Mitigating Exposure Bias in Online Learning to Rank Recommendation
A Novel Reward Model for Cascading Bandits

Mansoury, Masoud; Mobasher, Bamshad; van Hoof, Herke

DOI
10.1145/3627673.3679763

Publication date
2024

Document Version
Final published version

Published in
CIKM '24

Mitigating Exposure Bias in Online Learning to Rank Recommendation: A Novel Reward Model for Cascading Bandits

Masoud Mansoury*
Delft University of Technology
Delft, The Netherlands
m.mansoury@tudelft.nl

Bamshad Mobasher
DePaul University
Chicago, USA
mobasher@cs.depaul.edu

Herke van Hoof
University of Amsterdam
Amsterdam, The Netherlands
h.c.vanhoof@uva.nl

Abstract

Exposure bias is a well-known issue in recommender systems where items and suppliers are not equally represented in the recommendation results. This bias becomes particularly problematic over time as a few items are repeatedly over-represented in recommendation lists, leading to a *feedback loop* that further amplifies this bias. Although extensive research has addressed this issue in model-based or neighborhood-based recommendation algorithms, less attention has been paid to online recommendation models, such as those based on top-K contextual bandits, where recommendation models are dynamically updated with ongoing user feedback. In this paper, we study exposure bias in a class of well-known contextual bandit algorithms known as *Linear Cascading Bandits*. We analyze these algorithms in their ability to handle exposure bias and provide a fair representation of items in the recommendation results. Our analysis reveals that these algorithms fail to mitigate exposure bias in the long run during the course of ongoing user interactions. We propose an **E**xposure-**A**ware reward model that updates the model parameters based on two factors: 1) implicit user feedback and 2) the position of the item in the recommendation list. The proposed model mitigates exposure bias by controlling the utility assigned to the items based on their exposure in the recommendation list. Our experiments with two real-world datasets show that our proposed reward model improves the exposure fairness of the linear cascading bandits over time while maintaining the recommendation accuracy. It also outperforms the current baselines. Finally, we prove a high probability upper regret bound for our proposed model, providing theoretical guarantees for its performance.

CCS Concepts

• **Information systems** → **Users and interactive retrieval**; **Recommender systems**.

Keywords

recommender systems, contextual bandits, exposure fairness

ACM Reference Format:
Masoud Mansoury, Bamshad Mobasher, and Herke van Hoof. 2024. Mitigating Exposure Bias in Online Learning to Rank Recommendation: A Novel Reward Model for Cascading Bandits. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3627673.3679763

*Work done while the author was with Elsevier Discovery Lab and University of Amsterdam.

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679763

1 Introduction

Recommender systems utilize users' interaction data on different items to generate personalized recommendations [2, 7, 30, 40]. Traditionally, the success of these systems are measured based on the degree to which the recommendations generated matches the user preferences [16, 41]. However, this user-centric view for building recommendation models, while captures users' preferences, neglects the item-side utilities, or what is referred to as *exposure bias* [17, 31, 42].

Problem definition. Exposure bias in recommender systems refers to the phenomenon of items not being uniformly represented in the recommendation results: Few items are frequently shown in the recommendation lists, while the majority of other items rarely appear in the recommendation results [10, 31]. This bias, if not addressed, can result in a number of negative consequences for system performance. First, it can impact economic gains for items for suppliers of underexposed items, leading to unfair treatment and disincentivizing participation in the marketplace [37]. Secondly, it may hinder the system's ability to provide useful, but less popular, recommendations to consumers [11, 21]. Finally, users have a greater chance of interacting with over-exposed items, perpetuating their prominence in future recommendations, and amplifying existing biases. Amplification of exposure bias for a few items would be at the expense of the under-exposure for a majority of other items (including some that might be of interest for some users) and consequently may push those items out of the marketplace[33, 34, 43].

Research gap. Most existing research to study exposure bias has focused on classical recommendation models in static settings where a single round of recommendation results is analyzed [31, 45]. Although these studies reveal important aspects of exposure bias and propose solutions to tackle it, the long-term impact of this bias on online learning-to-rank recommendation models has yet to be explored significantly. This is a research gap which we seek to remedy in this paper. Filling this gap requires studying the task of recommendation problem in dynamic and interactive settings where users are engaged in ongoing interaction with the system and preference models are dynamically updated over time.

In our study, we focus specifically on *Cascading Bandits* (CB) [23, 24, 27, 53] which provide a principled solution for online learning of recommendation models. The ability of CB to handle *position bias* [12, 20] and perform *exploration* [6, 8, 36] makes it an interesting choice for developing online recommendation algorithms. The main

question in this research is how CB distribute exposure among items in the system? Although these algorithms perform exploration in the items space to collect user feedback on different items, our study in this paper shows that this exploration does not necessarily lead to a sufficiently fair exposure for items in the long run.

Contributions and findings. In this paper, we study exposure bias in cascading bandits and introduce a novel reward model to mitigate exposure bias in these algorithms. In cascading bandit algorithms, all selected items in the recommendation lists are similarly rewarded, regardless of what their position is in the list. This means that a clicked item on top of the list (e.g., at the first position) is equally rewarded as a clicked item at the bottom of the list (e.g., at position K). Also, the same formulation is considered to penalize ignored (unclicked) items. We hypothesize that considering the positional information of clicked/unclicked items when rewarding/penalizing those items would not only lead to a better adaptation of the model to user feedback, but also, most importantly, lead to a significant reduction in exposure bias over time.

We propose an **E**xposure-**A**ware (EA) reward model and integrate it into the existing cascading bandit algorithms. Our reward model updates the bandit model parameters based on two factors: 1) the user feedback on recommended items, whether the item is clicked or not, and 2) the position of the item in the list. In fact, the proposed model rewards or penalizes the clicked or unclicked items, respectively, based on their position in the recommendation list. This control over the degree of reward or penalization for items based on their exposure in the recommendation lists incentivizes more exploration and reduces exposure bias on items. Extensive experiments on two real-world datasets show that the proposed reward model not only reduces the exposure bias in cascading bandits, but also outperforms the state-of-the-art baselines in mitigating exposure bias while maintaining the recommendation accuracy. We also show theoretical guarantees for the performance of our reward model by proving a high probability upper regret bound for it.

2 Background

In this section, we review the CB and the definitions of exposure fairness in recommender systems. Formally, $\mathcal{I} = \{1, ..., m\}$ be the set of all items in the system. The task of generating recommendations in each round $t \in \{1, 2, ..., n\}$ is delivering a recommendation list of size K to a target user. Let denote this recommendation list as $\mathcal{L}_t \in \Pi_K(\mathcal{I})$, where $\Pi_K(\mathcal{I})$ is the set of all K-permutations of the set $\mathcal{I}$. $\mathcal{L}(k)$ denotes the item in the k-th position of $\mathcal{L}$.

2.1 Cascading bandit

The Cascade Bandit (CB) is a principled method of operationalizing recommendation models in an online environment under the assumption that users will behave according to a cascade model [23, 53]. The cascade click model [13] is a well-known click model to interpret the click behavior of users on the recommendation list. Given the recommendation list $\mathcal{L}$, the target user examines each recommended item in $\mathcal{L}$ from the first position to the last, clicks on the first attractive item and stops examining the rest of the items. In this way, the items above the clicked item are considered unattractive, the clicked item is considered attractive, and the rest of the items are considered as unobserved. The probability that a user clicks on an item $\mathcal{L}(k)$ is called *attraction probability* and we denote it as $\omega(\mathcal{L}(k))$. In the following, we describe the cascading bandit formulation for a user u interacting with the system.

Cascading bandits can be represented by a tuple ($\mathcal{I}$,K,P), where P is a probability distribution over a binary hypercube $\{0, 1\}^{\mathcal{I}}$. Also, let $\mathrm{w}_t \in \{0, 1\}^{\mathcal{I}}$ denote the preference weights for each item drawn from P, the degree to which u is interested to each item where $\mathrm{w}_t(\mathcal{L}(i)) = 1$ signifies that the item $\mathcal{L}(i)$ attracts u in round t. Also, assuming that the preference weights of items in the ground set $\mathcal{I}$ are independently distributed as:

$$P(\mathrm{w}) = \prod_{i \in \mathcal{I}} \mathrm{Ber}_{\omega(i)}(\mathrm{w}(i)) \tag{1}$$

where $\mathrm{Ber}_{\omega(i)}(.)$ is the Bernoulli distribution with mean $\omega(i)$.

In each round t, the learning agent provides a recommendation list of size K, $\mathcal{L}_t \in \Pi_K(\mathcal{I})$, to the target user. According to the cascade click model, the user examines $\mathcal{L}_t$ from the first item (i.e., $\mathcal{L}(1)$) to the last one (i.e., $\mathcal{L}(K)$) and clicks on the first item of interest. We use $C_t \in \{1, .., K, K+1\}$ to denote the position of the clicked item. Note that $C_t \leq K$ holds if user clicks on an item in $\mathcal{L}_t$, otherwise $C_t = K + 1$. Since user only clicks on the first "attractive" item, $\mathrm{w}_t(\mathcal{L}(k))$ can be defined as:

$$\mathrm{w}_t(\mathcal{L}(k)) = \mathbb{1}(C_t = k), \quad \text{where } k \in [1, ..., \min\{K, C_t\}] \tag{2}$$

where $\mathbb{1}[.]$ is the indicator function returning zero when its argument is False and 1 otherwise. And the reward is defined as:

$$\mathcal{R}(\mathcal{L}_t, \mathrm{w}_t) = 1 - \prod_{i=1}^{K} (1 - \mathrm{w}_t(\mathcal{L}_t(i))) \tag{3}$$

The goal of the agent is to minimize the disparity in reward observed on the generated recommendation list by the agent and the optimal ranker (or equivalently maximizing the number of clicks observed on recommended items) and can be computed as:

$$R(n) = \mathbb{E}\left[\sum_{t=1}^{n} \mathcal{R}(\mathcal{L}^*, \mathrm{w}_t) - \mathcal{R}(\mathcal{L}_t, \mathrm{w}_t)\right] \tag{4}$$

where $\mathcal{L}^*$ is the *optimal recommendation list* that maximizes the reward at each time t and is computed as follows.

$$\mathcal{L}^* = \underset{\mathcal{L} \in \prod_K(\mathcal{I})}{\operatorname{argmax}} \ \mathcal{R}(\mathcal{L}, \omega) \tag{5}$$

2.2 Measuring exposure fairness

In the existing literature, there are many metrics available to measure exposure fairness in recommender systems [18, 28, 39, 42, 51]. In this study, our focus is on assessing exposure fairness through various dimensions within the family of exposure metrics. Specifically, we scrutinize two critical dimensions: (i) consideration or disregard of the item's position in the recommendation list (w/ or w/o position, respectively), and (ii) allocation of exposure proportionately or irrespective of items' merit (w/ or w/o merit, respectively). Table 1 provides an overview of four distinct notions of exposure based on these dimensions.[1]

Exposure fairness, as we define it, refers to the equitable distribution of exposure among items. With an exposure distribution

[1]Merit, in this context, refers to any quality measure for items, such as relevance [5]. The definition of the merit measure employed in this paper is elucidated in Section 5.

Table 1: Four different notions of exposure for items ($\mathcal{U}$ is the set of all users, $\mathcal{L}_u$ is the recommendation list delivered to user u, and K is the size of the recommendation list).

	w/o merit	w/ merit
w/o position	*Exposure is binary without considering item's merit* $E^B(i) = \sum_{u \in \mathcal{U}} \mathbb{1}[i \in \mathcal{L}_u]$	*Exposure is binary in proportion to item's merit* $E^{BM}(i) = \frac{E^B(i)}{merit(i)}$
w/ position	*Exposure depends on the position without considering item's merit* $E^P(i) = \sum_{u \in \mathcal{U}} \sum_{k=1}^{K} \mathbb{1}[i \in \mathcal{L}_u] \frac{1}{\log_2(1+k)}$	*Exposure depends on position in proportion to item's merit* $E^{PM}(i) = \frac{E^P(i)}{merit(i)}$

representing the allocated exposure value for each item, our objective is to assess the extent to which this distribution achieves uniformity, with a uniform distribution being deemed the fairest. The Gini index [3, 48] is a well-known metric to measure the uniformity of a distribution. Given that the Gini index falls within the range of $[0, 1]$, for consistency, we report 1 minus the Gini index in this paper. Consequently, a Gini index value of 1 signifies the fairest outcome, while a value of 0 denotes the most unfair outcome. Calculating the Gini index on the exposure distribution derived from each definition outlined in Table 1 yields four notions of exposure fairness:

- **Equality of binary exposure (Equality$^{(B)}$)**: This computes the Gini Index over the exposure distribution of E^B (i.e., w/o position and w/o merit).
- **Equality of position-based exposure (Equality$^{(P)}$)**: This computes the Gini Index over the exposure distribution of E^P (i.e., w/ position and w/o merit).
- **Equity of binary exposure (Equity$^{(B)}$)**: This computes the Gini Index over the exposure distribution of E^{BM} (i.e., w/o position and w/ merit).
- **Equity of position-based exposure (Equity$^{(P)}$)**: This computes the Gini Index over the exposure distribution of E^{PM} (i.e., w/ position and w/ merit).

3 Exposure-Aware Cascading Bandits

The previous approaches based on cascading bandits do not consider the position of clicked or ignored (unclicked) items when assigning rewards or penalties[2]. This means that items clicked on at the top of the list receive the same reward as those clicked at the bottom. Users tend to select highly exposed items, often positioned at the top, either due to their accessibility or genuine interest [14, 29]. Conversely, less exposed items towards the bottom of the list require more effort from users to discover, and when clicked, are likely of higher importance. Thus, to better adjust the model to user behavior, clicked items at the bottom should be rewarded more than those at the top. This also provides additional incentives for assigning more exposure to less exposed items in the future.

[2]For the rest, when an item is examined, but it is not clicked, we call it an "unclicked" item. This is different from the unobserved items that have not even been examined.

Table 2: Weighting functions to define $\mathcal{F}_{t,k}$ in Eq. 6 (k is the position of the item in the list and t is the current round).

Function	Abbreviation	Formula	Parameters
Logarithmic	Log	$\log(1+k)$	–
Exponential	RBP	β^{k-1}	patience β
Linear	Linear	$\beta \times k$	patience β

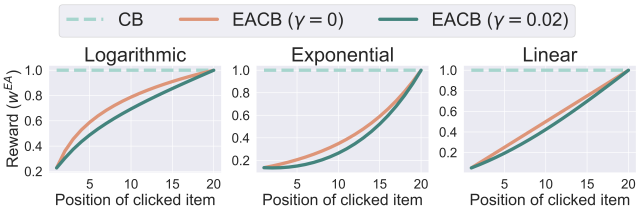


Figure 1: Reward distribution of CB and EACB for different weight functions when click is observed at varying positions in the list.

Similarly, unclicked items should be penalized differently depending on their position. Unclicked items on the top should receive a greater penalty than those at the bottom. When a highly exposed item is not clicked, it suggests that the recommendation model inaccurately assumed that it was of high interest. Penalizing such items more heavily than less exposed unclicked items helps refine the model to avoid prioritizing them in future recommendations. This also incentives for downgrading recently over exposed items and promoting less exposed items in the future. To address these issues, we propose an Exposure-Aware Cascading Bandit (EACB) that adjusts rewards based on the position of clicked items in the recommendation list. Hence, we reformulate Eq. 2 as:

$$\mathrm{w}_t^{EA}(\mathcal{L}(k)) = \mathcal{F}_{t,k} \times \mathbb{1}[C_t = k] - \gamma \mathcal{F}_{t,k} \times \mathbb{1}[C_t < k] \qquad (6)$$

where $k \in [1, ..., \min\{K, C_t\}]$ and $\mathcal{F}_{t,k}$ is the *exposure-aware weight function* that assigns weights to all examined items based on their position in the recommendation list. The indicator function ensures that the appropriate term (reward or penalty) is applied based on whether the item is clicked or unclicked: if the examined item is clicked, the first term (reward) applies as $\mathbb{1}[C_t = k] = 1$ and $\mathbb{1}[C_t < k] = 0$, otherwise, the second term (penalization) applies as $\mathbb{1}[C_t = k] = 0$ and $\mathbb{1}[C_t < k] = 1$. The hyperparameter γ controls the degree of penalization for unclicked items. A small γ value allows for slight penalization of unclicked items, with the focus primarily on learning user preferences from clicked items.

The choice of $\mathcal{F}_{t,k}$ is crucial for an effective exposure-aware reward model. It must meet two criteria: 1) positively weight clicked items at the bottom more than those at the top, and 2) negatively weight unclicked items at the top more than those at the bottom. In this paper, we consider three different weight functions, outlined in Table 2, which align with established browsing models [44]. For example, the logarithmic function follows the standard exposure drop-off [42] used in ranking metrics (e.g., nDCG), while the exponential function follows Rank-Biased Precision (RBP) [38]. Figure 1 shows the reward distribution of CB and EACB for different weight functions, showcasing the varying intensities of weighting assigned

to observed clicked items. The logarithmic function exhibits the highest intensity, followed by the linear and exponential functions.

According to the cascade click model and Eq. 6, the examination probability of an item $\mathcal{L}(k)$ in EACB would be:

$$\prod_{i=1}^{k-1} (1 - (\mathcal{F}_{t,k} \times \mathbb{1}[C_t = k] - \gamma \mathcal{F}_{t,k} \times \mathbb{1}[C_t < k])) \tag{7}$$

and the expectation of reward at round t is computed as follows:

$$\mathcal{R}(\mathcal{L}_t, \mathrm{w}_t^{EA}) = 1 - \prod_{i=1}^{K} (1 - (\mathcal{F}_{t,k} \times \mathbb{1}[C_t = k] - \gamma \mathcal{F}_{t,k} \times \mathbb{1}[C_t < k])) \tag{8}$$

3.1 Algorithm for learning EACB

Various algorithms have been developed within the Cascading Bandit (CB) framework [19, 26, 53]. In this paper, we specifically concentrate on the linear cascading bandit proposed by Zong et al. [53] and extend it to incorporate our exposure-aware reward model.

The algorithm 1 presents the algorithmic process of our EACB. In each round, the algorithm computes the *attraction probability*, $w(i)$, of a target item i, representing the likelihood of the target user liking the item. This probability is derived from the dot product of the item features, x_i, and the user preference vector θ^*, denoted as $w(i) = \theta^* x_i^T$. While item features are known to the algorithm, the user preference vector θ^* is unknown and must be learned through user interactions. Thus, in the initial step (line 4), the algorithm estimates the user preference vector from past observations on item features and their corresponding attraction probabilities.

This estimation process can be framed as a ridge regression problem, where $\hat{\theta}_t$ is computed as:

$$\hat{\theta}_t = (X_t^T X_t + \lambda I)^{-1} X_t^T \hat{W}_t \tag{9}$$

where $X_t \in \mathbb{R}^{m \times d}$ is the matrix of item features, $\hat{W}_t \in \mathbb{R}^{m \times 1}$ is the vector of items' attraction probabilities at round t, and λ is the regularization term. The algorithm iteratively updates the model parameters $M_t = X_t^T X_t + \lambda I$ and $B_t = X_t^T \hat{W}_t$.

To address uncertainty in estimating user preferences and enable exploration in the item space, an item selection strategy is employed. Examples of these strategies are ϵ-Greedy [46], Upper Confidence Bound [4, 25], and Thompson Sampling [9, 47]. In this paper, we focus on the Upper Confidence Bound (UCB) item selection strategy and leave the investigation on other strategies as our future work. According to UCB, the score for each item i is predicted by combining the estimation of attraction probability with an upper bound (line 7), as expressed by:

$$U_t(i) = \hat{\theta}_t x_i^T + \alpha \sqrt{x_i M_{t-1}^{-1} x_i^T} \tag{10}$$

where $M_{t-1} \in \mathbb{R}^{d \times d}$ is the co-variance matrix of item features. The term $\sqrt{x_i M_{t-1}^{-1} x_i^T}$ is the upper bound for the estimated weight of item i which covers the optimal weight and is computed by norm of x_i weighted by M_{t-1}^{-1} (i.e., $\|x_i\|_{M_{t-1}^{-1}}$). α is a hyperparameter that controls the degree of exploration. Given scores computed for each item using Eq. 10, K items with the largest $U_t(i)$ are returned as the recommendation list (lines 9-12).

Algorithm 1 Exposure-aware cascading bandit algorithm

Input: Number of rounds n, size of recommendation list K, number of feature d, learning rate σ

1: // Initialization
2: $\mathbf{M} \leftarrow \lambda I^{d \times d}$, $\mathbf{B} \leftarrow 0^d$
3: **for** $t = 1, ..., n$ **do**
4: $\quad \hat{\theta}_t \leftarrow \sigma^{-2} M_{t-1}^{-1} B_{t-1}$
5: $\quad$ // Recommend a list of K items
6: $\quad$ **for** $i \in \mathcal{I}$ **do**
7: $\quad\quad$ Compute $U_t(i)$ using Eq. 10
8: $\quad$ **end for**
9: $\quad$ **for** $k = 1, ...K$ **do**
10: $\quad\quad \mathbf{i}_k \leftarrow \mathrm{argmax}_{e \in \mathcal{I} \setminus \{\mathbf{i}_1, ..., \mathbf{i}_{k-1}\}} \mathcal{S}_t(e)$
11: $\quad$ **end for**
12: $\quad \mathcal{L}_t \leftarrow (\mathbf{i}_1, ..., \mathbf{i}_K)$
13: $\quad$ // Collect user's feedback on $\mathcal{L}$
14: $\quad$ Display $\mathcal{L}_t$ and observe click feedback $C_t \in \{1, ..., K, K+1\}$
15: $\quad$ //Update model parameters
16: $\quad$ **for** $k = 1, ..., \min\{K, C_t\}$ **do**
17: $\quad\quad M_t \leftarrow M_{t-1} + \sigma^{-2} x_{\mathcal{L}(k)}^T x_{\mathcal{L}(k)}$
18: $\quad\quad$ **if** $C_t == k$ **then**
19: $\quad\quad\quad B_t \leftarrow B_{t-1} + \mathcal{F}_{t,k} x_{\mathcal{L}(k)}$
20: $\quad\quad$ **else**
21: $\quad\quad\quad B_t \leftarrow B_{t-1} - \gamma \mathcal{F}_{t,k} x_{\mathcal{L}(k)}$
22: $\quad\quad$ **end if**
23: $\quad$ **end for**
24: **end for**

Upon receiving feedback from the user for the recommendation list $\mathcal{L}_t$ (line 14), the agent updates the model parameters (lines 16-23) based on the user's feedback. Specifically, if an examined item is clicked, the parameter B_t is rewarded; otherwise, it is penalized.

It should be noted that our proposed EACB algorithm involves several tunable hyperparameters, including α for exploration control, σ for the growth rate of M_t, λ for regularization and γ for the degree of penalization on unclicked items. Adjusting these hyperparameters enables fine-tuning of the algorithm's performance.

4 Analysis of regret upper-bound

In this section, we present the upper bound of n-step-regret for our proposed exposure-aware cascading bandits. Our analysis shows that with an extra condition (the choice of γ), our exposure-aware cascading bandit has the same upper bound for n-step-regret as the original cascading bandits [53] as follows:

Theorem 1. *For any $\sigma > 0$, $\|\theta^*\|_2 \leq 1$, and*

$$\alpha \geq \frac{1}{\sigma} \sqrt{d \log\left(1 + \frac{nK}{d\sigma^2}\right) + 2\log(n)} + \left\|\theta^*\right\|_2 \tag{11}$$

$$\gamma \leq \frac{1}{\mathcal{F}_{t,k}} - 1\,, \quad \forall k \in \{1, ..., K\} \tag{12}$$

we have,

$$R(n) \leq 2\alpha K \sqrt{\frac{dn \log\left[1 + \frac{nK}{d\sigma^2}\right]}{\log\left(1 + \frac{1}{\sigma^2}\right)}} + 1. \tag{13}$$

This theorem implies that for sufficiently optimistic α and γ, combining the equations 11, 12, and 13, we have $R(n) = O(dK\sqrt{n})$ where O ignores logarithmic factors. This is also the same upper bound for the original linear cascading bandits [53]. Moreover, this bound signifies two properties: (1) it states a near optimal bound with factor $\sqrt{n}$, (2) the bound is linear in the size of the recommendation lists and the number of features, which is a common dependence in learning bandit algorithms [1].

4.1 Proof of Theorem 1

Let $\Pi(\mathcal{I}) = \bigcup_{i=1}^{m} \Pi_i(\mathcal{L})$ be all possible recommendation lists in the item catalog $\mathcal{I}$ and $O : \Pi(\mathcal{I}) \leftarrow [0, 1]$ be an arbitrary weight function for the lists. According to the reward model in Eq. 8, the expected reward of a recommendation list can be computed as:

$$f(\mathcal{L}, O) = 1 - \prod_{i=1}^{K} \left(1 - \mathcal{F}_{t,i} \times O(\mathcal{L}(i)) + \gamma \times \mathcal{F}_{t,i} \times (1 - O(\mathcal{L}(i)))\right) \tag{14}$$

For each item in $\mathcal{L}$ we define O, its high probability upper-bound H_t, and its high probability lower-bound L_t as:

$$\begin{aligned} O(\mathcal{L}(i)) &= \theta^* x_{\mathcal{L}(i)}^T \\ H_t(\mathcal{L}(i)) &= \text{Func}_{[0,1]}\left(\hat{\theta}_t x_{\mathcal{L}(i)}^T + \alpha\sqrt{x_{\mathcal{L}(i)} M_t^{-1} x_{\mathcal{L}(i)}^T}\right) \\ L_t(\mathcal{L}(i)) &= \text{Func}_{[0,1]}\left(\hat{\theta}_t x_{\mathcal{L}(i)}^T - \alpha\sqrt{x_{\mathcal{L}(i)} M_t^{-1} x_{\mathcal{L}(i)}^T}\right) \end{aligned} \tag{15}$$

where $\text{Func}_{[0,1]}(.) = max(0, min(1, 0))$, projecting the estimated value onto range $[0, 1]$. We also define the following notation:

$$\psi_{t,i}^{\mathcal{X}} = \mathcal{F}_{t,i} \times \mathcal{X}(\mathcal{L}(i)), \quad \psi_{t,i}^{\prime\mathcal{X}} = \mathcal{F}_{t,i} \times (1 - \mathcal{X}(\mathcal{L}(i)) \tag{16}$$

where $\mathcal{X}$ refers to one of the functions (O, H_t, or L_t) in Eq. 15. Now, we start our proof by defining event

$$\mathcal{E}_t = \{L_t(\mathcal{L}(i)) \le O(\mathcal{L}(i)) \le H_t(\mathcal{L}(i)), \forall i \in [1, K], \forall \mathcal{L} \in \Pi(\mathcal{I})\}$$

and $\bar{\mathcal{E}}_t$ as its complement. $\mathcal{E}_t$ contains all the lists that the attraction probability estimation of its items falls into the upper and lower confidence bound which is the main ingredient of the UCB item selection strategy. We derive the regret bound for a single time step t and then extend it to the upper bound of cumulative regret of n time steps. Hence, we have,

$$\begin{aligned} &\mathbb{E}\left[\mathcal{R}(\mathcal{L}^*, \omega) - \mathcal{R}(\mathcal{L}_t, \mathrm{w}_t)\right] = \mathbb{E}\left[f(\mathcal{L}^*, O) - f(\mathcal{L}_t, O)\right] \\ &\overset{a}{\le} P(\mathcal{E}_t)\,\mathbb{E}\left[f(\mathcal{L}^*, O) - f(\mathcal{L}_t, O)\right] + P(\bar{\mathcal{E}}_t) \\ &\overset{b}{\le} P(\mathcal{E}_t)\,\mathbb{E}\left[f(\mathcal{L}^*, H_t) - f(\mathcal{L}_t, O)\right] + P(\bar{\mathcal{E}}_t) \\ &\overset{c}{\le} P(\mathcal{E}_t)\,\mathbb{E}\left[f(\mathcal{L}_t, H_t) - f(\mathcal{L}_t, O)\right] + P(\bar{\mathcal{E}}_t) \end{aligned} \tag{17}$$

where (a) holds because $\mathbb{E}\left[f(\mathcal{L}^*, O) - f(\mathcal{L}_t, O)\right] \le 1$; (b) holds because given the inequality

$$f(\mathcal{L}^*, H_t) \le \max_{\mathcal{L} \in \prod_K(\mathcal{I})} f(\mathcal{L}, H_t) \le f(\mathcal{L}_t, H_t) \tag{18}$$

and event $\mathcal{E}_t$, we have $f(\mathcal{L}^*, O) \le f(\mathcal{L}^*, H_t)$; (c) holds because $f(\mathcal{L}^*, H_t) - f(\mathcal{L}_t, O) \le [f(\mathcal{L}_t, H_t) - f(\mathcal{L}_t, O)]$ from Eq. 18.

Let $\mathcal{H}_t$ be the history of data collected up to time t. Then, for any $\mathcal{H}_t$ such that $\mathcal{E}_t$ holds, together with Eq. 14 and 15, we have,

$$\begin{aligned} &f(\mathcal{L}_t, H_t) - f(\mathcal{L}_t, O) = \prod_{i=1}^{K}\left(1 - \psi_{t,i}^{O} + (\gamma \times \psi_{t,i}^{\prime O})\right) - \prod_{i=1}^{K}\left(1 - \psi_{t,i}^{H} + (\gamma \times \psi_{t,i}^{\prime H})\right) \\ &\overset{a}{=} \sum_{i=1}^{K}\left[\prod_{j=1}^{i-1}\left(1 - \psi_{t,j}^{O} + (\gamma \times \psi_{t,j}^{\prime O})\right)\right]\left(\psi_{t,i}^{H} - (\gamma \times \psi_{t,i}^{\prime H}) - \psi_{t,i}^{O} + (\gamma \times \psi_{t,i}^{\prime O})\right) \\ &\qquad\qquad \left[\prod_{k=i+1}^{K}\left(1 - \psi_{t,k}^{H} + (\gamma \times \psi_{t,k}^{\prime H})\right)\right] \\ &\overset{b}{\le} \sum_{i=1}^{K}\left[\prod_{j=1}^{i-1}\left(1 - \psi_{t,j}^{O} + (\gamma \times \psi_{t,j}^{\prime O})\right)\right]\left(\psi_{t,i}^{H} - (\gamma \times \psi_{t,i}^{\prime H}) - \psi_{t,i}^{O} + (\gamma \times \psi_{t,i}^{\prime O})\right) \end{aligned}$$

where (a) follows Lemma 1 in [53]; and (b) holds because $\psi_{t,k}^{H} + \gamma \times \psi_{t,k}^{\prime H} \le 1$. Now, we define the event $\mathcal{G}_{t,i} = \{\text{item } \mathcal{L}_t(i) \text{ is examined}\}$ where we have $\mathbb{E}\left[\mathbb{1}(\mathcal{G}_{t,i})\right] = \prod_{j=1}^{i-1}(1 - \psi_{t,j}^{O} + \gamma \times \psi_{t,j}^{\prime O})$. Then, for any $\mathcal{H}_t$ under $\mathcal{E}_t$, we have,

$$\begin{aligned} &\mathbb{E}\left[f(\mathcal{L}_t, H_t) - f(\mathcal{L}_t, O) \mid \mathcal{H}_t\right] \\ &\le \sum_{i=1}^{K} \mathbb{E}\left[\mathbb{1}[\mathcal{G}_{t,i}] \mid \mathcal{H}_t\right]\left(\psi_{t,i}^{H} - (\gamma \times \psi_{t,i}^{\prime H}) - \psi_{t,i}^{L} + (\gamma \times \psi_{t,i}^{\prime L})\right) \\ &\overset{a}{\le} 2\alpha\,\mathbb{E}\left[\mathbb{1}[\mathcal{G}_{t,i}] \sum_{i=1}^{K}\left[\sqrt{x_{\mathcal{L}(i)} M_t^{-1} x_{\mathcal{L}(i)}^T}\right].\left[\mathcal{F}_{t,i} \times (1+\gamma)\right] \,\middle|\, \mathcal{H}_t\right] \\ &\overset{b}{\le} 2\alpha\,\mathbb{E}\left[\sum_{i=1}^{min\{K, C_t\}}\left[\sqrt{x_{\mathcal{L}(i)} M_t^{-1} x_{\mathcal{L}(i)}^T}\right].\left[\mathcal{F}_{t,i} \times (1+\gamma)\right] \,\middle|\, \mathcal{H}_t\right] \end{aligned}$$

where (a) follows the definition of H_t and L_t from Eq. 15; and (b) follows the definition of $\mathcal{G}_{t,i}$. Thus, with $\phi_{t,\mathcal{L}(i)} = \sqrt{x_{\mathcal{L}(i)} M_t^{-1} x_{\mathcal{L}(i)}^T}$, the cumulative regret of n rounds can be defined as:

$$\begin{aligned} R(n) &= \sum_{t=1}^{n} \mathbb{E}\left[\mathcal{R}(\mathcal{L}^*, \omega) - \mathcal{R}(\mathcal{L}_t, \mathrm{w}_t)\right] \\ &\le \sum_{t=1}^{n}\left[2\alpha\,\mathbb{E}\left[\sum_{i=1}^{min\{K, C_t\}} \phi_{t,\mathcal{L}(i)} \times \mathcal{F}_{t,i} \times (1+\gamma) \,\middle|\, \mathcal{E}_t\right] P(\mathcal{E}_t) + P(\bar{\mathcal{E}}_t)\right] \\ &\le 2\alpha\,\mathbb{E}\left[\sum_{t=1}^{n}\sum_{i=1}^{min\{K, C_t\}} \phi_{t,\mathcal{L}(i)} \times \mathcal{F}_{t,i} \times (1+\gamma)\right] + \sum_{t=1}^{n} P(\bar{\mathcal{E}}_t) \end{aligned} \tag{19}$$

The regret bound can be derived by finding the worst-case bound on $\sum_{t=1}^{n}\sum_{i=1}^{min\{K, C_t\}}\left[\phi_{t,\mathcal{L}(i)} \times \mathcal{F}_{t,i} \times (1+\gamma)\right]$ and $\sum_{t=1}^{n} P(\bar{\mathcal{E}}_t)$ terms in Eq. 19. It should be noted that this is the same problem as in the original cascading bandits in [19, 26, 53] except for the first term that contains an additional $\left[\mathcal{F}_{t,i} \times (1+\gamma)\right]$ term. **This is the main advantage of our proposed EACB which guarantees a lower upper-bound for the n-step-regret compared to CB. The reason is that with a proper choice for the value of $0 < \gamma < \frac{1}{\mathcal{F}_{t,i}} - 1$, we have $\left[\mathcal{F}_{t,i} \times (1+\gamma)\right] < 1$ which leads to a smaller value for the first term (compared to CB):**

$$\mathbb{E}\left[\sum_{t=1}^{n}\sum_{i=1}^{min\{K, C_t\}} \phi_{t,\mathcal{L}(i)} \times \mathcal{F}_{t,i} \times (1+\gamma)\right] \le \mathbb{E}\left[\sum_{t=1}^{n}\sum_{i=1}^{min\{K, C_t\}} \phi_{t,\mathcal{L}(i)}\right]$$

where this inequality makes the first term in Eq. 19 similar to CB. Therefore, according to Lemma 2 in [53], we have,

$$\sum_{t=1}^{n} \sum_{i=1}^{min\{K,C_t\}} \sqrt{x_{\mathcal{L}(i)} M_t^{-1} x_{\mathcal{L}(i)}^T} \leq K \sqrt{\frac{dn \log\left[1 + \frac{nK}{d\sigma^2}\right]}{\log\left(1 + \frac{1}{\sigma^2}\right)}} \quad (20)$$

which is the worst-case bound for the first term in 19. Also, for the second term in Eq. 19, according to Lemma 3 in [53], we have $P(\bar{\mathcal{E}}_t) \leq 1/n$ for any α that satisfies Eq. 11. Therefore, together with Eq. 20 and 19, we have,

$$R(n) \leq 2\alpha K \sqrt{\frac{dT \log\left[1 + \frac{TK}{d\sigma^2}\right]}{\log\left(1 + \frac{1}{\sigma^2}\right)}} + 1. \quad (21)$$

which proves the Theorem 1.

5 Experiments

Our experimental analysis on real-world datasets is designed to address the following research questions: (**RQ1**) What impact does adjusting the degree of exploration in the original linear cascading bandit have on exposure bias? (**RQ2**) Does our exposure-aware cascading bandit algorithm better mitigate the effect of exposure bias than existing exposure bias mitigation methods? (**RQ3**) How does varying the penalization parameter (γ) influence the performance of our exposure-aware cascading bandit algorithm?

5.1 Datasets

Our experiments are conducted on two publicly-available datasets: MovieLens 1M [15] and Yahoo Music[3]. The MovieLens dataset comprises 6K users who provided 1M ratings for 4K items. On the other hand, the Yahoo Music dataset contains ratings from 1.8M users for 136K songs, totaling 700M ratings. In both datasets, ratings fall within the range of $[1, 5]$.

We follow the data preprocessing approach in [19, 26]. First, we map the ratings onto a binary scale: rating 4 and 5 are converted to 1 and other ratings to 0. Then, on MovieLens dataset, we create a sample of the data by extracting the 1000 most active users from the interaction data. On Yahoo Music dataset, we extract the 1000 most active users and the 1000 most rated songs from the interaction data. After this preprocessing, approximately 9% and 1% of the original ratings are retained for MovieLens and Yahoo Music, respectively.

5.2 Evaluation metrics and baselines

In our experimental evaluation, we investigate the impact of integrating our exposure-aware reward model into the linear cascading bandit algorithm [53]. We compare the performance of this modified algorithm, termed `EALinUCB`, with the original linear cascading bandit algorithm (`LinUCB`), where our reward model is not employed. For brevity, we omit the term "Cascade" in the names of the algorithms. Additionally, we consider three variations of `EALinUCB`, each utilizing a different weight function for training, as detailed in Table 2. We also compare `EALinUCB` with the following baselines:

- **Exposure-Aware aRm Selection (`EARSLinUCB`) [22]:** This method adopts a post-processing approach to improve exposure fairness. It reranks the generated recommendations by shuffling less relevant items to the bottom of the list, thus enhancing exposure fairness through randomization.
- **Fairness Regret Minimization (`FRMLinUCB`) [49]:** This baseline addresses exposure bias by formulating the bandit problem to minimize both reward regret and fairness regret. It assigns exposure to items proportionally to their merit, thereby promoting exposure fairness. Our implementation of fairness regret is based on the Equity^P notion, where each item's exposure is proportional to its true relevancy score (see Section 5.3).

The key distinction between `EARSLinUCB` and `FRMLinUCB` compared to our proposed approach lies in their intervention strategy. Although these baselines intervene during the recommendation generation step, our exposure-aware cascading bandit algorithm intervenes during the reward/penalization step. This allows our approach to be more generalizable in various cascading bandit algorithms [19, 26], as its effectiveness is not contingent on the performance of the underlying bandit algorithm. We leave the research on the generalizability of our EACB as our future work.

To measure the degree of exposure bias in each bandit algorithm, we utilize four metrics introduced in Section 2.2. Higher values for these metrics indicate less exposure bias or a fairer exposure distribution among items. In addition, we evaluate the accuracy of the model using the following metrics:

- **Average number of clicks ($\overline{clicks}$):** This metric measures the total number of clicks (#clicks) normalized by the number of users and number of rounds, providing insight into the model's performance in generating relevant recommendations:

$$\overline{clicks} = \frac{\#clicks}{|\mathcal{U}| \times n} \quad (22)$$

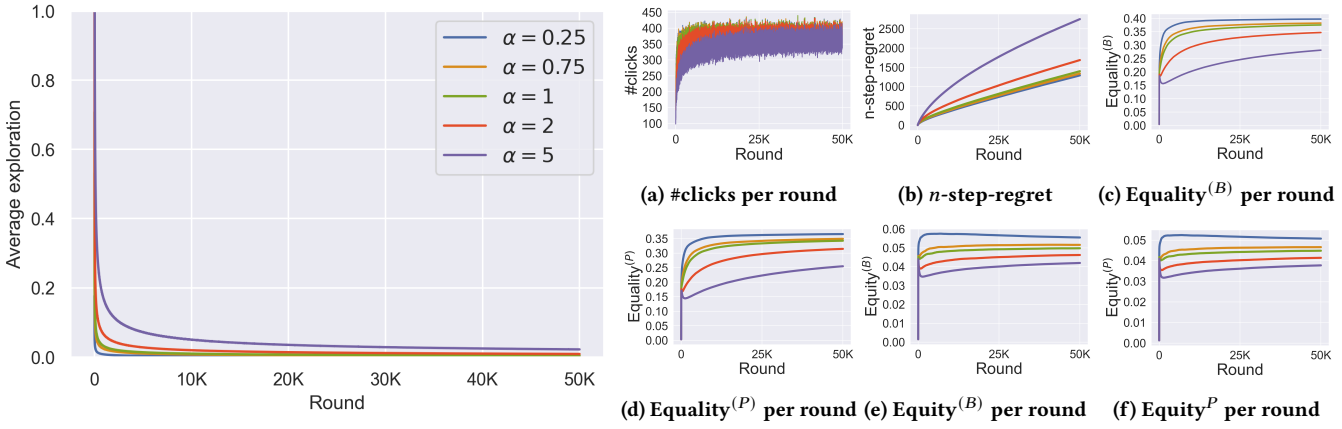
  where $0 \leq \overline{clicks} \leq 1$, 0 signifies no click and 1 indicates that all users clicked at least on one item at each round which is more desirable.
- **n-step-regret:** This metric measures cumulative regret, the difference in the observed number of clicks between the optimal ranker and the online ranker in n rounds as defined in Eq. 4. To ensure fair comparison, we use the original reward model (i.e., Eq. 3) for computing the n-step-regret for all algorithms, even though `EALinUCB` employs a different reward model during the training process (i.e., Eq. 8).

5.3 Simulation and experimental setup

The evaluation of interactive recommendation algorithms is usually done using off-policy evaluation approaches [50, 52]. However, because in our problem the action space is too large (i.e. exponential in K), we utilize a simulated interaction environment for our evaluation where the simulator is built based on offline datasets. This is the evaluation setup used in similar research involving cascading bandits [19, 26, 53], which we also follow for our experiments.

We randomly divide the user profiles into training and test sets, with 50% assigned to each. The training set is used to derive known variables and generate recommendations for users. The test set is

[3] R2- Yahoo! Music, https://webscope.sandbox.yahoo.com/catalog.php?datatype=r.



(a) #clicks per round (b) n-step-regret (c) Equality$^{(B)}$ per round

(d) Equality$^{(P)}$ per round (e) Equity$^{(B)}$ per round (f) Equity^{P} per round

Figure 2: The effect of varying the degree of exploration with $\alpha \in \{0.25, 0.75, 1, 2, 5\}$ on the performance of LinUCB in terms of clicks and exposure bias on MovieLens dataset for $d = 10$ and $K = 10$. Left plot shows the average exploration across all users at each round, exploration is computed using the second term in Eq. 10. Right plots: (a) number of observed clicks in each round, b) n-step-regret as in Eq. 4, c-f) fairness metrics computed on accumulated exposure values at each round.

used to model user feedback on recommendations and generate the optimal recommendation list to evaluate model performance.

To define the merits of the items, we adhere to definitions established in the existing literature [5, 28, 39], where the relevancy of an item to users serves as its merit. To determine this, we employ a matrix factorization model on the user-item interaction data to learn the embeddings for users and items. Subsequently, we compute the relevance score between each user-item pair by taking the dot product of their embeddings. Finally, we compute the average relevance score across all users for each item, representing its merit.

We performed experiments with different dimensions of item embeddings $d \in \{10, 20\}$ and different recommendation sizes $K \in \{5, 10\}$. We tune each bandit algorithm with varying degrees of exploration $\alpha \in \{0.25, 0.75, 1, 2, 5\}$. Our EALinUCB involves a hyperparameter, the penalization coefficient γ, for which we performed a sensitivity analysis with values $\gamma \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.2\}$. We set the patient parameter β involved in the weight functions to $\beta = 0.05$ for Linear and $\beta = 0.9$ for RBP. The experiments were carried out over $n = 50,000$ rounds.

6 Results

In this section, we provide evidence and observations from our experimental results to address our three research questions[4].

6.1 (RQ1) The effect of exploration degree on exposure bias in LinUCB

RQ1 explores the relationship between the degree of exploration in the UCB item selection strategy, controlled by the hyperparameter α (Eq. 10), and its impact on exposure bias and performance of LinUCB. We examine this relationship using experimental results obtained from the MovieLens dataset for $d = 10$ and $K = 10$.

Figure 2 (left) shows the average degree of exploration among all users in each round for varying values of α. Here, exploration refers to the second term in Eq. 10, which is computed for each recommended item for each user in each round. In particular, the average exploration value is derived by averaging exploration values across all recommended items for each user, and then averaging these values across all users at each round.

Several patterns emerge from Figure 2 (left). Increasing the value of α leads to a higher degree of exploration. Secondly, the degree of exploration rapidly decreases after several rounds. For example, with $\alpha = 2$, exploration substantially decreases after approximately 1000 rounds, reaching 0 around 10,000 rounds. This behavior aligns with the exploration/exploitation trade-off in bandit algorithms [6, 8]: as the algorithm accumulates more information over time, exploration decreases and exploitation increases.

Figures 2a-2f depict the performance of on LinUCB' with varying α values. It can be observed that exploration negatively affects the accuracy of the model. Looking at Figure 2a, lower clicks are observed for LinUCB with higher α values (e.g., $\alpha = 5$) compared to lower values (e.g., $\alpha = 1$). Furthermore, in Figure 2b, LinUCB consistently exhibits a better n step-regret with lower α values (e.g., $\alpha = 1$) compared to higher values (e.g., $\alpha = 5$).

It is also evident from Figures 2c-2f that only during the exploration phase is exposure fairness improving. However, after the model stops exploring the item space, the exposure bias does not decrease any further. Although this is the normal behavior of bandit algorithms, our aim is to achieve a higher degree of fairness before the algorithm stops the exploration. In addition, the plots show that for various values of α, the degree to which the exposure bias decreases is different. Surprisingly, a higher α value does not result in a higher exposure fairness for items. Hence, this confirms the necessity of an intervention in LinUCB, as its built-in exploration component does not adequately mitigate exposure bias. These results are consistent with findings in [32, 35]. Since LinUCB with $\alpha = 0.25$ yields the best performance across all the metrics, for the rest of the analysis in this paper, we set $\alpha = 0.25$.

[4]We report partial results in this paper. The full results are available at https://github.com/masoudmansoury/ealinucb.

Table 3: Performance of our EALinUCB with three different weight functions on MovieLens and Yahoo Music datasets for $d = 10$ and $K = 5$. For all metrics, higher value is more desired. † indicates that the result is significant with $p < 0.01$.

Method	$\mathcal{F}$	ML					Yahoo Music				
		$\overline{clicks}$	Equality^B	Equality^P	Equity^B	Equity^P	$\overline{clicks}$	Equality^B	Equality^P	Equity^B	Equity^P
LinUCB	-	**0.2166**	0.329	0.3081	0.0506	0.0476	0.1802	0.3602	0.3316	0.3559	0.3278
EARSLinUCB	-	0.2165	0.3257	0.3257	0.0495	0.0495	0.1807	0.3591	0.3591	0.3556	0.3556
FRMLinUCB	-	0.2005	0.3334	0.4586	0.0504	0.0505	0.1721	0.3514	0.42	0.3514	0.3501
EALinUCB (ours)	Log	0.2105	0.3799	**0.524**†	**0.055**	**0.0546**	0.1772	**0.3662**	0.5396	**0.3599**	**0.3641**
EALinUCB (ours)	RBP	**0.2166**	0.329	0.472	0.0506	0.0515	**0.1814**	0.365	0.5812	0.358	0.362
EALinUCB (ours)	Linear	0.2069	**0.392**†	0.5142	0.0545	0.0535	0.1706	0.3661	**0.5879**†	0.3563	0.3582

Table 4: Performance of our EALinUCB with three different weight functions on MovieLens and Yahoo Music datasets for $d = 10$ and $K = 10$. For all metrics, higher value is more desired. † indicates that the result is significant with $p < 0.01$.

Method	$\mathcal{F}$	ML					Yahoo				
		$\overline{clicks}$	Equality^B	Equality^P	Equity^B	Equity^P	$\overline{clicks}$	Equality^B	Equality^P	Equity^B	Equity^P
LinUCB	-	**0.3722**	0.3974	0.3656	0.0555	0.0507	0.3154	0.4469	0.404	0.4429	0.4002
EARSLinUCB	-	0.3721	0.3974	0.3848	0.0562	0.0541	0.3157	0.4467	0.4467	0.4429	0.4429
FRMLinUCB	-	0.3574	0.4029	0.4157	0.0572	0.055	0.3085	0.4517	0.4596	0.4517	0.4498
EALinUCB (ours)	Log	0.3605	0.494	**0.514**†	**0.065**†	**0.065**†	0.3073	**0.495**†	0.5174	**0.485**†	**0.473**†
EALinUCB (ours)	RBP	**0.3722**	0.4074	0.434	0.0555	0.0557	**0.3171**	0.4489	**0.553**†	0.4614	0.4585
EALinUCB (ours)	Linear	0.3585	**0.498**†	0.5062	0.0601	0.0604	0.3008	0.4552	0.5312	0.461	0.4542

6.2 (RQ2) Comparison to baselines

To address **RQ2**, we compare the performance of our EALinUCB with other baselines using different exposure bias metrics. We set $\alpha = 0.25$ and $\gamma = 0$ for all experiments. Tables 3 and 4 present the results for $d = 10$ and $K \in \{5, 10\}$. These results indicate that EALinUCB outperforms other algorithms consistently across all exposure bias metrics. This improvement is often significant, demonstrating its effectiveness in mitigating exposure bias. Figure 3 compares the Equality$^{(P)}$ per round between EALinUCB and LinUCB for $d = 10$, $K = 10$, $\alpha = 0.25$, and $\gamma = 0$. The plot reveals that EALinUCB significantly enhances the exposure fairness in the long run, particularly with logarithmic and linear weight functions.

Improving exposure fairness involves balancing the exposure for items by downgrading over-exposed items that are often ignored by users and promoting under-exposed items that are clicked more often. To examine the ability of EALinUCB to balance exposure compared to LinUCB, we calculate the percentage change in exposure for each item assigned by EALinUCB compared to LinUCB as:

$$\Delta E(i) = \frac{E_{EALinUCB}(i) - E_{LinUCB}(i)}{\frac{E_{EALinUCB}(i) + E_{LinUCB}(i)}{2}} \times 100 \tag{23}$$

where $E_{EALinUCB}(i)$ and $E_{LinUCB}(i)$ are the exposure given to item i by EALinUCB and LinUCB, respectively. Analogously, we compute the percentage change in $\overline{clicks}$ observed in each element by EALinUCB and LinUCB. Figure 4 shows how our EALinUCB assigns exposure to each item compared to LinUCB for $E^{(P)}$ exposure definition. The x-axis displays items sorted by $E^{(P)}$ by LinUCB in descending order, the y-axis shows $\Delta E^{(P)}$ computed by Eq. 23. The color bar also shows $\Delta\overline{clicks}$.

Figure 4 indicates that EALinUCB promotes under-exposed items in LinUCB while downgrading over-exposed ones, effectively balancing exposure for different items. In addition, the red points on the upper right indicate that EALinUCB predominantly promotes relevant items, as most of the promoted items also receive more clicks. These patterns are consistent across all datasets and weight functions. Similar patterns are observed for other exposure notions (reported in this website).

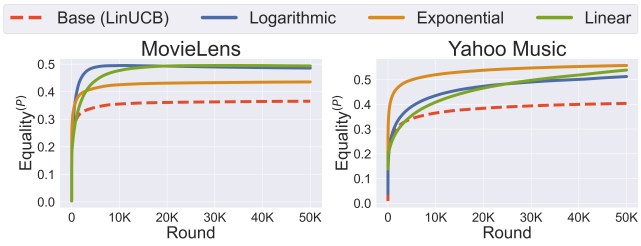
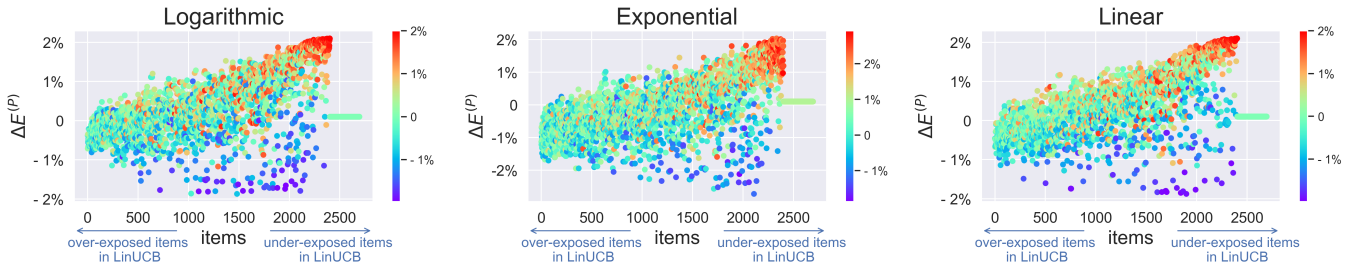


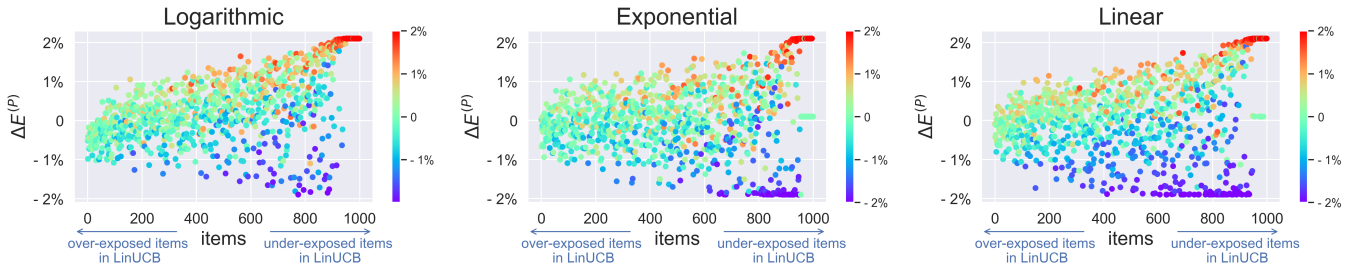
Figure 3: Comparison of LinUCB and EALinUCB with three weight functions in terms of Equality$^{(P)}$ per round for $d = 10$, $K = 10$, $\alpha = 0.25$, and $\gamma = 0$. At each round t, Equality$^{(P)}$ is computed over the accumulated exposure up to round t.

6.3 (RQ3) The impact of varying penalization degree (γ)

Our exposure-aware reward model, as defined in Eq. 8, involves a hyperparameter γ that regulates the extent of penalization for unclicked items. To explore the sensitivity of EALinUCB to different values of γ, we conducted a sensitivity analysis on the MovieLens dataset for $d = 10$ and $K = 10$, varying γ within the range $\{0, 0.001, 0.005, 0.01, 0.05, 0.1, 0.2\}$. Figure 5 presents the results of this analysis. Each plot corresponds to a specific exposure bias metric, with the $\overline{Clicks}$ values plotted on the x axis and the metric values on the y-axis. The points on the right side of the plots represent the results for $\gamma = 0$, while the points on the left show



(a) MovieLens



(b) Yahoo Music

Figure 4: Exposure analysis of our EALinUCB with three different weight functions for $d = 10$ and $K = 10$. Colorbar shows the percentage increase/decrease in $\overline{clicks}$. Items are sorted based on their exposure ($E^{(P)}$) by LinUCB in descending order from left to right where items in the left-side are the over-exposure ones and items in the right-side are under-exposed ones.

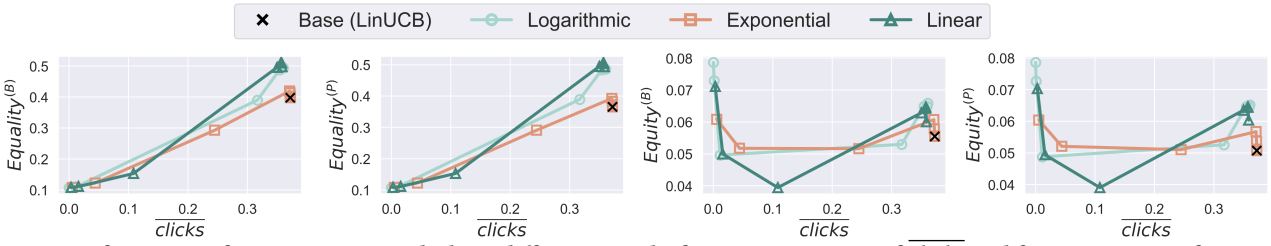


Figure 5: Performance of our EALinUCB with three different weight functions in terms of $\overline{clicks}$ and fairness metrics for varying $\gamma \in \{0, 0.001, 0.005, 0.01, 0.05, 0.1, 0.2\}$ on MovieLens dataset for $d = 10$ and $K = 10$. The cross shows the performance of LinUCB.

the results for $\gamma = 0.2$. The crosses represent the performance of LinUCB as the base algorithm.

The results show that with a proper choice of γ value, the penalization term can have a positive impact on mitigating exposure bias. For example, in all plots corresponding to the exponential weight function, increasing the value of γ from 0 to 0.005 leads to improvements in all exposure metrics. However, further increasing the value of this hyperparameter results in a decrease in performance. When $\gamma = 0.2$, for example, $\overline{clicks}$ approaches 0, indicating deteriorating performance, along with reductions in exposure metrics, which means increased exposure bias.

The observed trend can be attributed to the dominance of the penalization term in the learning process, especially with higher values of γ. When γ is large, the algorithm predominantly learns negative preferences due to the abundance of unclicked items compared to clicked items. Consequently, the performance of EALinUCB declines. Hence, careful tuning of this hyperparameter is essential to optimize the algorithm performance.

7 Conclusion and Future Work

In this paper, we studied the problem of exposure bias in linear cascading bandits. Although these algorithms partially mitigate exposure bias during the initial exploration phase, we show their limitations in balancing item exposure over the recommendation lifecycle. To improve exposure fairness throughout the recommendation process, we introduced an *exposure-aware reward model* and integrated it into the linear cascading bandit. This model leverages user feedback and item position in the recommendation list to reward clicked items and penalize unclicked ones. Our extensive experiments demonstrated the effectiveness of the proposed exposure-aware reward model in mitigating exposure bias while preserving recommendation accuracy. Additionally, we theoretically derived a gap-free bound on the n-step-regret for our exposure-aware cascading bandit. In future work, we plan to extend our analysis to other cascading bandits [19, 26] as well as broader classes of bandit algorithms like those based on Thompson Sampling.

Acknowledgments

This project was funded by Elsevier's Discovery Lab.

References

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. 2011. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems* 24 (2011).

[2] Charu C Aggarwal et al. 2016. *Recommender systems*. Vol. 1. Springer.

[3] Arda Antikacioglu and R Ravi. 2017. Post processing recommender systems for diversity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 707–716.

[4] Peter Auer. 2000. Using upper confidence bounds for online learning. In *Proceedings 41st annual symposium on foundations of computer science*. IEEE, 270–279.

[5] Aparna Balagopalan, Abigail Z Jacobs, and Asia J Biega. 2023. The role of relevance in fair ranking. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2650–2660.

[6] Andrea Barraza-Urbina. 2017. The exploration-exploitation trade-off in interactive recommender systems. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. 431–435.

[7] Robin Burke, Alexander Felfernig, and Mehmet H Göker. 2011. Recommender systems: An overview. *Ai Magazine* 32, 3 (2011), 13–18.

[8] H Henry Cao, Liye Ma, Z Eddie Ning, and Baohong Sun. 2024. How does competition affect exploration vs. exploitation? a tale of two recommendation algorithms. *Management Science* 70, 2 (2024), 1029–1051.

[9] Olivier Chapelle and Lihong Li. 2011. An empirical evaluation of thompson sampling. *Advances in neural information processing systems* 24 (2011).

[10] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2023. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems* 41, 3 (2023), 1–39.

[11] Giovanni Luca Ciampaglia, Azadeh Nematzadeh, Filippo Menczer, and Alessandro Flammini. 2018. How algorithmic popularity bias hinders or promotes quality. *Scientific reports* 8, 1 (2018), 1–7.

[12] Andrew Collins, Dominika Tkaczyk, Akiko Aizawa, and Joeran Beel. 2018. Position bias in recommender systems for digital libraries. In *International Conference on Information*. Springer, 335–344.

[13] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In *Proceedings of the 2008 international conference on web search and data mining*. 87–94.

[14] Laura Granka, Matthew Feusner, and Lori Lorigo. 2008. Eyetracking in online search. *Passive eye monitoring* (2008), 283–304.

[15] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[16] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 5–53.

[17] Maria Heuss, Daniel Cohen, Masoud Mansoury, Maarten de Rijke, and Carsten Eickhoff. 2023. Predictive uncertainty-based bias mitigation in ranking. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 762–772.

[18] Maria Heuss, Fatemeh Sarvi, and Maarten de Rijke. 2022. Fairness of exposure in light of incomplete exposure estimation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 759–769.

[19] Gaurush Hiranandani, Harvineet Singh, Prakhar Gupta, Iftikhar Ahamath Burhanuddin, Zheng Wen, and Branislav Kveton. 2020. Cascading linear submodular bandits: Accounting for position bias and diversity in online learning to rank. In *Uncertainty in Artificial Intelligence*. PMLR, 722–732.

[20] Katja Hofmann, Anne Schuth, Alejandro Bellogin, and Maarten de Rijke. 2014. Effects of position bias on click-based recommender evaluation. In *European Conference on Information Retrieval*. Springer, 624–630.

[21] Jin Huang, Harrie Oosterhuis, Masoud Mansoury, Herke van Hoof, and Maarten de Rijke. 2024. Going Beyond Popularity and Positivity Bias: Correcting for Multifactorial Bias in Recommender Systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 416–426.

[22] Olivier Jeunen and Bart Goethals. 2021. Top-k contextual bandits with equity of exposure. In *Fifteenth ACM Conference on Recommender Systems*. 310–320.

[23] Branislav Kveton, Csaba Szepesvari, Zheng Wen, and Azin Ashkan. 2015. Cascading bandits: Learning to rank in the cascade model. In *International conference on machine learning*. PMLR, 767–776.

[24] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. 2015. Combinatorial cascading bandits. *Advances in Neural Information Processing Systems* 28 (2015).

[25] Tze Leung Lai and Herbert Robbins. 1985. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics* 6, 1 (1985), 4–22.

[26] Chang Li, Haoyun Feng, and Maarten de Rijke. 2020. Cascading hybrid bandits: Online learning to rank for relevance and diversity. In *Fourteenth ACM Conference on Recommender Systems*. 33–42.

[27] Shuai Li, Baoxiang Wang, Shengyu Zhang, and Wei Chen. 2016. Contextual combinatorial cascading bandits. In *International conference on machine learning*. PMLR, 1245–1253.

[28] Yuanna Liu, Ming Li, Mozhdeh Ariannezhad, Masoud Mansoury, Mohammad Aliannejadi, and Maarten de Rijke. 2024. Measuring Item Fairness in Next Basket Recommendation: A Reproducibility Study. In *European Conference on Information Retrieval*. Springer, 210–225.

[29] Lori Lorigo, Maya Haridasan, Hrönn Brynjarsdóttir, Ling Xia, Thorsten Joachims, Geri Gay, Laura Granka, Fabio Pellacini, and Bing Pan. 2008. Eye tracking and online search: Lessons learned and challenges ahead. *Journal of the American Society for Information Science and Technology* 59, 7 (2008), 1041–1052.

[30] Linyuan Lü, Matúš Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and Tao Zhou. 2012. Recommender systems. *Physics reports* 519, 1 (2012), 1–49.

[31] Masoud Mansoury. 2021. Understanding and Mitigating Multi-Sided Exposure Bias in Recommender Systems. *PhD Dissertation, Eindhoven University of Technology* (2021).

[32] Masoud Mansoury, Himan Abdollahpouri, Bamshad Mobasher, Mykola Pechenizkiy, Robin Burke, and Milad Sabouri. 2021. Unbiased cascade bandits: Mitigating exposure bias in online learning to rank recommendation. *arXiv preprint arXiv:2108.03440* (2021).

[33] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. 2020. Feedback loop and bias amplification in recommender systems. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 2145–2148.

[34] Masoud Mansoury and Bamshad Mobasher. 2023. Fairness of exposure in dynamic recommendation. *CONSEQUENCE'23 Workshop on Causality, Counterfactuals, and Sequential Decision-Making in conjunction with ACM RecSys 2023* (2023).

[35] Masoud Mansoury, Bamshad Mobasher, and Herke van Hoof. 2022. Exposure-Aware Recommendation using Contextual Bandits. *5th FAccTRec Workshop on Responsible Recommendation in conjunction with ACM RecSys 2022* (2022).

[36] James McInerney, Benjamin Lacker, Samantha Hansen, Karl Higley, Hugues Bouchard, Alois Gruson, and Rishabh Mehrotra. 2018. Explore, exploit, and explain: personalizing explainable recommendations with bandits. In *Proceedings of the 12th ACM conference on recommender systems*. 31–39.

[37] Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Fernando Diaz. 2018. Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In *Proceedings of the 27th acm international conference on information and knowledge management*. 2243–2251.

[38] Alistair Moffat and Justin Zobel. 2008. Rank-biased precision for measurement of retrieval effectiveness. *ACM Transactions on Information Systems (TOIS)* 27, 1 (2008), 1–27.

[39] Amifa Raj and Michael D Ekstrand. 2022. Measuring fairness in ranked results: An analytical and empirical comparison. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 726–736.

[40] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*. 175–186.

[41] Guy Shani and Asela Gunawardana. 2011. Evaluating recommendation systems. *Recommender systems handbook* (2011), 257–297.

[42] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2219–2228.

[43] Ayan Sinha, David F Gleich, and Karthik Ramani. 2016. Deconvolving feedback loops in recommender systems. *Advances in neural information processing systems* 29 (2016).

[44] Ramakrishnan Srikant, Sugato Basu, Ni Wang, and Daryl Pregibon. 2010. User browsing models: relevance versus examination. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. 223–232.

[45] Tom Sühr, Asia J Biega, Meike Zehlike, Krishna P Gummadi, and Abhijnan Chakraborty. 2019. Two-sided fairness for repeated matchings in two-sided markets: A case study of a ride-hailing platform. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3082–3092.

[46] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[47] William R Thompson. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25, 3-4 (1933), 285–294.

[48] Saúl Vargas and Pablo Castells. 2014. Improving sales diversity by recommending users to items. In *Proceedings of the 8th ACM Conference on Recommender systems*. 145–152.

[49] Lequn Wang, Yiwei Bai, Wen Sun, and Thorsten Joachims. 2021. Fairness of exposure in stochastic bandits. In *International Conference on Machine Learning*. PMLR, 10686–10696.

[50] Yu-Xiang Wang, Alekh Agarwal, and Miroslav Dudık. 2017. Optimal and adaptive off-policy evaluation in contextual bandits. In *International Conference on Machine*

Learning. PMLR, 3589–3597.

[51] Meike Zehlike and Carlos Castillo. 2020. Reducing disparate exposure in ranking: A learning to rank approach. In *Proceedings of the web conference 2020*. 2849–2855.

[52] Ruohan Zhan, Vitor Hadad, David A Hirshberg, and Susan Athey. 2021. Off-policy evaluation via adaptive weighting with data from contextual bandits. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2125–2135.

[53] Shi Zong, Hao Ni, Kenny Sung, Nan Rosemary Ke, Zheng Wen, and Branislav Kveton. 2016. Cascading bandits for large-scale recommendation problems. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*. 835–844.