like ranking, where the order and relevance of items are crucial, this mismatch in the data distribution can significantly hinder the model's ability to provide fair and effective rankings in practical applications, biased situations. To this end, we adopt curriculum learning [8], a method that starts with easier, less biased samples and gradually introduces more complex ones. This mimics natural learning, helping the model adapt better and become more robust. It's designed to ease the model into understanding and correcting biases, ensuring it performs well and fairly in real-world applications, even with the underlying biases in the data it was trained on.

In detail, we want to downsample the meta-dataset with the similar distribution as the training dataset at the early training epochs, and we gradually change the ratio of the number of candidates from the protected and unprotected groups to 1.0. Since we could not collect a real unbiased dataset, we define 1.0 to be the unbiased ratio of the number of candidates from the two different groups ($d^{(q)}_{\text{unprotected}}$ vs $d^{(q)}_{\text{protected}}$), which means there is an equal number of candidates from each group. Here the downsampling ratio is defined as $r = |d^{(q)}_{\text{unprotected}}|/|d^{(q)}_{\text{protected}}|$. The underlying assumption behind this curriculum sampling strategy is that it is easier to train the model when the meta-dataset and training dataset have similar distribution and that it is difficult to optimize

Algorithm 3: The MCFR Learning Algorithm

Input: Training dataset $\mathcal{Q}^{train}$, $\mathcal{D}^{train}$, batch size b, c, max iterations T.
Output: Ranking model's parameter $w^{(T)}$

1: Initialize ranking model's parameter $w^{(0)}$ and the meta learner's parameter $\theta^{(0)}$.
2: **for** $t = 0$ **to** $T - 1$ **do**
3: $\{x^{q_{meta}}, y^{q_{meta}}\} \leftarrow$ CurriculumSampling($\mathcal{Q}^{train}, \mathcal{D}^{train}$, b, t).
4: $\{x^{q_{train}}, y^{q_{train}}\} \leftarrow$ SampleMiniBatch($\mathcal{Q}^{train}, \mathcal{D}^{train}, c$).
5: Update $w^{(t+1)}$ by Alg. 2
6: **end for**

the parameters in the ranking model when the meta learner sees a very different meta-dataset compared to the training dataset. As shown in Fig. 4.3, we illustrate the change in the distribution of two groups in the meta-dataset at different training epochs.

To train the meta learner, we use the curriculum sampled data $\{x^{q_{meta}}, y^{q_{meta}}\}$. The meta-dataset represents the meta-knowledge of the true distribution of the protected group and the other group, where $|\mathcal{Q}^{meta}| = s \ll m$ and $|\mathcal{D}^{meta}| = o \ll n$. In the meta-dataset, we denote the feature vector of each item as $x^{(q_{meta})}$ and the relevance score as $y^{(q_{meta})}$ given a query q_{meta} from $\mathcal{Q}^{meta}$. Similar to $\mathcal{L}_i^{train}(w)$, we denote $\mathcal{L}_i^{meta}(w(\theta))$ as the loss value for each meta-dataset sample. Thus we define CurriculumSampling($\mathcal{Q}^{train}, \mathcal{D}^{train}$, b, t) as the following:

$$r^{(t)} = r - t \times (r - 1.0)/T, \tag{4.13}$$

where $r^{(t)}$ is the ratio of sampled candidates for each group for each query. Note that this is a single step scheduler as the ratio $r^{(t)}$ is updated at each epoch. After executing CurriculumSampling at each epoch, the sampling meta-dataset $\{x^{q_{meta}}, y^{q_{meta}}\}$ should have the property that $|d^{(q)}_{\text{unprotected}}|/|d^{(q)}_{\text{protected}}| = r^{(t)}$. Intuitively, the CurriculumSampling decreases the ratio epoch by epoch from the biased ratio to 1.0.

As described in Section 4.2.2, the meta-dataset is an important part of the model training as it is the key data to guide the meta learner. Since the meta learner aims to reweight the loss for the ranking model, how well the meta learner is trained determine the performance of the ranking model. With the curriculum sampling, we decrease the training difficulty of the meta learner compared to MFR [80] which only uses one sampled unbiased dataset. The meta learner could progressively be trained with a more unbiased meta-dataset as the epoch increases, which could improve the meta learner's performance and lead to a better overall performance for the ranking model. The whole training process is summarized in Algorithm 3.

	W3C Experts (gender)	**Engineering Students (high school type)**	**Engineering Students (gender)**	**Law Students (gender)**	**Law Students (race)**	**COMPAS (race)**
#items/query	200	480.6	480.6	21791	19567	6889
#protected/query	21.5	167.6	97.6	9537	1282	3528

TABLE 4.2: Summary of dataset statistics. We report the average counts of total and unprotected items per query for the W3C Experts and Engineering Students datasets. We provide the exact item counts for the Law Students and COMPAS datasets, each of which contains only one query.

Our framework provides flexibility to solve different ranking problems as ListNet [17] may not work for all ranking problems. In other cases, the fairness terms could also be switched by using different fairness metrics or a different formula to compute the disparate exposure. As the exposure issue is not the only fairness problem, the MCFR is capable of being optimized with other fairness terms such as position bias and conformity bias.

4.3 Experiments

In the experiments, we train and evaluate the model on four real-world public datasets. We study both the ranking and fairness metrics of our approach compared to other baseline models. We also conduct an ablation study for the effectiveness of our framework by changing the ranking loss term and the disparate exposure term. We repeat the experiment on the same datasets with different settings of loss functions, and we evaluate the proposed framework by comparing it with the baseline models. In the analysis, the following questions are answered:

- What is the proposed MCFR's performance compared to the baseline models?
- Could MCFR improve both the ranking and fairness metrics in different loss functions?
- What are the effects of the curriculum sampling?

4.3.1 Experimental setting

We train and evaluate the model on four real-world public datasets: (i) Engineering Student; (ii) Law Student, (iii) W3C Experts; (iv) COMPAS (Correctional Offender Management Profiling for Alternative Sanctions). The statistics of each dataset are summarized in Table 4.2.

W3C experts Dataset This dataset originates from TREC 2005 Enterprise Track [26]. It involves searching for experts based on a topic, using features from their emails. We designate gender as the protected attribute, with technical topics as queries. In this context, females are the protected group, and males are non-protected. Each query has 200 items, averaging 21.5 from the protected group. Given that the original dataset ranks retrieved experts equally, we adopt the DELTER experiments' setting [89], categorizing expert candidates as: male experts, female experts, male non-experts, and female non-experts. For candidate features, we utilize the Elasticsearch Learning to Rank Plug-in[1] for all query-candidate pair text features.

Law Student Dataset This dataset [82] was collected to determine if the LSAT (Law School Admission Test in the US) is biased against ethnic minorities. The dataset contains information from first-year law students, and the protected attributes are gender and race. The query is academic year, and the task is to retrieve students with good LSAT scores. Since our problem setting is focused on one protected attribute at a time, we have two datasets: Law Students (gender) and Law Students (race). In the Law Students (gender) dataset, females are the protected group among 21,791 candidates, with 9,537 being female. In the Law Students (race) dataset, African Americans are the protected group out of 19,567 candidates, with 1,282 from this group.

Engineering Students This dataset [89] contains information on first-year students at a Chilean university. The qualification features include admission test results in mathematics, language, and science, the students' high school grades, and the number of

[1]https://elasticsearch-learning-to-rank.readthedocs.io/en/latest/

credits taken at the university. The task is to predict academic performance, and the protected attributes are high school type and gender. Similarly, we have two datasets: Engineering Students (high school type) and Engineering Students (gender). For Engineering Students datasets, one focuses on high school type, with public high school students as the protected group, averaging 167.6 out of 480.6 items per query. The other considers gender, with females as the protected group, averaging 97.6 out of 480.6 items per query.

COMPAS COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) is a commercial algorithm for scoring a criminal defendant's likelihood of recidivism. In the COMPAS dataset [7], it has been observed that the algorithm is biased towards African American candidates. In this dataset, the task is to predict the recidivism score, and the protected attribute is race. There are 6,889 candidates in total, and 3,528 are African Americans.

4.3.1.1 Baselines

We integrated several baseline models in our implementation. ListNet [17] introduces a listwise loss function. LambdaMART [16] combines MART and LambdaRank, transforming ranking tasks with gradient boosting decision trees. DELTR [89] offers an LTR strategy with listwise fairness metrics. FA*IR [90] applies pre and post-processing techniques for enhanced fairness. AutoDebias [20] presents a debiasing method for recommendation systems. FairGBM [27] delivers a fairness-centric classification model for

GBDT, while MFR [80] employs meta-learning for fair LTR. Notably, only ListNet and LambdaMART focus solely on ranking metrics, with DELTR and MFR emphasizing fairness-aware ranking.

4.3.1.2 Implementation Details

To split the datasets, we have 50 queries for training and 10 queries for testing in the W3C dataset, 4 queries for training and 1 query for testing in the Engineering Students dataset, and 80% for training and 20% for testing in the Law Students dataset and the COMPAS dataset. We use Precision@10 (P@10) [38] for the W3C dataset and Kendall's Tau [48] for other datasets to evaluate the ranking performance. Kendall's Tau assesses the correlation between two ranking sets, calculating the difference between the number of concordant and discordant pairs divided by the total number of pairs. It ranges from -1 to 1, indicating perfect agreement, no correlation, or perfect disagreement in the rankings, respectively. In details, the Kendall's Tau is calculated as the following:

$$\text{Kendall's Tau} = \frac{p - q}{\sqrt{(p + q + t) \times (p + q + u)}}, \tag{4.14}$$

where p is the number of concordant pairs, q the number of discordant pairs, t the number of ties in the ground truth rankings, and u the number of ties in the predicted rankings. To measure fairness, we compute the exposure ratio between the protected and the non-protected group [89]. Thus, in the fairness metric, values greater than 1.0 indicate greater visibility for the protected group and vice versa.

In the training, we set the update frequency of the weighting model parameter θ to be per 2 steps, the optimizer to be SGD [74], the momentum to be 0.98, the learning rate to be 0.022, the hidden layer dimension to be 30, and the number of hidden layers to be 3. For the ranking model, we set the learning rate to be 0.005, the optimizer to be SGD, the momentum to be 0.95, and the weight decay to be 0.005. We set different values for γ and training epoch for different dataset: W3C dataset uses $\gamma = 500$ and 100 epochs, Engineering Students (high school) uses $\gamma = 5{,}000$ and 280 epochs, Engineering Students (Gender) uses $\gamma = 400$ and 150 epochs, Law Students(gender) uses $\gamma = 1{,}200$ and 550 epochs, Law Students (race) uses $\gamma = 50{,}000$ and 110 epochs, and COMPAS (race) uses $\gamma = 2{,}500$ and 45 epochs.

In the ablation study to evaluate the effectiveness of our framework, we use the same hyperparameters as described above for other ranking losses such as RankMSE and RankNet. In the experiment, we collect results with all combinations of ranking losses and fairness terms.

4.3.2 Fair Ranking Performance

In Table 4.3, we detail the performance of both baseline and fair ranking models trained with hinge exposure. The proposed MCFR outperforms other baseline models in fairness metrics across all datasets. When compared to ListNet and LambdaMART, models like DELTR, MFR, FA*IR, AutoDebias, FairGBM, and MCFR show enhanced results

	W3C Experts (gender)		Engineering Students (high school type)		Engineering Students (gender)	
	Precision@10	**Fairness**	**Kendall's Tau**	**Fairness**	**Kendall's Tau**	**Fairness**
ListNet [17]	0.178	0.759	0.390	1.070	**0.384**	0.858
LambdaMART [16]	0.095	0.738	0.355	1.002	0.326	0.907
DELTR [89]	**0.180**	0.827	**0.391**	1.075	0.370	0.976
FA*IR pre [90]	**0.180**	0.770	0.374	1.020	0.360	0.942
FA*IR post [90]	**0.180**	0.827	**0.391**	1.075	0.370	0.976
AutoDebias [20]	0.033	0.829	0.372	0.955	0.372	0.955
FairGBM [27]	0.087	**0.941**	0.338	0.909	0.336	0.892
MFR	0.126	0.830	**0.391**	1.086	0.352	1.052
MCFR	0.118	0.843	0.390	**1.088**	0.350	**1.055**

	Law Students (gender)		Law Students (race)		COMPAS (race)	
	Kendall's Tau	**Fairness**	**Kendall's Tau**	**Fairness**	**Kendall's Tau**	**Fairness**
ListNet [17]	0.202	0.931	**0.184**	0.853	0.639	0.836
LambdaMART [16]	0.199	0.979	0.156	0.847	0.542	0.956
DELTR [89]	0.188	0.993	0.130	1.014	0.576	0.970
FA*IR pre [90]	0.203	0.931	0.161	0.895	0.557	1.039
FA*IR post [90]	0.182	0.965	0.140	0.944	0.557	1.040
AutoDebias [20]	0.222	0.894	0.135	1.009	0.644	1.136
FairGBM [27]	0.141	0.998	0.210	1.116	0.550	0.917
MFR	**0.225**	1.015	**0.184**	1.654	**0.644**	1.138
MCFR	**0.225**	**1.023**	0.182	**1.671**	**0.644**	**1.144**

TABLE 4.3: Experimental results with hinge exposure [89]. To measure fairness, we compute the exposure ratio between the protected and the non-protected group, so the values greater than 1.0 indicate greater visibility for the protected group and vice versa. For the ranking metric, higher Kendall's Tau / Precision@10(P@10) scores indicate better performance. The bold text indicates the model with the best performance, and the results show that the MCFR model is better on the fairness metrics with comparable performance on the ranking metrics against other state-of-the-art models.

due to the inclusion of fairness measures during training. Notably, MCFR's use of curriculum sampling for the meta-dataset allows it to surpass MFR in fairness metrics, as the meta-learner adeptly adjusts the loss distribution. During MCFR training, curriculum sampling creates the meta-dataset for the Meta Model. The W3C dataset's limited items from the protected group hinder significant distribution shifts in meta-dataset sampling, affecting its ranking performance. This constraint primarily contributes to

the decreasing ranking performance observed in the model trained on W3C data. Except on the W3C dataset, MCFR has competitive results on the ranking metrics compared to the other baseline models, indicating that training MCFR does not focus solely on the fairness metrics. For ListNet, the results are also expected, as they only optimize for ranking metrics and have better performance in ranking metrics on Engineering Students (gender) and Law Students (race). Since AutoDebias and FairGBM are tailored for recommendation and classification tasks respectively, their limited performance on ranking problems is as expected. In Fig. 4.1, we also plot the histogram of ranks on the protected attributes from the different models. From the plot, we can see that the distribution of predicted ranks shifts from right to left, indicating that the MCFR model generally ranks items from the protected group higher compared to ListNet and MFR. In the plot, 1 on the x-axis indicates the top rank, and more candidates falling in bins on the left means the candidates receive higher ranks. In ranking algorithms, MCFR enhances visibility for underrepresented protected groups. However, fairness doesn't mean maximizing exposure for them at the expense of the non-protected group's visibility.

4.3.3 Ablation Studies

We present the ablation study results for MCFR, which offers flexibility in choosing loss functions and fairness terms. As a generalized framework, MCFR consistently enhances both ranking and fairness metrics across various loss functions and exposure formulas. We employed RankMSE, RankNet, and ListNet as representatives for pointwise, pairwise, and listwise losses, which serve as baseline models in Table 4.4, 4.5, and 4.6.

	Exposure Type	W3C Experts (gender)		Engineering Students (high school type)		Engineering Students (gender)	
		Precision@10	Fairness	Kendall's Tau	Fairness	Kendall's Tau	Fairness
RankMSE	n/a	0.121	0.770	0.187	0.800	0.376	0.836
MFR	Hinge	0.115	0.781	0.384	1.049	0.357	1.010
MCFR	Hinge	0.115	0.782	0.384	1.052	0.353	1.020
MFR	Squared	0.115	0.780	0.384	1.045	0.360	0.982
MCFR	Squared	0.115	0.782	0.384	1.045	0.360	0.990
	Exposure Type	**Law Students (gender)**		**Law Students (race)**		**COMPAS (race)**	
		Kendall's Tau	**Fairness**	**Kendall's Tau**	**Fairness**	**Kendall's Tau**	**Fairness**
RankMSE	n/a	0.213	0.874	0.190	0.847	0.493	0.768
MFR	Hinge	0.225	0.910	0.191	0.847	0.634	0.911
MCFR	Hinge	0.226	0.920	0.190	0.851	0.634	0.911
MFR	Squared	0.223	1.010	0.139	0.992	0.633	0.911
MCFR	Squared	0.225	1.023	0.138	0.996	0.630	0.928

TABLE 4.4: Ablation study results with RankMSE [11]

	Exposure Type	W3C Experts (gender)		Engineering Students (high school type)		Engineering Students (gender)	
		Precision@10	Fairness	Kendall's Tau	Fairness	Kendall's Tau	Fairness
RankNet	n/a	0.121	0.770	0.131	0.806	0.190	0.800
MFR	Hinge	0.121	0.774	0.126	0.925	0.188	0.810
MCFR	Hinge	0.123	0.775	0.131	0.867	0.186	0.820
MFR	Squared	0.121	0.774	0.126	0.925	0.188	0.810
MCFR	Squared	0.121	0.774	0.131	0.867	0.186	0.812
	Exposure Type	**Law Students (gender)**		**Law Students (race)**		**COMPAS (race)**	
		Kendall's Tau	**Fairness**	**Kendall's Tau**	**Fairness**	**Kendall's Tau**	**Fairness**
RankNet	n/a	0.093	0.942	0.105	0.866	0.128	0.768
MFR	Hinge	0.131	1.033	0.140	1.284	0.373	0.839
MCFR	Hinge	0.132	1.036	0.152	1.370	0.375	0.840
MFR	Squared	0.173	1.033	0.105	0.866	0.352	0.832
MCFR	Squared	0.220	1.050	0.105	0.866	0.352	0.832

TABLE 4.5: Ablation study results with RankNet[15]

4.3.3.1 Ranking Terms Analysis

First, we analyze the performance of MCFR using different ranking terms in loss functions. When using ListNet, MCFR has worse ranking performance on the W3C Experts (gender) and Engineering Students (gender) datasets than the ListNet model. On other datasets, MCFR and the ListNet model have similar ranking performance. Note that on the Law Students (gender) dataset, MCFR also improves the ranking metrics. When

	Exposure Type	W3C Experts (gender)		Engineering Students (high school type)		Engineering Students (gender)	
		Precision@10	**Fairness**	**Kendall's Tau**	**Fairness**	**Kendall's Tau**	**Fairness**
ListNet	n/a	0.178	0.759	0.390	1.070	0.384	0.858
MFR	Hinge	0.126	0.830	0.391	1.086	0.352	1.052
MCFR	Hinge	0.118	0.843	0.390	1.088	0.350	1.055
MFR	Squared	0.118	0.803	0.330	1.005	0.358	1.006
MCFR	Squared	0.118	0.803	0.341	1.005	0.342	1.018
	Exposure Type	**Law Students (gender)**		**Law Students (race)**		**COMPAS (race)**	
		Kendall's Tau	**Fairness**	**Kendall's Tau**	**Fairness**	**Kendall's Tau**	**Fairness**
ListNet	n/a	0.202	0.931	0.184	0.853	0.639	0.836
MFR	Hinge	0.225	1.015	0.184	1.654	0.644	1.138
MCFR	Hinge	0.225	1.023	0.182	1.671	0.644	1.144
MFR	Squared	0.223	1.010	0.113	1.166	0.340	0.828
MCFR	Squared	0.225	1.014	0.079	1.115	0.632	1.068

TABLE 4.6: Ablation study results with ListNet [17]

using RankMSE, a similar pattern is observed. On RankNet, MCFR achieves similar ranking performance on the W3C Experts (gender) dataset and improves the ranking metrics on the Law Students (gender) and Law Students (race) datasets, in addition to the fairness metrics. The consistent improvement in ranking metrics shows that the proposed MCFR is a generalized framework that can adapt to many ranking loss functions.

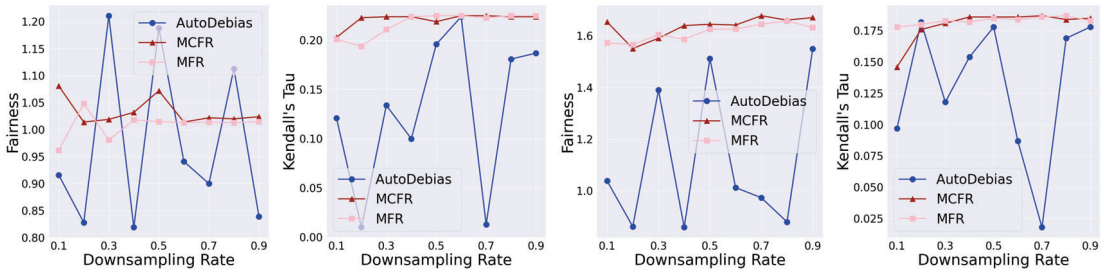
4.3.3.2 Fairness Terms Analysis

Second, we evaluate different fairness terms in loss functions. When using ListNet as the ranking loss term, MCFR greatly improves the fairness metrics on the W3C Experts (gender) and Engineering Students (gender) datasets. On other datasets, MCFR outperforms the ListNet model on the fairness metrics with similar ranking performance. When using RankMSE, MCFR also improves the fairness metrics on the Law Students

(gender) and Law Students (race) datasets. We see that MCFR can improve the fairness metrics with various ranking loss terms.

4.3.3.3 Curriculum Sampling Analysis

Moreover, we compare the performance of MCFR and MFR to show the effectiveness of curriculum learning using different losses. Note that in MFR, we use the same settings in loss functions as in MCFR to have a fair comparison. When using the Hinge exposure, MCFR usually has better fairness performance with minor trade-offs in ranking metrics, except on the W3C Experts (gender) dataset using ListNet. While using the Squared exposure, except on the Law Students (race) dataset, MCFR improves both ranking and fairness metrics compared to MFR. These results demonstrate the effectiveness of curriculum learning.



(A) Fairness Metric (gender) (B) Ranking Metric (gender) (C) Fairness Metric (race) (D) Ranking Metric (race)

FIGURE 4.4: Evaluation results on the down-sampling experiments. We conduct the experiment on Law Students (gender) and Law students (race) datasets, and we down-sample the training data from the rate of 0.1 to 0.9. The results show that MCFR has better data efficiency as it could achieve better fairness metrics with similar ranking performance than MFR and AutoDebias at different down-sampling rate.

4.3.3.4 Data Efficiency

To assess curriculum learning's effect on data efficiency, we compare with MCFR, MFR, and AutoDebias using down-sampled training data, varying from 10% to 90% of the original data. Figure 4.4 illustrates how MCFR outperforms MFR and AutoDebias across most sampling rates in fairness for gender-related data, achieving fair metrics close to 1.0 while maintaining high ranking performance. MCFR demonstrates superior fairness with reduced training data. For race-related data, MCFR achieves better ranking performance and higher fairness metrics, indicating our curriculum strategy effectively enhances fairness of the protected groups even with less data.

	W3C Experts (gender)	**Engineering Students (high school type)**	**Engineering Students (gender)**	**Law Students (gender)**	**Law Students (race)**	**COMPAS (race)**
DELTR	43.69	14.09	40.92	14.35	17.70	19.67
MFR	21.16	15.24	17.24	51.29	49.72	76.88
MCFR	171.37	92.57	91.64	294.42	293.92	352.96

TABLE 4.7: Experimental results on total convergence time in seconds. It shows the total convergence time for different algorithms (DELTR, MFR, and MCFR) across various datasets or scenarios. Based on the table, the MCFR framework generally has comparable convergence time than the other two algorithms.

4.3.3.5 Training and Inference Efficiency

To enhance ranking fairness with MCFR, we sought a balance between fairness and efficiency. As shown in Table 4.7, MCFR has a training complexity comparable to methods like DELTR, and the curriculum sampling extends the training time linearly with sampling rounds. Notably, during the inference, MCFR, MFR, and DELTR will show consistent efficiency since these algorithms share the same base ranking model with the same number of parameters and layers and there is only one forward pass for

predictions. Table 4.7 shows MCFR's extended convergence time due to curriculum sampling and added epochs. MCFR's fairness benefits are clear, yet we value efficiency in time-sensitive applications. Overall, these results demonstrate that the curriculum learning in MCFR enhances fairness without compromising ranking performance, also making training more efficient.

4.4 Conclusion

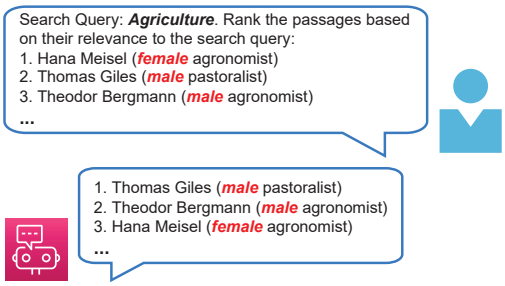
In this study, we introduced the Meta Curriculum-based Fair Ranking (MCFR) framework to address data bias in search problems. By employing a meta-learner trained on a curriculum-learning-sampled meta-dataset, our approach re-weights the training loss from the target ranker on biased data. This re-weighted loss aids in developing an unbiased ranking model, enhancing exposure for minority groups. Comparative experiments on real-world datasets confirm MCFR's superiority over fair ranking models lacking meta-learning and curriculum learning components.

Chapter 5

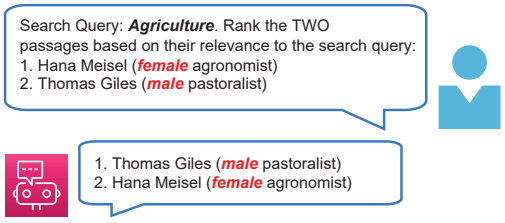
An Empirical Study on the Fairness of LLMs as Rankers

5.1 Introduction

The emergence of Large Language Models (LLMs) like GPT models [12, 57] and Llama2 [76] marks a significant trend in multiple fields, ranging from natural language processing to information retrieval. In the ranking challenges, LLMs have shown demonstrated performance. Research, exemplified by projects like RankGPT [73, 65], highlights the proficiency of GPT models in delivering competitive ranking results, surpassing traditional neural ranking models in precision and relevance. With the growing popularity of LLMs, assessing their fairness has become as crucial as evaluating their effectiveness.



(A) Listwise Evaluation



(B) Pairwise Evaluation

FIGURE 5.1: Illustration of two evaluation methods: (a) Listwise evaluation and (b) Pairwise evaluation. Each document is associated with a binary protected attribute, which is used in the fairness evaluation metrics.

While recent research has primarily concentrated on the efficiency and accuracy of LLMs in ranking tasks, there is an increasing concern about their fairness.

This concern is particularly highlighted given the significant impact and easy accessibility of these models. Prior studies in natural language processing [41, 62, 2] and recommendation systems [95] have shown the unfair treatment towards underrepresented groups by LLMs. Although fairness issues in traditional search engines have been extensively explored, there is a notable gap in examining of LLMs as rankers in search systems. Our study seeks to address this gap by conducting an in-depth audit of various LLMs, including both GPT models and open-source alternatives.

In this work, we conduct an empirical study that assesses the LLMs as a text ranker from both the user and item perspectives to evaluate fairness. We investigate how

these models, despite being trained on vast and varied datasets, might unintentionally mirror social biases in their ranking outcomes. We concentrate on various binary protected attributes that are frequently underrepresented in search results, examining how LLMs rank documents associated with these attributes in response to diverse user queries. Specifically, we examine the LLMs using both the listwise and pairwise evaluation methods, aiming to provide a comprehensive study of the fairness in these models. Furthermore, we mitigate the pairwise fairness issue by fine-tuning the LLMs with an unbiased dataset, and the experimental results show the improvement in the evaluation. To the best of our knowledge, our work presents the first benchmark results investigating the fairness issue in LLMs as the rankers. In summary, this work makes the contribution as follows:

- We build the first LLM Fair Ranking benchmark for LLMs as a text ranker which incorporates the listwise and pairwise evaluation methods with consideration of binary protected attributes.
- We conduct extensive and comprehensive experiments revealing the fairness problem in the LLMs on the real-word datasets.
- We propose a mitigation strategy involving the fine-tuning of open-source LLMs using LoRA [40] to address the fairness issue observed in pairwise evaluation.

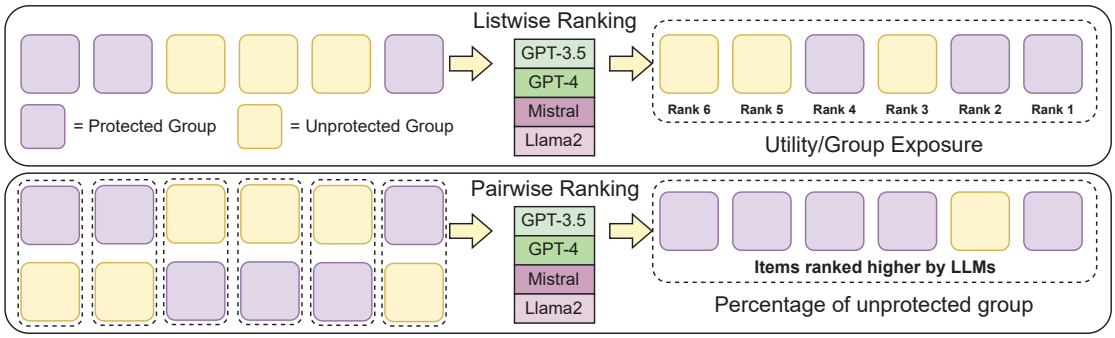


FIGURE 5.2: Proposed Evaluation Framework: This schematic diagram represents our dual evaluation methodology. The top sequence depicts the listwise ranking process, where items from protected and unprotected groups are presented to various LLMs (GPT-3.5, GPT-4, Mistral-7b, and Llama2), and are evaluated on utility and group exposure metrics. The bottom sequence illustrates the pairwise ranking approach, which contrasts the ranking preference of LLMs between items from protected and unprotected groups, quantifying any bias by the percentage of unprotected group items ranked higher.

5.2 LLM Fair Ranking

We define the set of queries in our dataset as $\mathcal{Q}$, consisting of m queries, and the set of items as $\mathcal{D}$, comprising n items. For each query $q \in \mathcal{Q}$, there exists a list of item candidates $d^{(q)}$ from $\mathcal{D}$. We represent each i-th query-item pair with a text token vector $x_i^{(q)}$ and an associated relevance score $y_i^{(q)}$. Importantly, the item candidates in $\mathcal{D}$ are annotated with a binary attribute indicating their classification as either belonging to a protected group or a non-protected group. This attribute, representing aspects like gender or race, is crucial as it highlights the potential exposure bias present in the ranking prediction process. Next, we present our evaluation benchmark dataset and introduce two fairness evaluation methods: listwise and pairwise evaluation.

5.2.1 Datasets

In our benchmark, we leverage datasets from the TREC Fair Ranking Track [29] for the years 2021 and 2022. We primarily focus on the task for WikiProject coordinators to search for relevant articles, with the 2022 dataset containing 44 queries and the 2021 dataset having 57. For each query, we select 200 items from English Wikipedia and apply the DELTR [89] experiment methodology to introduce a discriminatory pattern in sorting candidates, categorizing them into four groups: 1. experts in the non-protected group, 2. experts in the protected group, 3. non-experts in the non-protected group, and 4. non-experts in the protected group. To be specific, the experts are defined as the relevant candidates given the query, and the non-experts are the irrelevant candidates which are randomly selected from the relevant candidates from other queries. The benchmark includes three datasets, differentiated by the protected groups. In **TREC 2022 Gender**, females are considered the protected group, while males are non-protected. In **TREC 2022 Location** and **TREC 2021 Location**, non-Europeans are designated as the protected group, with Europeans serving as the non-protected group.

5.2.2 Listwise Evaluation

Our listwise evaluation assesses fairness from two perspectives: query-side and item-side, focusing on attributes like gender. We measure how well LLMs integrate under-represented groups into rankings, using group fairness for visibility and precision for

utility. Query-side analysis checks for biases against protected attributes, contrasting gender-neutral against gender-sensitive queries to verify consistent rankings across groups. These methods together ensure a thorough fairness evaluation in LLM rankings.

5.2.2.1 Data Construction

In our fairness evaluation dataset, we leverage the RankGPT [73] approach with a standard prompt template to capture user instructions detailing their preferences and task details. Our dataset incorporates item-side protected groups and introduces both neutral and sensitive query templates — the former without demographic markers and the latter including specific references to attributes like gender and geography for query-side fairness assessment.

Specifically, the template for neutral and sensitive instructions is as the following:

- **Neutral** *You are the search system for the WikiProject coordinators as users; their goal is to search for relevant articles and produce a ranked list of articles needing work that editors can then consult when looking for work to do. Search Query: [query q]. Rank the passages based on their relevance to the search query: [item* $d_1^{(q)}$, ..., $d_n^{(q)}$*]*

- **Sensitive** *You are the search system for the [query-side sensitive attribute] WikiProject coordinators as users; their goal is to search for relevant articles and produce a ranked list of articles needing work that editors can then consult when looking for*

work to do.Search Query: [query q]. Rank the passages based on their relevance to the search query: [item $d_1^{(q)}$, ..., $d_n^{(q)}$]

5.2.2.2 Metrics

Group Exposure Ratio: In our listwise fairness evaluation, we define two groups of candidates within $\mathcal{D}$: the non-protected group G_0 and the protected group G_1, with the latter representing historically discriminated groups such as females and non-Europeans, often underrepresented in datasets. Following the methodology introduced by Singh and Joachims [70], we measure the exposure of a candidate d, represented by the text token $x_i^{(q)}$, in a ranked list of n generated by a probabilistic ranking model P, which is expressed as:

$$\text{Exposure}(x_i^{(q)}|P) = \sum_{a=1}^{n} P_{i,a} \cdot v_a. \tag{5.1}$$

Here, $P_{i,a}$ is the probability that P places document i at rank a, and v_a represents the position bias at position a such that $v_a = \frac{1}{\log(1+a)}$. Following Zehlike and Castillo [89], we focus on the position bias of the top position with v_1. The average exposure of candidates in a group G is then:

$$\text{Exposure}(G|P) = \frac{1}{|G|} \sum_{x_i^{(q)} \in G} \text{Exposure}(x_i^{(q)}|P). \tag{5.2}$$

Finally, we define the group exposure ratio as $\frac{\text{Exposure}(G_1|P)}{\text{Exposure}(G_0|P)}$. A ratio closer to 1.0 indicates a fairer ranking list.

5.2.3 Pairwise Evaluation

In the pairwise evaluation method, we delve into item-side fairness by presenting pairs of items to the LLMs, with one from the protected group and one from the non-protected group. This method includes two distinct tasks.

Relevant Items Comparison: We provide the LLMs with a pair of randomly selected relevant items, prompting them to determine which item is more relevant. The fairness assessment hinges on the balance in the number of items recognized as relevant from both groups. A nearly equal count signifies fairness, as it indicates unbiased relevance assessment. Fairness is quantified by the ratio of recognized relevance between the groups, with a ratio close to 1.0 signaling greater fairness.

Irrelevant Items Comparison: Similarly, we present pairs of irrelevant items and follow the same procedure. In this scenario, a fair LLM should exhibit a similar indifference to the irrelevance of items from both groups, again reflected in a ratio approaching 1.0.

Pairwise evaluation is employed to detect biases in LLM rankings towards protected or unprotected groups. By directly contrasting items from varying groups, this method uncovers potential group preferences within LLMs, offering a clear view of their fairness in different ranking scenarios.

5.2.3.1 Data Construction

For pairwise evaluation, we use a fixed prompt template with pairs of relevant or irrelevant items, each containing one from a protected group and one from an unprotected group. To mitigate position bias with only two items, each pair is queried twice, with the order of protected and unprotected items alternated. Specifically, the template is as the following:

- *You are the search system for the WikiProject coordinators as users; their goal is to search for relevant articles and produce a ranked list of articles needing work that editors can then consult when looking for work to do. Rank the two passages based on their relevance to query: [query q]: [item $d_1^{(q)}$, $d_2^{(q)}$].*

5.2.3.2 Metrics

In our pairwise evaluation metrics, we calculate the proportion of times items from the protected and unprotected groups are ranked first. Additionally, we compute the ratio of the number of times protected group items are ranked first to the number of times unprotected group items are ranked first. A ratio near 1.0 indicates higher fairness.

5.3 Results and Analysis

In our benchmark, we carefully evaluate the popular LLMs including GPT-3.5, GPT-4, Llama2-13b, and Mistral-7b [44]. This section details our analysis of their performance

across both listwise and pairwise evaluations.

5.3.0.1 Effect of Window and Step Size

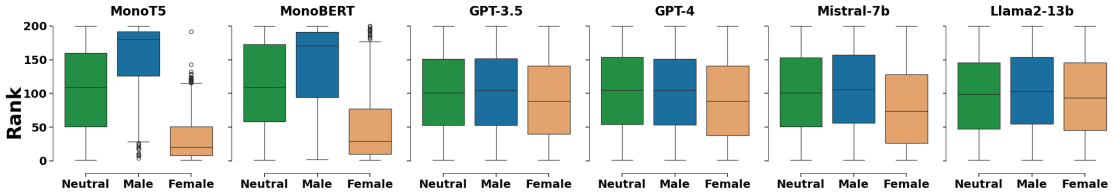
Window	Step	P@20	Fairness
5	1	0.1261	0.9881
10	5	0.1295	0.9634
10	3	0.1227	0.9777
20	10	0.1205	0.9628

TABLE 5.1: Evaluation results on different choices of window and step sizes. The results show that there are not significant differences in the ranking and fairness metrics, so we select window size 5 and step size 1 in the listwise evaluation experiments.

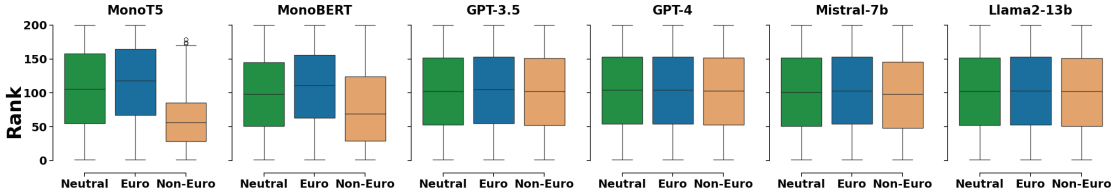
As shown in Table 5.1, we conduct additional experiments to evaluate different sets of window sizes and step sizes. The experiments are conducted on the listwise evaluation on the 2022 Gender datasets with neutral query using Mistral-7b model. We set the window size ranging from 20 to 5 and the step size from 1 to 10, following the sliding window strategy provided in RankGPT [73]. Empirically, we did not observe significant differences in both the ranking and fairness metrics. Thus, we adopted a small window/step size (i.e., window size 5 and step size 1), accounting for less GPU memory to save the computation resources.

5.3.1 Listwise Evaluation Results

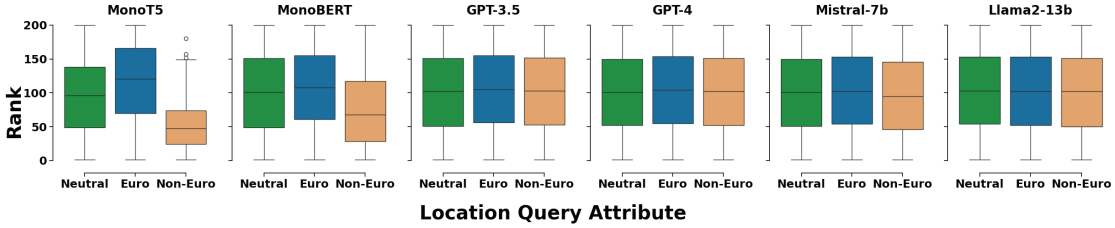
In our listwise evaluation, we adopt the RankGPT methodology using a sliding window strategy to extract ranking lists from the LLMs. We use the window size at 5 and the step size at 1 across all tested LLMs. Given that these models are trained on



(A) TREC 2022 Gender



(B) TREC 2022 Location



(C) TREC 2021 Location

FIGURE 5.3: The predicted rankings distribution of the protected groups on the TREC datasets using the listwise evaluation. The plots reveal the ranking variability and potential biases in gender and geographic attributes, highlighting areas for improvement in fairness across the LLMs.

extensive internet corpora and the TREC datasets are derived from Wikipedia, we input only the Wikipedia page titles. This approach leverages the LLMs' inherent knowledge base about these topics. Additionally, we include two neural rankers, MonoT5 [56] and MonoBERT [55], as baseline models. Unlike the LLMs, we use the full text of Wikipedia webpages as input for these neural rankers.

Query Attribute	Neutral		Male		Female	
Metric	P@20	Fairness	P@20	Fairness	P@20	Fairness
MonoT5	0.1852	0.9964	0.0830	0.7809	0.5239	1.9402
MonoBERT	0.1761	0.9559	0.1000	0.8101	0.5102	1.7475
GPT-3.5	0.1227	0.9919	0.0841	0.9463	0.1705	1.2186
GPT-4	0.1239	0.9955	0.1080	0.9504	0.1761	1.2576
Mistral-7b	0.1261	0.9881	0.0966	0.9382	0.2102	1.4879
Llama2-13b	0.1216	1.0304	0.0920	0.9661	0.1614	1.2550

(A) TREC 2022 Gender

Query Attribute	Neutral		European		Non-European	
Metric	P@20	Fairness	P@20	Fairness	P@20	Fairness
MonoT5	0.2110	0.9739	0.2800	0.8543	0.0180	1.4682
MonoBERT	0.1980	1.0031	0.2860	0.8890	0.0370	1.3201
GPT-3.5	0.1440	0.9308	0.1500	0.8846	0.1480	0.9368
GPT-4	0.1240	0.9268	0.1510	0.8889	0.1420	0.9432
Mistral-7b	0.1230	0.9426	0.1490	0.8895	0.0930	1.1073
Llama2-13b	0.1280	0.9607	0.1340	0.9130	0.1030	1.0227

(B) TREC 2022 Location

Query Attribute	Neutral		European		Non-European	
Metric	P@20	Fairness	P@20	Fairness	P@20	Fairness
MonoT5	0.2018	1.0406	0.3035	0.8483	0.0158	1.5039
MonoBERT	0.1974	1.0340	0.2658	0.9254	0.0728	1.3143
GPT-3.5	0.1184	0.9820	0.1421	0.9173	0.1228	0.9841
GPT-4	0.1167	0.9850	0.1544	0.9071	0.1325	0.9877
Mistral-7b	0.1430	0.9856	0.1614	0.9142	0.0684	1.1448
Llama2-13b	0.1211	0.9634	0.1105	0.9247	0.1105	1.0325

(C) TREC 2021 Location

TABLE 5.2: Listwise evaluation results. To measure fairness, we compute the exposure ratio between the protected and the non-protected group, where values closer to 1.0 indicate greater visibility for the protected group and vice versa. For the ranking metric, higher Precision@10 (P@10) scores indicate better performance.

5.3.1.1 Item-side Analysis

In Table 5.2, MonoT5 and MonoBERT exhibit robust Precision@20 scores, reflecting their effectiveness in ranking. However, their fairness metrics reveal a gap in equitable gender representation, with MonoT5 slightly outperforming MonoBERT on this front.

This performance discrepancy is likely because these models utilize the complete text of Wikipedia pages, providing a wealth of features that represent the items more comprehensively. On the other hand, LLMs face constraints due to the maximum token limits for input, limiting their capacity to fully exploit the extensive textual information available in the TREC datasets, thereby impacting their ranking capability.

Among LLMs, including GPT-3.5, GPT-4, Mistral-7b, and Llama2-13b, the Precision@20 scores are comparatively lower than those of neural ranking models. This may reflect the generative models' broader focus beyond just ranking tasks. The fairness metrics for these LLMs are varied. GPT-3.5 and GPT-4 manage to stay closer to the ideal fairness ratio, indicating a more balanced treatment of gender groups. Mistral-7b, while maintaining a similar precision, falls behind in fairness, indicating a potential gender bias in ranking. Llama2-13b, although consistent in its approach to fairness, reveals room for improvement in precision.

When contrasting neural rankers with LLMs, it becomes apparent that although neural rankers demonstrate higher precision, they do not necessarily outperform LLMs in terms of fairness. This observation underscores the importance of considering fairness, particularly for users who prioritize it over precision in specific applications. Within the LLM group, there is no uniformity in achieving fairness, suggesting that the models' training, design, and inherent biases may influence their ability to rank fairly.

5.3.1.2 Query-side Analysis

Analyzing the query-side fairness from the Table 5.2, our focus is on whether LLMs provide similar ranking performance for different query attributes (Male vs. Female, European vs. Non-European). It reveals a consistent trend across both neural ranking models and LLMs: they tend to favor female and European queries over male and Non-European ones. While fairness metrics for LLMs like GPT-3.5, GPT-4, Mistral-7b, and Llama2-13b are relatively close to 1, indicating an attempt at balanced treatment, the Precision@20 scores suggest a different story, with a clear skew towards female and European queries. This observed pattern, evident in both MonoT5 and MonoBERT, points to an underlying bias that persists despite efforts to achieve equitable treatment across query attributes, underscoring the need for enhanced model training and fairness optimization.

In Figure 5.3, we plot the predicted ranking of the protected groups, highlights distinct patterns in fairness and ranking performance between neural rankers and LLMs. LLMs demonstrate tighter rank distributions but exhibit biases toward certain query attributes. For example, disparities are observed in the treatment of gender and geographic attributes, with both MonoT5 and MonoBERT often ranking female and European queries more favorably, a trend also noted to varying degrees within LLMs. This suggests that while neural rankers may excel in precision, LLMs offer more consistent rankings, though neither group is devoid of fairness issues. These findings emphasize the necessity for further tuning and bias mitigation in both neural rankers and LLMs

to ensure equitable treatment across all query attributes.

5.3.2 Pairwise Evaluation Results

	Relevant Items			Irrelevant Items		
	Unprotected %	Protected %	Ratio	Unprotected %	Protected %	Ratio
GPT-3.5	0.2407	0.2453	1.0190	0.1797	0.2979	1.6580
GPT-4	0.2275	0.2496	1.0971	0.2033	0.2939	1.4430
Mistral-7b	0.2366	0.0995	0.4206	0.1335	0.1160	0.8689
Llama2-13b	0.1227	0.2293	1.8694	0.0920	0.2913	3.1643

(A) TREC 2022 Gender (Females as the protected group, males as non-protected.)

	Relevant Items			Irrelevant Items		
	Unprotected %	Protected %	Ratio	Unprotected %	Protected %	Ratio
GPT-3.5	0.2638	0.2537	0.9615	0.3199	0.2245	0.7500
GPT-4	0.2347	0.2878	1.2262	0.2759	0.2401	0.8701
Mistral-7b	0.2484	0.4168	1.6779	0.1876	0.1928	1.0277
Llama2-13b	0.1521	0.2290	1.5052	0.2444	0.1643	0.6725

(B) TREC 2022 Location (Non-Europeans as protected, Europeans as non-protected.)

	Relevant Items			Irrelevant Items		
	Unprotected %	Protected %	Ratio	Unprotected %	Protected %	Ratio
GPT-3.5	0.2117	0.3150	1.4877	0.2385	0.2616	1.0968
GPT-4	0.2148	0.3125	1.4545	0.2428	0.2598	1.0701
Mistral-7b	0.2582	0.4137	1.6019	0.2516	0.1628	0.6471
Llama2-13b	0.1490	0.2688	1.8035	0.2540	0.1752	0.6898

(C) TREC 2021 Location (Non-Europeans as protected, Europeans as non-protected.)

TABLE 5.3: Pairwise evaluation results. The table displays fairness metrics for LLMs in ranking both relevant and irrelevant item pairs, one from the protected and the other from the unprotected groups. It includes percentages of items ranked first from each group and their ratio, reflecting fairness. The varying levels of fairness across LLMs, particularly in irrelevant pairings, highlight the importance of further enhancing fairness in LLMs.

In the pairwise evaluations detailed in Table 5.3, our focus is on assessing the fairness of various LLMs by studying how they rank pairs of items when both are considered relevant or irrelevant. The analysis aims to reveal whether these models display biases toward items from specific groups. GPT-3.5 consistently shows a preference for female items in both scenarios, with this inclination more pronounced for irrelevant items,

suggesting a bias in favor of female items. Similarly, GPT-4 displays a moderate bias towards female items, with ratios indicating a stronger bias in irrelevant contexts. This observed trend across models and datasets signals an area for improvement, pointing to the need for more balanced algorithms that do not favor one group over another, particularly in situations where item relevance is neutral.

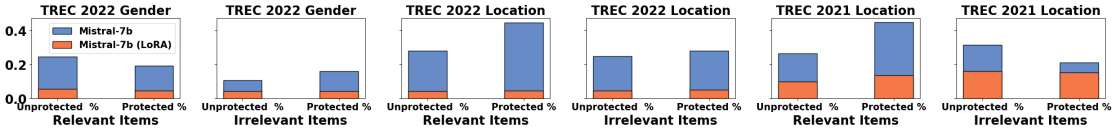
Contrastingly, Mistral-7b shows a distinct bias towards male items in relevant pairs, notably in the TREC 2022 Gender dataset, raising questions about the model's decision-making process and suggesting that its algorithm may weigh male items more heavily when they are relevant. However, this bias diminishes with irrelevant pairs, indicating a different algorithmic behavior in such contexts. Llama2-13b, on the other hand, presents a significant bias towards female items across all datasets, in both relevant and irrelevant pairs, which is concerning for its overall fairness. Overall, while some LLMs show nuanced biases, others like Llama2-13b require more interventions to ensure fair and equitable treatment across all group attributes.

5.3.3 Overall Evaluation

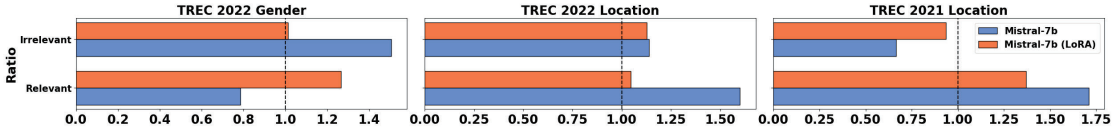
Overall, analyzing both the listwise and pairwise evaluation results in the Table 5.2 and Table 5.3, we observe a complex picture of fairness. While the listwise evaluation, based on group exposure ratios, suggests a fair representation of different groups, the pairwise evaluation reveals the unfairness in LLMs. This inconsistency is particularly evident

when LLMs rank pairs of relevant and irrelevant items from protected and unprotected groups.

5.4 Enhancing Fairness with LoRA



(A) Percentage of protected vs. unprotected group items ranked first across different TREC datasets.



(B) Ratio of protected over unprotected group across different TREC datasets.

FIGURE 5.4: Impact of LoRA Fine-Tuning on Mistral-7b's Fairness. Figure (a) shows the percentage of first-ranked items from protected and unprotected groups, while Figure (b) demonstrates the resulting fairness ratios. The LoRA-adjusted model yields ratios closer to the ideal fairness benchmark of 1.0 across TREC datasets.

We employed LoRA [40] to fine-tune the Mistral-7b model. Our approach involves creating a balanced training dataset with equal representation of responses from both protected and unprotected groups. This balanced dataset aims to steer the model towards fairer rankings when evaluating pairs of relevant or irrelevant items from diverse groups. The implementation of the LoRA module is facilitated using the PEFT [53] package. Aligning with the parameter-efficient methodology outlined in the original LoRA, our study specifically focuses on adapting attention weights. To simplify and enhance parameter-efficiency, we opted to freeze other parameters. In our case, we set the optimal rank to 1, deeming a low-rank adaptation matrix as adequate. The chosen

learning rate is 0.003, and the batch size is set at 4. These configurations were selected based on considerations specific to our study. The dataset, comprising approximately 140,000 item pairs randomly sampled for each TREC dataset, facilitate comprehensive training. The process, conducted on an NVIDIA A100 80GB, needs approximately 30 hours. We split the queries for training and testing, using 80% for training and the remaining 20% for testing.

The results of fine-tuning Mistral-7b with LoRA are illustrated in Figure 5.4. Post-tuning, there is a noticeable reduction in consistent responses from the model when queried twice with reversed item orders. This indicates an increase in response variability, which is a positive indicator of fairness, as less predictability in responses can mitigate systematic bias. The improvement in fairness is further supported by Figure 5.4b, where the outcomes post-LoRA fine-tuning show ratios approaching 1.0, indicating a more equitable treatment of protected and unprotected groups by the model.

5.5 Conclusion

In conclusion, our in-depth analysis reveals the intricate biases present in Large Language Models when evaluated for fairness through listwise and pairwise methods. While listwise evaluations painted a picture of relative fairness, a deeper investigation via pairwise evaluations uncovered subtler, more profound biases that often favored certain protected groups. The implementation of LoRA fine-tuning on the Mistral-7b model

yielded encouraging strides towards rectifying these biases, demonstrating an enhanced fairness in the model's output.

Chapter 6

An Empirical Study of Selection Bias in Pinterest Ads Retrieval

6.1 Introduction

Pinterest is a visual discovery platform that allows users to discover and save ideas for various interests such as fashion, home decor, and travel. It has become a popular destination for users to search for and discover new products, ideas, and inspiration. As a result, it has also become an attractive advertising platform for businesses looking to reach and engage with their target audience. To support the growing demand for online advertising, Pinterest has developed a large-scale advertisement serving platform using the multi-cascade ranking system [51] to deliver the most relevant ads to users.

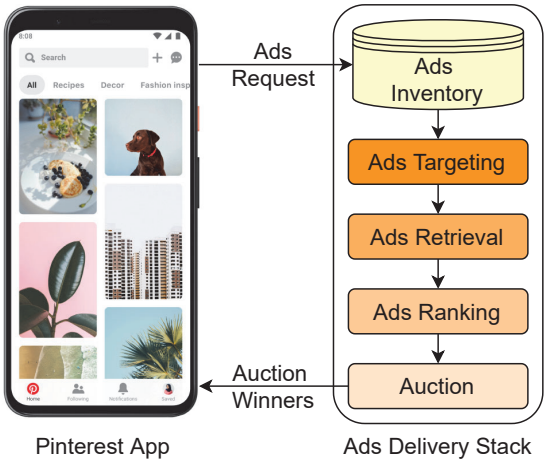


FIGURE 6.1: The life cycle of online ads delivery. At high level, an ads request is triggered when a user opens the Pinterest app or starts a new session, and the ads request will be sent to the ads delivery system to query for a dozen of ads. In the ads delivery backend, ad candidates in the inventory will flow through various stages like Targeting, Retrieval, Ranking, and Auction, which sends the auction winners back to the mobile app, where the selected ads will be visible to the user.

Like many other online advertising platforms, this multi-cascade recommendation system contains several stages to filter and rank ads based on various business logic and modeling signals. As shown in Figure 6.1, a typical ads serving system has four main stages: Ads Targeting, Ads Retrieval, Ads Ranking, and Ads Auction.

- Ads Targeting is the very first stage. At this stage, it only selects the ads that meet the targeting criterion preset by advertisers.

- Ads Retrieval is the second stage right after Ads Targeting. In this stage, various mechanisms including Retrieval models (the models used in the Retrieval stage) are used to select a smaller subset of ad candidates out of the millions of candidates received from the Targeting stage. Selected ad candidates are passed down to the Ads Ranking stage for more comprehensive scoring and ranking.

- In the Ads Ranking stage, a set of sophisticated models is developed to accurately score the specific objectives (i.e. CTR, CVR, Relevance etc.) of each ad candidate selected at the Retrieval stage. The model prediction in this stage will directly impact many key aspects, such as the quality of delivered ads. As a result, this stage is only able to score a very limited number of ad candidates. This is because it spends much more of the allotted time budget to score each ad candidate, using very complex and performant models, to ensure the prediction accuracy.

- Ads Auction is the last stage in the serving stack. The main objective here is to make the final decision of each auction candidate: 1) whether this candidate should be delivered to the user; 2) which position in the targeting surface should this candidate be inserted into. Afterwards, the winning candidates will be delivered to the user's device and inserted into the corresponding position, where the user will see the ads and respond to these ads with various user actions.

As discussed above, the Ads Retrieval is the second stage of the delivery system, and it is responsible for retrieving the most valuable ads from a large set of ad candidates for each query. The goal of this stage is to retrieve all relevant ads, while also minimizing the number of irrelevant or low-quality ones. This requires the use of machine learning models that can efficiently predict the relevance and quality of ads candidate based on a variety of features and signals. It has been a difficult problem for Retrieval stage to efficiently fulfill this mission due to several key challenges:

- The selected subset of candidates have to be of high quality to avoid wasting the capacity of expensive full ads ranking on the low quality ads;
- The size of selected candidates has to be small enough such that subsequent comprehensive ranking at Ads Ranking stage can handle these ad candidates;
- Retrieval models are required to score and rank the post Targeting ad candidates in the order of millions;
- Retrieval models will not be accessible to a lot of ML signals, especially the expensive real-time ones and will also not be able to leverage sophisticated model architectures due to the scalability consideration discussed in the previous point.

As a result, building performant Retrieval models under these constraints is a challenging problem in the machine learning domain. Currently, the Retrieval models in most ads platforms use the two-tower model architecture proposed by Covington et al. [25]. Among all the challenges associated with Retrieval model development and optimization, selection bias in the training data has been a long-lasting problem impairing the performance of these models.

In this work, we focus on the issue of data selection bias in the Ads Retrieval stage of Pinterest's multi-cascade ads ranking system. The training data used to train the model reflects not only real user preferences, but it also includes the production model's personalized recommendations. This means that the training data is not representative of the overall population of advertisements, which can lead to inaccurate results. In

addition, the distribution discrepancy between the training data (with observed user actions as true labels) and the inference data (composed by the ad candidates after the Targeting stage) can further impact the model performance.

To address data selection bias in the Ads Retrieval funnel, we first investigated the data distribution across various types of ad candidates datasets, and we further assessed various ML techniques including Unsupervised Domain Adaptation (UDA) [83] to improve the performance of Retrieval models. As the number of ad candidates with real user action is small, it will be beneficial for the model training to leverage the unlabeled ad candidates data, particularly the ones with similar distribution as the inference data. One difficulty with this model training strategy is determining how to effectively use these unlabeled data points, which have more consistent distribution as compared to the model inference data. In this work, we have leveraged various state-of-the-art (SoTA) methods to incorporate unlabeled data in training Retrieval models. Additionally, we developed a modified version of UDA (MUDA) to improve the performance of naive implementation of UDA in the Retrieval model training. Our online experimental results show that a couple of methods could potentially improve the performance of the ads ranking system, as compared to the knowledge distilled model in the current production environment and a few other methods. Thus, our contribution could be summarized as the following:

- We identified and characterized the selection bias issue in the upper funnel of the multi-cascade advertisement recommendation system.

- We surveyed a series of SoTA modeling strategies and evaluated their performance in both offline and online settings.

- We further proposed a modified version of Unsupervised Domain Adaptation (MUDA) that provides the best online performance among all the modeling strategies we have examined, and the online experiments show that MUDA also outperforms the current production model.

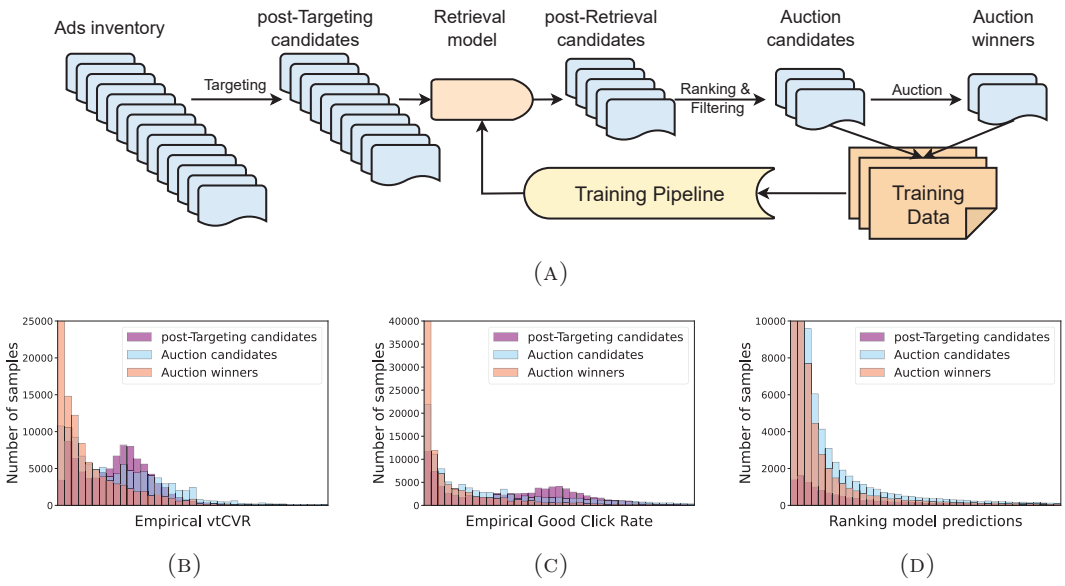


FIGURE 6.2: Distribution of features and labels across three ads datasets related to Retrieval modeling. (a) shows the flow of major ad candidates along the ads delivery funnel. (b) shows the distribution of Empirical vtCVR (one of key Retrieval model's features) across three datasets for Retrieval training/serving. (c) shows the distribution of Empirical Good Click Rate (one of key Retrieval model's features) across three datasets for Retrieval training/serving. (d) shows the distribution of the Ranking model predictions (used as the pseudo label in Retrieval model training) across three datasets. Note that the exact values on x-axes are hidden for confidentiality reasons.

6.2 BIAS IN PINTEREST ADS

As illustrated in Figure 6.1, Pinterest's ads serving system consists of four stages: Ads Targeting, Ads Retrieval, Ads Ranking, and Ads Auction. Each stage scores and/or filters ad candidates based on the request and ads content features. Given an ad request, Ads Retrieval narrows down millions of ad candidates to a couple of thousands. These candidates are then sent to Ads Ranking for further accurate prediction of user action as well as filtering. Finally we run Ads Auctions on survivors and determine auction winners based on a predefined utility function and advertiser's bid.

In the Retrieval stage, the latency limit is crucial because of the large number of ad candidates in the database. We adopt a two-tower DNN structure [25], where candidate embedding could be computed offline. During serving, the model will produce the score of each ad candidate by calculating the dot-product between the precomputed candidate embedding and the query embedding computed on-the-fly for each request.

6.2.1 Datasets and Training Pipeline

As mentioned earlier, the ads serving system consists of Targeting, Retrieval, Ranking, and Auction. As shown in figure 6.2a, millions of candidates in the ads inventory will flow through various stages across the ads delivery funnel, and only a small set of valuable ads will survive and be delivered to users. Specifically, the initial ads inventory candidates will be selected through Ads Targeting to refine the set of ad candidates (a.k.a

post-Targeting candidates), which will then be scored and ranked by Retrieval models. After being selected by Retrieval models, the survivors (post-Retrieval candidates) will be further filtered or selected by various business logics and models in the Ranking stage. This leads to a new set of ad candidates (a.k.a Auction candidates), which will be evaluated in the Auction stage. The Auction stage will pick a dozen of winners out of the auction candidates and deliver these final survivors (a.k.a Auction winners) to Pinterest's users. For existing Retrieval models, two types of training data are collected — Auction candidates and Auction winners. The latter dataset includes observed user actions as true labels, and the former one includes ranking model predictions as pseudo labels.

The Ranking model predictions are used in the Auction stage to determine the winning ads from the auction candidates pool. Currently, we use these Ranking model predictions as pseudo labels to train Retrieval models, with the aim to maximize the funnel efficiency to deliver the most valuable ad candidates to Pinterest users. To ensure the model freshness, Retrieval models are continuously trained and evaluated on a daily basis. Specifically, the model snapshot trained on day $X-1$ data is loaded to train on day X data, and the newly trained model is evaluated on day $X+1$ data. This daily training setup enables the model to capture the most recent patterns, keeping it responsive to new trends. The second-day evaluation allows for detection of possible overfit and abnormal behavior before serving production traffic.

6.2.2 Selection Bias

As mentioned above, Retrieval models are currently trained on both Auction candidates and Auction winners, where the Ranking model predictions are used as pseudo labels. Such setup inevitably introduces the data with selection bias, particularly the inconsistency in the dataset between training and serving [79]. In serving time, however, the model needs to make predictions on the post-Targeting ad candidates. As Auction candidates and winners are a small subset of post-Targeting candidates (generated through various business logics and Ranking models), the distribution of these datasets will be inconsistent between model training and serving.

Figure 6.2a illustrates the concept of inconsistency on the ads datasets used in training and inferencing in the cycle of the Retrieval models. To further demonstrate the bias, we analyzed the distributions of pseudo labels and two important Retrieval model features across three different datasets: post-Targeting candidates, Auction candidates, Auction winners. Figure 6.2b, 6.2c, and 6.2d demonstrates that the distributions are different across all three datasets, and this distribution difference is much more significant between the two datasets used in current Retrieval model training and the one used in Retrieval model serving.

For simplicity, in the rest of this work, we will interchangeably use the following terms: post-Targeting candidates, serving datasets for Retrieval models, and unbiased dataset.

6.2.3 Problem Formulation

For simplicity, we represent each data record as a tuple of three elements: (u, a, y):

- u: the feature of a request, containing user profile features and context features (e.g., search term if from Search surface),
- a: the advertisement candidate features,
- y: the groundtruth label, i.e., observed user actions.

Additionally, let $<\mathbb{U}, \mathbb{A}>$ represent distribution of request features, advertisement features in inventory and $\mathbb{D} = \mathbb{U} \times \mathbb{A}$ represent the full distribution of all request and ad candidates pairs. Finally, let F_θ and l represent the model with trainable parameter θ and the loss function we want to minimize:

- $\mathrm{F}(u, a) \rightarrow \mathbb{R}$: a model maps the request and candidate features to a numeric value,
- $l(y, y) \rightarrow \mathbb{R}$: a function maps two numeric values to a scalar (the loss value).

Ideally we want to minimize the training loss on unbiased data, i.e.,

$$\min_{\theta} \mathcal{L}_{ideal}(\mathrm{F}_\theta) = \frac{1}{|\mathbb{D}|} \sum_{(u,a)\in\mathbb{D}} l(y, \mathrm{F}_\theta(u, a)). \tag{6.1}$$

In reality, it is impossible to calculate the above loss function on the unbiased dataset, as the true labels are not available. As a result, we have to leverage the biased dataset

whose true labels are available to us. In the next section, we will describe a series of methods to use both biased and unbiased datasets to build a model to score the post-Targeting ad candidates in our system.

6.3 Solution

6.3.1 Naive Method: Binary Classification

The naive method is to train a simple classification model in the common way, i.e., training a click classification model based on the dataset with observed user actions, where the ones with user clicks are treated as positive examples and the ones with no clicks are treated as negatives. In this naive method, we will optimize the following loss function:

$$\min_{\theta} \mathcal{L}_{naive}(\mathrm{F}_\theta) = \frac{1}{|\mathbb{O}|} \sum_{(u,a)\in\mathbb{O}} l(y, \mathrm{F}_\theta(u,a)). \tag{6.2}$$

The dataset $\mathbb{O}$ denotes the set of request and auction winners pairs, where there are observed user actions.

6.3.2 In-batch Negative Classification

Similar to the naive classification method, we will build a classification model based on the biased dataset with observed user actions as the true labels. In the real-world

advertising system, the viewed ads without user clicks are not necessarily reliable negative examples, Users could still find these ads to be valuable even if they did not take actions on them at that moment. Different from the naive classification method, we generate negative examples by introducing ad candidates from the other requests in the same training batch as the current request following the common setup [35, 45, 84, 85]. Specifically, only the delivered ads with user clicks are included in training data, and clicked ads in different requests in the same batch are treated as negative examples.

6.3.3 Knowledge Distillation

Ranking models are trained with complex architectures and numerous input features. In contrast, Retrieval models have to limit the architecture to two-tower DNN as well as available features due to the demanding requirement of scalability and low serving latency. To minimize the performance loss, knowledge distillation (kd) [37] is adopted, which means Retrieval models are trained with Ranking model's predictions as pseudo labels. Formally, with R denoting the Ranking model, we optimize the following loss function:

$$\min_{\theta} \mathcal{L}_{kd}(\mathrm{F}_\theta) = \frac{1}{|\mathbb{O}|} \sum_{(u,a)\in\mathbb{O}} l(\mathrm{R}(u,a), \mathrm{F}_\theta(u,a)). \tag{6.3}$$

6.3.4 Transfer Learning

The core idea of transfer learning is to train a model on source domain data and then fine tune part of its parameters on the target domain. Particularly for a DNN model, the

early layers are usually fixed during fine tuning as they are shown to represent primitive and general features [59]. In our case, the Retrieval model is a two-tower DNN, and the data distribution discrepancy across different datasets is only from the ad candidates. As a result, we use the unbiased data to fine tune the ad's embedding tower, and keep the query tower unchanged.

6.3.5 Adversarial Regularization

Another view of bias issue is that the representation learned from biased data is not general enough to be applied to the unbiased dataset, leading to a performance degradation. We can therefore add regularization on the learning so that the intermediate output of the model has no information indicating its data source, a technique known as Adversarial (adv) Learning [36].

For a DNN model, we can split it into two parts. The former one takes the raw input and gives the intermediate output. The latter one takes the intermediate output and gives the final prediction. The Adversarial regularization trains a data source classifier from the intermediate output, the negative of whose loss function is then added to the original one as regularization.

Formally, let F_1 and F_2 denote such two parts of DNN, while the H denotes the classifier. The loss function of data source classifier is defined as Equation (6.4).

$$\mathcal{L}_{cls}(u,a) = -\mathbf{1}_{(u,a)\in\mathbb{D}} \log \mathrm{H}(\mathrm{F}_1(u,a)) - \mathbf{1}_{(u,a)\in\mathbb{O}} \log\left[1 - \mathrm{H}(\mathrm{F}_1(u,a)\right] \tag{6.4}$$

The final loss function for adversarial regularization is shown in Equation (6.5):

$$\mathcal{L}_{adv} = \mathcal{L}_{target}(\mathrm{F}_2(\mathrm{F}_1(u, a)), y) - \lambda \mathcal{L}_{cls}(u, a), \tag{6.5}$$

where the $\mathcal{L}_{target}$ is the original loss function that trains the target model and the λ is hyper parameter weighting the regularization.

The goal is to minimize the $\mathcal{L}_{adv}$ with regarding to F_1, F_2 and the $\mathcal{L}_{cls}$ with regards to H.

6.3.6 Unsupervised Domain Adaptation

UDA (Unsupervised Domain Adaptation) is a technique to train a model that works well on the target domain with unlabeled data by only using labeled samples on the source domain. UDA method has been applied to the situation where the feature distribution and the data labeling are different between the source and target domains. In Pinterest Ads system, the source domain is the biased dataset with labels and the target domain is the unbiased dataset without labels. As a result, this data selection bias could be formulated as a UDA problem[83].

6.3.6.1 Naive UDA

The naive method is to directly train the model on the unbiased dataset so that there will be no inconsistency between training and serving. As the ground truth labels of

the unbiased dataset are missing, the pseudo labels will be generated from a separate model that is trained on the biased dataset from the source domain where the ground truth label is available. Following the same annotation scheme as above, let R denotes the Ranking model that is used to generate the pseudo labels for the unbiased dataset from source domain. The optimization goal becomes the following:

$$\min_{\theta} \mathcal{L}_{naiveUDA}(\mathrm{F}_\theta) = \frac{1}{|\mathbb{D}|} \sum_{(u,a)\in\mathbb{D}} l(\mathrm{R}(u,a), \mathrm{F}_\theta(u,a)), \tag{6.6}$$

where $\mathbb{D}$ is the data in the source domain.

However, this method has many drawbacks. In reality, the unbiased data is only a sampling, and the volume is small due to infra cost. This might lead to performance degradation. Additionally, the high-quality candidates might not be sufficiently representative in this training data from the source domain. We will discuss the performance in the experiment section.

6.3.6.2 Modified UDA

In UDA, the quality of pseudo labels is critical to the performance of trained models. In the above naive UDA, there is no mechanism to guarantee the quality of the pseudo labels, especially when the pseudo label generating model remains not sufficiently accurate. Previously, Saito et al. [67] proposed to use an asymmetric tri-training method where two separate pseudo label generating models are used as the mechanism to ensure the pseudo label quality. However, the requirement to maintain a second pseudo

label generating model with reasonable performance will be too costly for the real-world advertising system where tens or even hundreds of Retrieval models are needed and retrained on a daily basis. Additionally, it will be inhibitively costly when a pseudo label has to be derived from a set of models, and then a second set of several models will be required to be developed and maintained to leverage the tri-training method.

To address the pseudo label quality issue for real-world ads retrieval, we transform the original numeric pseudo label (prediction of Ranking model) to a binary classification label based on carefully chosen thresholds. Formally, let δ_l and δ_h denote the two thresholds with $\delta_l < \delta_h$. As shown in Equation 6.8, numeric pseudo labels lower than the first threshold are treated as negative, those higher than the second threshold are treated as positive. Data records with numeric pseudo labels falling between these two thresholds are removed from the training dataset.

The rationale behind this is to only keep the records that the Ranking model is confident about and discard the ones that are close to the hyperplane of the Ranking classifier. Now, the optimization goal of training the Retrieval model becomes the following:

$$\min_{\theta} \mathcal{L}_{MUDA}(\mathrm{F}_\theta) = \frac{1}{|\mathbb{O}|\cdot|\mathbb{D}|} \sum_{(u,a)\in\mathbb{O}\cup\mathbb{D}\wedge(\mathrm{R}(u,a)\leq\delta_l\vee\mathrm{R}(u,a)\geq\delta_h)} l(\Phi_{\delta_l}^{\delta_h}(\mathrm{R}(u,a)), \mathrm{F}_\theta(u,a)) \quad (6.7)$$

where $\Phi_{\delta_l}^{\delta_h}(\cdot)$ is a pseudo classification label indicator, converting ranking model predictions to binary label according to given thresholds, as shown in the equation below:

$$\Phi_{\delta_l}^{\delta_h}(y) = \begin{cases} 1, \text{ if } y \geq \delta_h \\ \\ -1, \text{ if } y \leq \delta_l \end{cases} . \tag{6.8}$$

To select the thresholds, we adopt a data driven method. Particularly, we bucketize the Ranking model prediction and check the corresponding empirical click rate for each bucket. Thresholds are chosen when there is sudden change of empirical click rates.

6.4 Experiments and Results

In this section, we will first describe the model training details and introduce the evaluation settings and metrics. We will then present and discuss the results from offline and online experiments to compare the performance of the proposed solutions.

6.4.1 Datasets

As described in Section 6.2.1, the two existing training data sources are the Auction candidates and Auction winners (both are biased datasets). In Section 6.2, we introduced an unbiased dataset randomly sampled from the post-Targeting dataset. This unbiased dataset is required to be scored and ranked by Retrieval models in the production system. Taking into consideration the infrastructure cost and the volume of the resulting

dataset, we sample 100,000 queries and 6,000 advertisement candidates for each query to create the unbiased dataset every day.

6.4.2 Experimental setting

To examine the performance of de-biasing methods on the Pinterest's ads dataset, we implement the models and conduct systematic experiments to collect evaluation results on the real-world production system. The binary classification models are trained on Auction winners with real user actions as the true labels, which aims to provide supplemental evidence to indicate the reason why the current production model is not directly trained with real user actions. The following describes the details of the baseline models:

- **Binary Classification**: Since the regression model is trained on the pseudo labels generated by the Ads Ranking models, the performance of the classification model directly trained on the user actions is worth examining. To train this model, we use the Auction winner as the training dataset with labels defined in Section 6.3.1 and binary cross entropy (BCE) as the loss function.

- **In-batch Negative Classification**: We also train a classification model with in-batch negative sampling which uses other candidates from the same batch of data as negative samples for a given query. We use 1000 as the batch size and use the batch size as the number of hard negatives in the loss function. The model is trained with only the Auction winner dataset, and we only use the candidates with user clicks.

- **Knowledge Distillation**: In the current production model's training, we use the Ads Ranking model's output as the pseudo label and mean absolute logarithmic error (LogMAE) as the loss function. For the production model, the training dataset includes the Auction candidates and Auction winners. Besides this production model, we also train another one with only Auction winners. In evaluation, we refer to the first one as the Production model and the second one as the knowledge distillation model.

We summarize the implementation details of the debiasing model as the following:

- **Transfer Learning**: For the transfer learning model, we use both the biased and unbiased dataset. We also use Ranking model predictions as the pseudo labels and LogMAE as the loss function to train the Retrieval model.
- **Adversarial Learning**: For the adversarial learning model, we implement the data source discriminator as a one-layer MLP with sigmoid as the activation function. Both the biased and unbiased datasets are used to train the Retrieval model, and the Ads Ranking model is used to generate pseudo labels for the training datasets.
- **Naive Unsupervised Domain Adaptation (UDA)**: To train the naive UDA model, we only use the unbiased dataset with pseudo labels generated from the Ads Ranking model predictions and LogMAE as the loss function.

- **Modified Unsupervised Domain Adaptation (MUDA)**: Here we use the unbiased dataset with pseudo labels derived as discussed in Section 6.3.6.2 by transforming the Ranking model predictions into binary classes and BCE for the loss function.

For model training hyper-parameters, we use 6144 as the batch size and 0.0001 as the learning rate unless defined specifically. In the two-tower model, we use four fully connected layers, and the final layer's output dimension is 32. We use `sigmoid` as the activation function for the output layer and use `selu` [46] for the other layers.

6.4.3 Evaluation Metrics

For offline evaluation metrics, we use AUC-ROC score for both the classification and regression models. We evaluate the models on one day of the Auction winners dataset. For online A/B experiments, we compare these models to the production model and report the change of total impressions numbers (ΔIMP), click through rate (ΔCTR), and 30 seconds click through rate (ΔgCTR30)

For ads evaluation, besides the user-side metrics mentioned above, we also report metrics that relate to advertiser experience. These metrics are:

- **Impression to Conversion Rate Ratio (iCVR)** measures the effectiveness of an ad campaign in converting impressions into conversions.
- **Cost per Action (CPA)** measures the cost to the advertiser for each positive user action and is currently exclusively applied to the conversion ads.

Due to information confidentiality, we only report the lift of these metrics compared to the current production model.

6.4.4 Offline Evaluation

Models	AUC-ROC
Production Model	0.895
Binary Classification	0.895
In-batch Negative	0.701
Knowledge Distillation	0.896
Transfer Learning	0.890
Adversarial Learning	0.896
Naive UDA	0.841
MUDA	0.844

TABLE 6.1: AUC-ROC on evaluation dataset. The models such as knowledge distillation, adversarial learning, binary classification trained with Auction Winners dataset usually have better offline evaluation results.

For offline evaluation, we evaluate both the regression and classification models using AUC-ROC. For the evaluation dataset, we use the Auction winners which contain real user clicks. As shown in Table 6.1, compared to the production model, the models such as knowledge distillation, transfer learning, binary classification, and adversarial models have similar performance in terms of AUC-ROC score. The results are expected because the training datasets include the Auction winners for these models. For the in-batch negative model, it is trained with only the positive candidates in the Auction winners dataset, so it does not perform well in the offline evaluation because the negative candidates were not included in the training dataset. For Native UDA and MUDA, both models are trained with only the post-Targeting datasets, and the feature distribution discrepancy (Figure 6.2a) leads to the lower performance than other models.

To summarize, the offline evaluation is as expected because we see models trained and evaluated on the same source of data have better performance than those trained with different sources of data. However, the offline evaluation could not necessarily reflect the true model performance in the production system especially when the serving data used in the online experiments are from a completely different distribution. Thus, in the following sections, we conduct systematic online A/B experiments to compare the performance of aforementioned models.

6.4.5 Online A/B Experiments

6.4.5.1 Overall evaluation

Table 6.2 shows the overall online evaluation results of all models. Among the metrics, we will focus on the change of gCTR30 as our models are optimized towards this objective. The binary classification model has decreased gCTR30, indicating a significant drop in the quality of user engagement with the recommended ads. It also shows the largest decrease in CTR and highest increase in impressions, which means that while more ads were delivered to users, fewer of them got clicked. In contrast, the in-batch negative and knowledge distillation models have positive changes in gCTR30. However, the decrease in impression could be the main reason for the gCTR30 increase because less ads were shown to the users.

Although all three models (binary classification, in-batch negative and knowledge distillation) suffer from selection bias in the training dataset, the latter two perform better.

In the binary classification model's training, the negative candidates are always from the same query; whereas the in-batch negative classification model is trained with random sampled candidates of different queries within the batch. The difference between the source of negative candidates provides the model with more diverse and informative training data, which results in not overfitting to the specific query. For knowledge distillation, the training data labels are the Ranking model predictions, whose values contain richer information than raw binary click-or-not labels.

Models	ΔIMP	ΔCTR	ΔgCTR30
Binary Classification	0.95%	-5.51%	-12.66%
In-batch Negative	-2.25%	4.45%	4.68%
Knowledge Distillation	-3.26%	0.25%	5.97%
Transfer Learning	0.43%	-1.88%	-4.35%
Adversarial Learning	0.28%	-0.45%	-0.66%
Naive UDA	0.45%	-3.05%	-4.80%
MUDA	0.92%	0.47%	5.07%

TABLE 6.2: Online lifts of impression (IMP), click-through rate (CTR), and good long click (gCTR30) observed with various models on all types of ads. Both in-batch negative and knowledge distillation methods improve gCTR30 at the cost of impression drop, and MUDA is the only method to recommend more ads with higher quality, as observed by the increased gCTR30 without impression drop.

The transfer learning model had a small increase in impressions, but also had a negative change in gCTR30. With the warm start weights, the transfer learning model has similar results with the decrease in user engagements. In our case, the problem of the transfer learning model is the fine tuning on the unbiased dataset. Candidates in unbiased dataset are randomly sampled for each query, where high quality ones might be underrepresented.

The adversarial model has similar results with a decrease in gCTR30 and a slight increase in impressions. Compared to the transfer learning model, the adversarial model has

better performance in user engagements. In adversarial model, the classifier serves as a regularizer to prevent the embedding tower from learning a domain specific embedding for a certain log source. Unlike the transfer learning model, the debiasing technique in the adversarial model does not rely on the quality of the unbiased training data. The training of the classifier is unsupervised because we use the log source (i.e. Auction winner, Auction candidates) as the ground truth label. When we are able to successfully train a classifier to classify the log source, the classifier could be used as an adversarial regularizer to help train an unbiased embedding model. However, compared to the production model, the decrease in gCTR30 may indicate that the restriction on the embedding learning makes the model drop the information that are critical in online evaluations.

The Naive UDA model has an average performance compared to the other baseline models. The Naive UDA model is trained on the unbiased dataset which contains the pseudo label generated from the Ranking model. The reason why the Naive UDA model performs badly is similar to the reason why the transfer learning model performed poorly. Since the unbiased dataset is collected by random sampling of post-Targeting ad candidates in addition to the existing queries, these sampled candidates are mostly negative samples, which does not help to train a good Retrieval model.

In contrast, the Modified UDA (MUDA) model has a much higher gCTR30 than the production model. When the number of impressions increases, the higher user engagement suggests that the MUDA model delivers more ads with higher quality to users. Compared to the Naive UDA model, the MUDA model transforms numerical pseudo

labels generated by the Ranking model into binary classes determined by certain thresholds. The model also uses BCE loss. The lift in user engagement metrics suggests that such label transformation improves the quality of pseudo labels used in MUDA. By transforming the numerical pseudo labels to binary ones, we prevented the model from overly fitting into the Ranking model's prediction of every single candidate, but to rank those on which the Ranking model has high confidence.

6.4.5.2 Evaluation by ads objective type

Models	Awareness			Traffic			Web Conversion		
	ΔIMP	ΔCTR	ΔgCTR30	ΔIMP	ΔCTR	ΔgCTR30	ΔIMP	ΔCTR	ΔgCTR30
In-batch Negative	-8.70%	2.41%	13.74%	1.03%	1.16%	2.56%	1.31%	1.69%	0.39%
MUDA	0.32%	2.71%	1.97%	0.43%	-4.28%	-3.07%	3.15%	5.19%	8.88%

TABLE 6.3: Online lifts of impression (IMP), click-through rate (CTR), and good long click (gCTR30) observed with two promising models on each type (awareness, traffic, web-conversion) of ads. In-batch negative classification model works better on the traffic ads, and MUDA model helps web-conversion ads the most.

In overall evaluation, the in-batch negative and MUDA are two methods that demonstrate promising metrics. In Table 6.3 we show the evaluation results of the two methods broken down by different ads objective types, i.e., awareness, traffic and web conversion ads. Awareness ads aim to increase the visibility of a brand or product. Analyzing the performance of awareness ads helps understand the effectiveness of a brand's marketing strategy. As shown in Table 6.3, the in-batch negative model has significant increase on gCTR30 compared to other models for awareness ads. However such a boost might be due to the huge decrease in impressions. In-batch negative model is trained only on candidates with user long clicks. The awareness ads essentially have a lower chance of being clicked than other types, since its main goal is to increase the visibility of a brand.

As a result, the in-batch negative model could bias towards other ads types, leading to the huge impression drop of awareness ads.

Traffic ads are designed to drive traffic to a specific website or landing page. They are typically used to increase brand awareness, generate leads, or drive sales. By analyzing metrics such as CTR and gCTR30, businesses can determine whether their ads are resonating with their target audience and whether they are successfully achieving their advertising goals. As can be seen, the in-batch negative model is the only ones that yield an increase in both ads impression and gCTR30. Traffic ads aim to attract users to click and could occupy a big portion of records with positive user actions. Therefore the in-batch negative model's training dataset, which only includes candidates with positive user actions, may have a higher proportion of candidates that are well-suited for driving traffic. As a result, the model could better identify candidates that are likely to drive traffic, resulting in an improvement in the gCTR30 metric for traffic ads.

Web-conversion ads aim to drive users to take a specific action on a website, such as making a purchase. These ads can provide insight for measuring the success of an online advertising campaign. As shown in the Table 6.3, the MUDA model favors the web-conversion ads objective type, as it has the highest improvement in CTR and gCTR30 among all models for this objective type. The in-batch negative model also performs well for web-conversion ads, with improvements in both CTR and gCTR30. The MUDA may favor web-conversion ads because pseudo labels generated by the Ads Ranking model may favor web-conversion ads as they are designed to attract users to stay on target websites longer for potential conversion behaviors. Additionally, the threshold selection

strategy used in the MUDA model may be more effective at identifying high-quality candidates for web-conversion ads, which could also contribute to its better performance for this type of ad.

6.4.5.3 Conversion ads

Models	ΔiCVR	ΔCPA
In-batch Negative	-2.55%	1.11%
MUDA	1.89%	-4.40%

TABLE 6.4: Online metrics performance of in-batch negative classification and MUDA models on web-conversion ads. In-batch negative classification model leads to lower conversion probability on each ads impression (iCVR) and thus has a higher CPA cost to advertisers. In contrast, MUDA model recommended ad candidates with higher conversion rate and therefore a lower CPA cost.

In Table 6.4, we show the performance of in-batch negative and MUDA models with regard to conversion related metrics as these two show good performance on web-conversion ads. In general, the in-batch negative model has a decreased iCVR and increased CPA. This is not favorable for the advertiser as it increases their costs as measured by CPA. On the other hand, the MUDA model shows an opposite result with increased iCVR and decreased CPA, reducing the ads campaign cost to advertisers. These metrics indicate that while both increase the long clicks, MUDA model performs much better by generating more conversions out of these increased long clicks.

One reason why the MUDA model performs better is that the model could improve the performance of identifying the high-quality candidates that are more likely to lead to conversions, and thus decrease the cost per action for advertisers. Additionally, the fact that the MUDA model is trained on only unbiased data with pseudo labels generated

from the Ads Ranking model could have an impact. The pseudo labels may capture more relevant information about the users' behaviors and preferences, leading to better performance in terms of CPA.

6.4.6 Variants of MUDA

In the MUDA method, we believe different threshold selection mechanisms could impact the quality of binary pseudo labels. As a result, we further investigate the impact of different thresholding mechanisms on the performance of trained Retrieval models. In the unbiased dataset, we first bucketize the candidates according to their numerical pseudo labels (gCTR30) predicted by the Ranking model, where we compute the percentile of the labels and use the adjacent percentile to create buckets. Then for each bucket, we adapt the following two strategies to calculate empirical gCTR30:

- compute the gCTR30 for candidates with the real user actions,
- divide the number of true good clicks by the number of candidates in the bucket.

Models	ΔIMP	ΔCTR	ΔgCTR30
MUDA v1	-0.07%	11.26%	30.78%
MUDA v2	0.56%	3.52%	13.04%
MUDA v3	0.92%	0.47%	5.07%

TABLE 6.5: Online lifts of impression (IMP), click-through rate (CTR), good long click (gCTR30) observed with various MUDA variants on all types of ads. MUDA v1 achieves the highest gain on ads engagement (both CTR and gCTR30), and MUDA v3 achieves the most balanced gain across different metrics with good gCTR30 and impression lift.

We select the threshold by determining the elbow point of the graph. For example, when there is a sudden drop of true good clicks or good clicks rate between two adjacent bins, we use one of the bins as the negative threshold. This means, in the label transformation, we treat candidates with pseudo labels smaller than the threshold as the negative samples. For positive labels, we check if there is a sudden increase of true good clicks or good clicks rate between the bins. To study different threshold selection strategy, we propose three variants of the MUDA models:

- v1: We train the MUDA model on both the biased and unbiased datasets with the first threshold selection strategy.
- v2: We train the MUDA model on only the unbiased datasets with the first threshold selection strategy.
- v3: We train the MUDA model on only unbiased datasets with the second threshold selection strategy.

Models	Awareness			Traffic			Web Conversion		
	ΔIMP	ΔCTR	ΔgCTR30	ΔIMP	ΔCTR	ΔgCTR30	ΔIMP	ΔCTR	ΔgCTR30
MUDA v1	-2.13%	2.77%	7.97%	-5.22%	0.47%	17.34%	12.98%	21.52%	29.63%
MUDA v2	0.10%	1.72%	5.97%	-1.53%	-2.69%	1.18%	5.16%	11.14%	17.83%
MUDA v3	0.32%	2.71%	1.97%	0.43%	-4.28%	-3.07%	3.15%	5.19%	8.88%

TABLE 6.6: Online lifts of impression (IMP), click-through rate (CTR), and good long click (gCTR30) observed with MUDA variants on each type (awareness, traffic, web-conversion) of ads, where MUDA v3 shows best balanced impression gains among them.

Table 6.5 shows the overall performance of three variants of the UDA models, as measured by several evaluation metrics: impression, and click-through rate (CTR). At first glance, the v1 model may seem to work best, hugely increasing user engagement while

keeping the impression neutral. However, if broken down by ad types, this mode actually leads to a large impression shift from awareness (-2.13%) and traffic ads (-5.12%) toward web conversion ads (+12.98%), as shown in Table 6.6. This observation may indicate that training UDA models on biased data would make the model favor web-conversion ads more than others. Comparing v2 and v3 models, the latter one shows a better balanced impression gains across all ad objective types. It could be due to the second strategy of calculating the approximate gCTR30 for the unbiased dataset. This strategy may better represent the true performance of the candidates and result in a more accurate threshold selection, leading to improved performance in the MUDA model.

Models	ΔHDR	ΔRPR
MUDA v1	4.80%	13.88%
MUDA v2	6.35%	4.43%
MUDA v3	-2.81%	1.43%

TABLE 6.7: Online lifts of ads hide rate (HDR), re-pin rate (RPR) observed with MUDA variants on all types of ads. MUDA v3 achieves the most balanced performance with fewer ads being hidden and more ads being repined by the users.

To better understand the performance of these MUDA variants, we also measure their online performance on two other useful engagement metrics: hide rate (HDR) and re-pin rate (RPR). Note that re-pin is a user action indicating if the user saves an ad to a Pinterest board. In Table 6.7, we show the change of the two metrics compared to the production model. Although the RPR is increased, both MUDA v1 and v2 models recommend more ads that will be hidden by users, suggesting some of the recommended ads from these models do not provide a good user experience. In contrast, MUDA v3 model generally has the most balanced improvement across all metrics, which shows

positive lift in the user engagement and reduction in the unwanted user experience (HDR).

6.5 Conclusion

In conclusion, this work has analyzed the impact of selection bias in Pinterest's online advertising system. We propose and evaluate several debiasing methods to mitigate the negative impacts of selection bias on recommender's performance. The results of our experiments show that our proposed methods, specifically the MUDA model, can effectively improve the performance of advertising systems by handling the selection bias. Additionally, our online experiment shows that this model also improves the cost efficiency of the ad campaigns. These findings demonstrate the importance of addressing selection bias in recommendation systems and provide valuable insights for practitioners in this field.

Chapter 7

Conclusion

The conclusions of the presented works collectively highlight significant strides in addressing fairness within ranking and search systems, alongside mitigating selection bias in online advertising platforms. Through innovative approaches like the Meta-learning based Fair Ranking (MFR) and the Meta Curriculum-based Fair Ranking (MCFR) frameworks, we have demonstrated the potential to significantly improve fairness metrics and minority group exposure by re-weighting training losses and employing meta-learning techniques with curriculum learning. These methods have shown promising results in real-world datasets, underscoring their effectiveness over traditional fair ranking models. Furthermore, our exploration into Large Language Models (LLMs) has uncovered biases that challenge fairness, prompting the development of fine-tuning strategies such as LoRA to foster more equitable outcomes in ranking tasks. Our research also

delves into the issue of selection bias in Pinterest's multi-cascade advertising recommendation system, presenting debiasing methodologies like the Modified Unsupervised Domain Adaptation (MUDA) model, which not only enhances recommendation system performance but also boosts ad campaign cost-efficiency.

Future directions for this body of work include refining meta-dataset collection methods for meta-learning, expanding the applicability of fairness frameworks to accommodate multiple protected attributes, and exploring diverse ranking tasks and datasets. Moreover, efforts will focus on balancing accuracy and equity in LLM applications through improved ranking performance and fairness strategies. Additionally, the insights garnered from mitigating selection bias in online advertising systems pave the way for further innovation in addressing biases across recommendation systems, contributing to the broader discourse on fairness and transparency in machine learning and AI applications.

Bibliography

[1] Abubakar Abid, Maheen Farooqi, and James Zou. Persistent anti-muslim bias in large language models. In Marion Fourcade, Benjamin Kuipers, Seth Lazar, and Deirdre K. Mulligan, editors, *AIES '21: AAAI/ACM Conference on AI, Ethics, and Society, Virtual Event, USA, May 19-21, 2021*, pages 298–306. ACM, 2021.

[2] Abubakar Abid, Maheen Farooqi, and James Zou. Large language models associate muslims with violence. *Nature Machine Intelligence*, 3:461–463, 06 2021. doi: 10.1038/s42256-021-00359-2.

[3] Marcin Andrychowicz, Misha Denil, Sergio Gómez Colmenarejo, Matthew W. Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, page 3988–3996, Red Hook, NY, USA, 2016. Curran Associates Inc.

[4] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your MAML. In *International Conference on Learning Representations*, 2019.

[5] Abolfazl Asudeh, H. V. Jagadish, Julia Stoyanovich, and Gautam Das. Designing fair ranking schemes. In *Proceedings of the 2019 International Conference on Management of Data*, page 1259–1276, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450356435.

[6] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022.

[7] Matias Barenstein. ProPublica's COMPAS Data Revisited. *arXiv e-prints*, art. arXiv:1906.04711, Jun 2019.

[8] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, page 41–48, New York, NY, USA, 2009. ACM.

[9] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H. Chi, and Cristos Goodrow. Fairness in recommendation ranking through pairwise comparisons. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 2212–2220, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016.

[10] Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. Equity of attention: Amortizing individual fairness in rankings. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, page 405–414, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356572.

[11] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[12] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[13] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott M. Lundberg, Harsha Nori, Hamid Palangi, Marco Túlio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with GPT-4. *CoRR*, abs/2303.12712, 2023.

[14] Chumphol Bunkhumpornpat, Krung Sinapiromsaran, and Chidchanok Lursinsap. Dbsmote: density-based synthetic minority over-sampling technique. *Applied Intelligence*, 36:664–684, 2012.

[15] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, page 89–96, New York, NY, USA, 2005. ACM.

[16] Christopher JC Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11(23-581):81, 2010.

[17] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136, 2007.

[18] L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. Ranking with fairness constraints. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107, pages 28:1–28:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[19] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1): 321–357, jun 2002. ISSN 1076-9757.

[20] Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang. Autodebias: Learning to debias for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 21–30, New York, NY, USA, 2021. Association for Computing Machinery.

[21] Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang. Autodebias: Learning to debias for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 21–30, New York, NY, USA, 2021. Association for Computing Machinery.

[22] Myra Cheng, Esin Durmus, and Dan Jurafsky. Marked personas: Using natural language prompts to measure stereotypes in language models. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1504–1532. Association for Computational Linguistics, 2023.

[23] Xiangxiang Chu, Bo Zhang, and Ruijun Xu. Fairnas: Rethinking evaluation fairness of weight sharing neural architecture search. In *International Conference on*

Computer Vision, 2021.

[24] Daniel Cohen, Bhaskar Mitra, Katja Hofmann, and W. Bruce Croft. Cross domain regularization for neural ranking models using adversarial learning. In *The 41st International ACM SIGIR Conference on Research Development in Information Retrieval*, SIGIR '18, page 1025–1028, New York, NY, USA, 2018. Association for Computing Machinery.

[25] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, page 191–198, New York, NY, USA, 2016. Association for Computing Machinery.

[26] Nick Craswell, Arjen P De Vries, and Ian Soboroff. Overview of the trec 2005 enterprise track. In *Trec*, volume 5, pages 1–7, 2005.

[27] André Cruz, Catarina G Belém, João Bravo, Pedro Saleiro, and Pedro Bizarro. FairGBM: Gradient boosting with fairness constraints. In *The Eleventh International Conference on Learning Representations*, 2023.

[28] Michael D Ekstrand, Anubrata Das, Robin Burke, Fernando Diaz, et al. Fairness in information access systems. *Foundations and Trends® in Information Retrieval*, 16(1-2):1–177, 2022.

[29] Michael D. Ekstrand, Graham McDonald, Amifa Raj, and Isaac Johnson. Overview of the trec 2021 fair ranking track. In *The Thirtieth Text REtrieval Conference (TREC 2021) Proceedings*, 2022.

[30] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, page 1126–1135. JMLR.org, 2017.

[31] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 06–11 Aug 2017.

[32] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pages 1568–1577. PMLR, 2018.

[33] Batya Friedman and Helen Nissenbaum. Bias in computer systems. *ACM Trans. Inf. Syst.*, 14(3):330–347, jul 1996.

[34] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *Findings*, 2020.

[35] Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 528–537, Hong Kong, China, November 2019. Association for Computational Linguistics.

[36] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[37] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021.

[38] Cyril Goutte and Eric Gaussier. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In David E. Losada and Juan M. Fernández-Luna, editors, *Advances in Information Retrieval*, pages 345–359, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[39] Fabian Haak and Philipp Schaer. Auditing search query suggestion bias through recursive algorithm interrogation. In *14th ACM Web Science Conference 2022*, page 219–227, New York, NY, USA, 2022. ACM.

[40] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[41] Ben Hutchinson, Vinodkumar Prabhakaran, Emily Denton, Kellie Webster, Yu Zhong, and Stephen Denuyl. Social biases in NLP models as barriers for persons with disabilities. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel

Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5491–5501, Online, July 2020. Association for Computational Linguistics.

[42] Gert Jacobusse and Cor Veenman. On selection bias with imbalanced classes. In Toon Calders, Michelangelo Ceci, and Donato Malerba, editors, *Discovery Science*, pages 325–340, Cham, 2016. Springer International Publishing.

[43] Muhammad Abdullah Jamal and Guo-Jun Qi. Task agnostic meta-learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11719–11727, 2019.

[44] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

[45] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, November 2020. Association for Computational Linguistics.

[46] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In *Proceedings of the 31st International Conference*

on Neural Information Processing Systems, NIPS'17, page 972–981, Red Hook, NY, USA, 2017. Curran Associates Inc.

[47] Jon Kleinberg and Manish Raghavan. Selection Problems in the Presence of Implicit Bias. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference*, volume 94 of *Leibniz International Proceedings in Informatics*, pages 33:1–33:17, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[48] William R. Knight. A computer method for calculating kendall's tau with ungrouped data. *Journal of the American Statistical Association*, 61(314):436–439, 1966.

[49] Preethi Lahoti, Gerhard Weikum, and Krishna P. Gummadi. ifair: Learning individually fair data representations for algorithmic decision making. *2019 IEEE 35th International Conference on Data Engineering*, pages 1334–1345, 2019.

[50] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli,

Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. Holistic evaluation of language models, 2023.

[51] Shichen Liu, Fei Xiao, Wenwu Ou, and Luo Si. Cascade ranking for operational e-commerce search. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 1557–1565, New York, NY, USA, 2017. Association for Computing Machinery.

[52] Hanchao Ma, Sheng Guan, Christopher Toomey, and Yinghui Wu. Diversified subgraph query generation with group fairness. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, page 686–694, New York, NY, USA, 2022. ACM.

[53] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`, 2022.

[54] Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1953–1967, Online, November 2020. Association for Computational Linguistics.

[55] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert, 2020.

[56] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. Document ranking with a pretrained sequence-to-sequence model. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, Online, November 2020. Association for Computational Linguistics.

[57] OpenAI. Gpt-4 technical report, 2023.

[58] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

[59] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[60] German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019. ISSN 0893-6080.

[61] Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel R. Bowman. BBQ: A hand-built bias benchmark for question answering. In Smaranda Muresan, Preslav Nakov, and

Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2086–2105. Association for Computational Linguistics, 2022.

[62] Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3419–3448, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[63] Andrea Dal Pozzolo, Olivier Caelen, Reid A. Johnson, and Gianluca Bontempi. Calibrating probability with undersampling for unbalanced classification. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 159–166, New York, NY, USA, 2015. IEEE.

[64] Jiarui Qin, Jiachen Zhu, Bo Chen, Zhirong Liu, Weiwen Liu, Ruiming Tang, Rui Zhang, Yong Yu, and Weinan Zhang. Rankflow: Joint optimization of multi-stage cascade ranking systems as flows. SIGIR '22, page 814–824, New York, NY, USA, 2022. Association for Computing Machinery.

[65] Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. Large language models are effective text rankers with pairwise ranking prompting, 2023.

[66] Aida Ramezani and Yang Xu. Knowledge of cultural moral norms in large language models. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 428–446. Association for Computational Linguistics, 2023.

[67] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 2988–2997. JMLR.org, 2017.

[68] Sebastin Santy, Jenny T. Liang, Ronan Le Bras, Katharina Reinecke, and Maarten Sap. Nlpositionality: Characterizing design biases of datasets and models. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9080–9102. Association for Computational Linguistics, 2023.

[69] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32, 2019.

[70] Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 2219–2228, New York, NY, USA, 2018. ACM.

[71] Dylan Slack, Sorelle A. Friedler, and Emile Givental. Fairness warnings and fair-maml: Learning fairly with minimal data. page 200–209, New York, NY, USA, 2020. Association for Computing Machinery.

[72] Julia Stoyanovich, Ke Yang, and HV Jagadish. Online set selection with fairness and diversity constraints. In *Proceedings of the EDBT Conference*, 2018.

[73] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is ChatGPT good at search? investigating large language models as re-ranking agents. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14918–14937, Singapore, December 2023. Association for Computational Linguistics.

[74] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR, 2013.

[75] Zhiqiang Tao, Yaliang Li, Bolin Ding, Ce Zhang, Jingren Zhou, and Yun Fu. Learning to mutate with hypergradient guided population. In *Advances in Neural Information Processing Systems*, volume 33, pages 17641–17651. Curran Associates, Inc., 2020.

[76] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[77] Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, Sang T. Truong, Simran Arora, Mantas Mazeika, Dan Hendrycks, Zinan Lin, Yu Cheng, Sanmi Koyejo, Dawn Song, and Bo Li. Decodingtrust: A comprehensive assessment of trustworthiness in GPT models. *CoRR*, abs/2306.11698, 2023.

[78] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. Combating selection biases in recommender systems with a few unbiased ratings. In *Proceedings of the 14th*

ACM International Conference on Web Search and Data Mining, WSDM '21, page 427–435, New York, NY, USA, 2021. Association for Computing Machinery.

[79] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. Learning to rank with selection bias in personal search. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, page 115–124, New York, NY, USA, 2016. Association for Computing Machinery.

[80] Yuan Wang, Zhiqiang Tao, and Yi Fang. A meta-learning approach to fair ranking. In *The 45th International ACM SIGIR Conference on Research & Development in Information Retrieval*, page 2539–2544, New York, NY, USA, 2022. ACM.

[81] Zhenlei Wang, Jingsen Zhang, Hongteng Xu, Xu Chen, Yongfeng Zhang, Wayne Xin Zhao, and Ji-Rong Wen. Counterfactual data-augmented sequential recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 347–356, New York, NY, USA, 2021. Association for Computing Machinery.

[82] Linda F Wightman. Lsac national longitudinal bar passage study. *LSAC Research Report Series*, 1998.

[83] Garrett Wilson and Diane J. Cook. A survey of unsupervised deep domain adaptation. *ACM Trans. Intell. Syst. Technol.*, 11(5), jul 2020. ISSN 2157-6904.

[84] Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online, November 2020. Association for Computational Linguistics.

[85] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 55–64, New York, NY, USA, 2017. Association for Computing Machinery.

[86] Ke Yang and Julia Stoyanovich. Measuring fairness in ranked outputs. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450352826.

[87] Ke Yang, Vasilis Gkatzelis, and Julia Stoyanovich. Balanced ranking with diversity constraints. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 6035–6042. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[88] Huaxiu Yao, Xian Wu, Zhiqiang Tao, Yaliang Li, Bolin Ding, Ruirui Li, and Zhenhui Li. Automated relational meta-learning. In *8th International Conference on Learning Representations*, 2020.

[89] Meike Zehlike and Carlos Castillo. *Reducing Disparate Exposure in Ranking: A Learning To Rank Approach*, page 2849–2855. Association for Computing Machinery, New York, NY, USA, 2020. ISBN 9781450370233.

[90] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. Fa*ir: A fair top-k ranking algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, page 1569–1578, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349185.

[91] Meike Zehlike, Philipp Hacker, and Emil Wiedemann. Matching code and law: Achieving algorithmic fairness with optimal transport. *Data Min. Knowl. Discov.*, 34(1):163–200, jan 2020. ISSN 1384-5810.

[92] Meike Zehlike, Ke Yang, and Julia Stoyanovich. Fairness in ranking: A survey. *arXiv preprint arXiv:2103.14000*, 2021.

[93] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 325–333, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[94] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference*

on AI, Ethics, and Society, AIES '18, page 335–340, New York, NY, USA, 2018. Association for Computing Machinery.

[95] Jizhi Zhang, Keqin Bao, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. Is chatgpt fair for recommendation? evaluating fairness in large language model recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys '23, page 993–999, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702419.

[96] Chen Zhao and Feng Chen. Unfairness discovery and prevention for few-shot regression. In *2020 IEEE International Conference on Knowledge Graph*, pages 137–144, 2020.

[97] Chen Zhao, Feng Chen, Zhuoyi Wang, and Latifur Khan. A primal-dual subgradient approach for fair meta learning. In *2020 IEEE International Conference on Data Mining*, pages 821–830. IEEE, 2020.

[98] Chen Zhao, Feng Chen, and Bhavani Thuraisingham. Fairness-aware online meta-learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, page 2294–2304, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325.