Observation-Free Attacks on Online Learning to Rank

Sameep Chattopadhyay[†] **Nikhil Karamchandani**[◇] **Sharayu Moharir**[◇]

[†]Paul G. Allen School of Computer Science & Engineering, University of Washington
[◇]Department of Electrical Engineering, Indian Institute of Technology Bombay
sameepch@uw.edu nikhilk@ee.iitb.ac.in sharayum@ee.iitb.ac.in

Abstract

Online learning to rank (OLTR) plays a critical role in information retrieval and machine learning systems, with a wide range of applications in search engines and content recommenders. However, despite their extensive adoption, the susceptibility of OLTR algorithms to coordinated adversarial attacks remains poorly understood. In this work, we present a novel framework for attacking some of the widely used OLTR algorithms. Our framework is designed to promote a set of target items so that they appear in the list of top-K recommendations for $T - o(T)$ rounds, while simultaneously inducing linear regret in the learning algorithm. We propose two novel attack strategies: `CascadeOFA` for CascadeUCB1 and `PBMOFA` for PBM-UCB. We provide theoretical guarantees showing that both strategies require only $O(\log T)$ manipulations to succeed. Additionally, we supplement our theoretical analysis with empirical results on real-world data.

1 Introduction

Online Learning to Rank (OLTR) [1] is a sequential decision-making framework widely used in information retrieval systems [2] to rank items based on user feedback. In an OLTR setup, the learning agent presents a ranked list of items to the user in each round, and the user provides implicit feedback by interacting with the list in some manner. One of the most widely studied forms of such feedback is click behavior, which the agent seeks to model and optimize in order to improve its recommendations. Over the years, several click feedback models have been studied, with two of the most prominent being the *Cascade* model [3], where the user examines the list sequentially from top to bottom and clicks the first item they find "attractive", therefore resulting in a single click per round, and the *Position-based model* [4], which allows multiple clicks in a round and assumes that the probability of an item being examined depends on its position in the list.

Due to the sequential nature of OLTR, several studies have attempted to reformulate the problem within the framework of *Multi-Armed Bandits* (MABs) [5]. Recent works have developed efficient learning algorithms for the Cascade [6, 7, 8] and the Position-based [9] feedback models. Additionally, more general algorithms that accommodate a broader class of user feedback structures, encompassing both the above models, have also been proposed and analyzed [10, 11].

Given the widespread use of online decision-making systems, understanding their response to coordinated attacks is crucial. While extensive research has explored adversarial attack strategies for the MABs, with frameworks like reward manipulation and action manipulation [12, 13, 14, 15, 16], relatively little attention has been given to such attacks in the OLTR setting [17, 18].

This work focuses on a specific class of reward manipulation strategies known as observation-free attacks [13], where a collective adversary generates (or manipulates) reward signals to promote a set of pre-determined items in a learning algorithm without having access to the algorithm's feedback. A key feature of these observation-free attacks is that they can be executed in the real world by groups of ordinary individuals who, in spite of lacking access to the feedback from other users, can influence the algorithm to promote unattractive yet socially beneficial content through their own coordinated actions. This makes them a compelling instance of *algorithmic collective action* [19, 20].

39th Conference on Neural Information Processing Systems (NeurIPS 2025)
Workshop on Algorithmic Collective Action (ACA@NeurIPS 2025).

Despite their real-world applications and proven success for MABs [13], no effective observation-free attack strategy has yet been demonstrated for even the most common OLTR algorithms [18].

1.1 Our Contributions

In this paper, we present the first study on observation-free attacks against OLTR algorithms with click feedback. Our technical contributions are summarized as follows. In Section 3, we introduce a general framework for a limited-time observation-free attack strategy to promote a set of target items in OLTR. Based on this framework, we propose two novel attack strategies, namely `CascadeOFA` and `PBMOFA` for the Cascade and the Position-based feedback models, respectively. We prove that both attack strategies require just $O(\log T)$ reward manipulations to successfully promote their target elements for $T - o(T)$ rounds and impose $\Omega(T)$ regret on their respective learning algorithms. Our work also addresses some of the major drawbacks of the earlier attack strategies for OLTR, specifically the need for continuous manipulation of rewards in [17] and the inability to attack UCB-based OLTR algorithms through reward manipulation in [18]. We supplement our analysis with empirical evaluation of our strategies using the MovieLens dataset [21].

1.2 Related Work

Numerous recent studies have explored adversarial attacks on classical multi-armed bandit algorithms [12, 13, 14, 15, 16]. However, adversarial attacks on OLTR [17, 18] remain relatively unexplored.

While most of the attacks on MABs could potentially be extended to OLTR, the combinatorial action space and restricted feedback structure in OLTR make these extensions non-trivial [17]. To the best of our knowledge, only two studies have investigated adversarial attacks on OLTR algorithms. The first one [17], introduced a reward manipulation attack on CascadeUCB1 and PBM-UCB, demonstrating that a target item could be recommended for $T - o(T)$ rounds with only $o(T)$ corruptions. A key limitation of this attack was its reliance on observing rewards and performing costly computations in every round. The second study [18] proposed an attack-then-quit (ATQ) strategy for OLTR algorithms based on item elimination. The study also discusses the hindrances in applying ATQ-like attacks to UCB-based OLTR algorithms as one of its major drawbacks.

2 Setting

In this section, we discuss our problem setting for designing an observation-free attack on OLTR algorithms. For the ease of notation, we define the following quantities. Let the universal set of all available items be $\mathcal{A} = \{1, 2, \ldots, L\}$; at each time-step (round) t, the user is presented with an ordered list $\mathcal{L}_t = (a_{1,t}, \ldots, a_{K,t})$ of K items selected from $\mathcal{A}$, where $K \leqslant L$. Here, $a_{i,t}$ is the item present at the i^{th} position of $\mathcal{L}_t$, i.e., $\mathcal{L}_t[i] = a_{i,t}$. The user examines $\mathcal{L}_t$ and provides feedback to the OLTR algorithm through clicks, following a specified click feedback model.

2.1 Click Feedback Models

This work primarily focuses on two of the most common click feedback models for OLTR: the Cascade model [3] and the Position-based model [4]. In both models, the items are characterized by attraction probabilities $w \in [0, 1]^L$. When a user examines an item $a = a_{i,t}$ located at position i in list $\mathcal{L}_t$ during round t, the probability of clicking on it is given by w_a, which is independent of previous rounds and the attraction probability of other items. To formally characterize the user feedback, we define the following quantities:

- *Examination Feedback*: $\mathrm{X}_t \in \{0, 1\}^K$, with $\mathrm{X}_{i,t} = 1$, if and only if $a_{i,t}$ is examined in round t.
- *Click Feedback*: $\mathcal{C}_t \in \{0, 1\}^K$, where $\mathcal{C}_{i,t} = 1$, if and only if $a_{i,t}$ is clicked in round t.

Using the above-defined quantities, we now describe the following click feedback models in detail.

2.1.1 Cascade Model

The Cascade model [3] is one of the earliest and most well-studied user feedback models. In this model, the user examines $\mathcal{L}_t$ sequentially from top to bottom, selecting the first item they find *attractive*, which, in the context of web search, is indicated by a click. Once an item $a_{i,t}$ is clicked, the user concludes the search, leaving all the subsequent items unexamined for that round. Conversely, if $a_{i,t}$ fails to attract the user, they proceed to examine the next item $a_{i+1,t}$, and continue to do so until the first attractive item is found or $\mathcal{L}_t$ is completely examined. The Cascade model assumes that only a single item can be clicked in any given round.

2.1.2 Position-based Model (PBM)

The *Position-based model* [4] serves as a prominent alternative to the Cascade model for simulating user behavior in OLTR. Unlike the Cascade model, where the items are sequentially examined, the PBM suggests that the likelihood of examination is influenced by the *position bias*, captured by $\mathcal{P} = (p_1, \ldots, p_K)$. Specifically, an item appearing in the i^{th} position of $\mathcal{L}_t$ is examined with a probability p_i, independently of the other positions and their corresponding items. The examination probabilities are assumed to be constant through time and decreasing as the position moves down the list, with $p_1 \geqslant p_2 \geqslant \ldots \geqslant p_K$.

In PBM, for item $a = a_{i,t}$, the click feedback is given by $\mathcal{C}_{i,t} = \mathrm{X}_{i,t} \cdot Y_{a,t}$, where $\mathrm{X}_{i,t} \sim \mathrm{Bernoulli}(p_i)$ represents the examination feedback of $a_{i,t}$ in round t, while $Y_{a,t} \sim \mathrm{Bernoulli}(w_a)$ reflects the clicking of item $a_{i,t}$, conditioned on it being examined. A key difference between the PBM and the Cascade model is that PBM allows for multiple items in $\mathcal{L}_t$ to be clicked within a single round.

2.2 OLTR with Click Feedback

For real-world use cases, the attractiveness of the items is often unknown, and the recommenders have to learn them from the user feedback, which brings us to the problem of online learning to rank. As per the OLTR setup, at each time step t, the learning agent recommends a list $\mathcal{L}_t$ to the user and observes the corresponding feedback. If a user clicks on the item $a = a_{i,t}$, i.e., $\mathcal{C}_{i,t} = 1$, the agent receives a click feedback for the position, and a reward $\mathrm{Z}_{a,t} = 1$ for the item. If an item is not clicked, the agent receives $\mathrm{Z}_{a,t} = 0$. Therefore, in the absence of any external manipulations,

$$\mathrm{Z}_{a,t} = \sum_{i=1}^{K} \mathcal{C}_{i,t} \cdot \mathbb{1}\{a = a_{i,t}\}.$$

Given the attraction probabilities w, the expected reward in round t is denoted by $f(\mathcal{L}_t, w)$, which equals the expected number of clicks received by the given list:

$$f(\mathcal{L}_t, w) = \mathbb{E}\left[\sum_{a \in \mathcal{L}_t} \mathrm{Z}_{a,t}\right] = \mathbb{E}\left[\sum_{a \in \mathcal{L}_t} \sum_{i=1}^{K} \mathcal{C}_{i,t} \cdot \mathbb{1}\{a = a_{i,t}\}\right] = \mathbb{E}\left[\|\mathcal{C}_t\|_1\right].$$

At each round, the learning agent aims to maximize its expected reward by recommending the most attractive items as a part of $\mathcal{L}_t$. Without loss of generality, let the items in $\mathcal{A}$ be indexed in decreasing order of their attraction probabilities. We define $\mathcal{L}^* = (1, \ldots, K)$ as the list of the K most attractive items, ordered by descending attractiveness. An optimal static policy, recommending $\mathcal{L}^*$ to the user at all rounds, maximizes the expected reward under both the click models. Any learning policy is evaluated on the *cumulative expected regret* up to horizon T, which is given by

$$\mathcal{R}(T) = \mathbb{E}\left[\sum_{t=1}^{T} \mathrm{R}(\mathcal{L}_t, w)\right], \text{ where } \mathrm{R}(\mathcal{L}_t, w) = f(\mathcal{L}^*, w) - f(\mathcal{L}_t, w).$$

Over the past few years, various algorithms [6, 7, 9, 10, 11] have been proposed for OLTR with click feedback. Details for two such UCB-based OLTR algorithms, namely CascadeUCB1 (Algorithm 3) for the Cascade model and PBM-UCB (Algorithm 4) for PBM, are provided in Appendix A.

2.3 Attacking OLTR Algorithms

Since OLTR algorithms optimize exclusively for click feedback, they often overlook items that are less likely to attract clicks, regardless of their other attributes. Our study examines strategies that a group of adversarial users could employ to promote such less attractive items in OLTR. We focus on a class of adversarial strategies known as *observation-free attacks* [13], in which users manipulate their reward vector $\mathrm{Z}_t = \{\mathrm{Z}_{a,t}\}_{a \in \mathcal{A}}$ into $\hat{\mathrm{Z}}_t$ without access to any feedback or rewards from other users (i.e., X_t, $\mathcal{C}_t$, Z_t for $t \in \{1, 2 \ldots, T\}$), in order to promote their target items.

3 Observation-Free Attack Strategies for OLTR

Our attack strategies for OLTR are inspired by the attack given in [13] for the classical MAB algorithms. Given a set $\tilde{\Gamma}$ of target items containing $N \leqslant K$ items, we propose a three-phase attack framework. The attack is initialized with a list $\tilde{\mathcal{L}}$ that includes all items in $\tilde{\Gamma}$, together with $K - N$

items arbitrarily selected from $\mathcal{A} \setminus \tilde{\Gamma}$, and it begins at the very start of the learning process. In the first phase, lasting C_1 rounds, the collective adversary transmits zero rewards for all items. In the second phase, for the next C_2 rounds, it assigns positive rewards to a subset of items in $\tilde{\mathcal{L}}$ while forcing zero rewards for the rest. In the third phase, the adversary applies no further manipulations.

The first phase of the attack forces all items in $\mathcal{A}$ to have low empirical reward estimates, while the second phase ensures that the items in $\tilde{\mathcal{L}}$ are brought into $\mathcal{L}_t$ and are significantly differentiated from the rest. The attack actively occurs only in the first $\mathrm{C}_1 + \mathrm{C}_2$ rounds, while ensuring with high probability that the OLTR algorithm recommends a permutation of $\tilde{\mathcal{L}}$ to the users in all the subsequent rounds until horizon T. Assuming $\mathrm{C}_1 + \mathrm{C}_2 \ll T$, such an attack can be conceptualized as the rapid injection of misleading feedback by a group of adversarial users at the start of the learning process.

Algorithm 1 `CascadeOFA`

Input: Horizon T, Item set $\mathcal{A}$, and Target set $\tilde{\Gamma}$.
Initialize list $\tilde{\mathcal{L}}$ containing all elements in $\tilde{\Gamma}$.
Calculate $\mathrm{w_m}$, C_1 and C_2 as per Section 3.1.1.
for $t = 1 \dots T$ **do**
  if $t \leqslant \mathrm{C}_1$ **then**
    $\hat{\mathrm{Z}}_{a,t} = 0 \; \forall a \in \mathcal{A}$
  else if $\mathrm{C}_1 < t \leqslant \mathrm{C}_1 + \mathrm{C}_2$ **then**
    $i = \left\lceil \frac{K(t-\mathrm{C}_1)}{\mathrm{C}_2} \right\rceil$
    Set $\hat{\mathrm{Z}}_{a,t} = \mathbb{1}\{a = \tilde{\mathcal{L}}[i], a \in \mathcal{L}_t\}, \; \forall a$
  else
    $\hat{\mathrm{Z}}_t = \mathrm{Z}_t$
  end if
end for

We implement the aforementioned framework for designing attacks on two UCB-based OLTR algorithms: CascadeUCB1 [6] and PBM-UCB [9]. UCB-based OLTR algorithms are known to be quite resistant to such limited-time attack strategies [18]. This resistance arises because, once the adversary stops, the true feedback for other items is gradually revealed to the algorithm, allowing the learner to recognize that the targeted items are not the most attractive ones. To prevent this, by the end of the second phase, our framework ensures that all the non-target items have sufficiently low UCBs, and thus, with a high probability, they do not receive any further examinations (and clicks) even after the manipulations conclude.

3.1 Attacking CascadeUCB1

To efficiently attack the CascadeUCB1 algorithm and promote items from a given target set $\tilde{\Gamma}$, we propose the `CascadeOFA` strategy outlined in Algorithm 1, with parameters specified in Section 3.1.1. The analytical results for this strategy are presented in Theorem 3.1.

3.1.1 CascadeOFA: Skeleton and Details

`CascadeOFA` requires initializing a list $\tilde{\mathcal{L}}$ of length K that contains all the items of $\tilde{\Gamma}$. Following this, we define an attack parameter $\mathrm{w_m}$, which is used to specify the phase durations, with

$$\mathrm{w_m} = (1-\epsilon)\min\left\{\frac{1}{K}, w_{\min}\right\}, \text{where } \epsilon \in (0,1) \text{ and } w_{\min} = \min_{a \in \tilde{\mathcal{L}}} w_a.$$

Note that `CascadeOFA` does not require knowledge of the exact values of w_a's; any $\mathrm{w_m}$ smaller than $\min\{1/K, w_{\min}\}$ is sufficient to launch the attack. An observation-free attack on CascadeUCB1 with a parameter $\alpha > 1$ (described in Remark A.1) proceeds in the following manner:

Phase 1. Set $\hat{\mathrm{Z}}_{a,t} = 0 \; \forall \; a \in \mathcal{A}$ for the first C_1 rounds, where $\mathrm{C}_1 = L\left\lceil \frac{\alpha \log T}{K \mathrm{w_m^2}} \right\rceil$.

Phase 2. The second phase lasts for C_2 rounds with $\mathrm{C}_2 = K\left\lceil \frac{\mathrm{w_m} K \mathrm{C}_1/L + L - K + 1}{1 - K\mathrm{w_m}} \right\rceil$.

Split this phase into K sub-phases of equal length with the i^{th} sub-phase lasting from round $\mathrm{C}_1 + (i-1)\mathrm{C}_2/K + 1$ to $\mathrm{C}_1 + i\mathrm{C}_2/K$. In the i^{th} sub-phase, enforce $\hat{\mathrm{Z}}_{a,t} = \mathbb{1}\{a = \tilde{\mathcal{L}}[i], a \in \mathcal{L}_t\}$.

Phase 3. No further reward manipulation is applied in the remaining rounds of CascadeUCB1.

3.1.2 Theoretical Analysis

Theorem 3.1. *If a collective adversary attacks CascadeUCB1 using the* `CascadeOFA` *strategy outlined in Algorithm 1, then with $O(\log T)$ reward manipulation, it can ensure that each item $a \in \tilde{\Gamma}$ is recommended for at least $T - O(\log T)$ rounds, with probability exceeding $1 - \frac{K}{T}$, thereby imposing $\mathcal{R}(T) = \Omega(T)$ (linear regret) on CascadeUCB1.*

A detailed proof of Theorem 3.1 is provided in Appendix B.1.

3.2 Attacking PBM-UCB

Following the approach of CascadeOFA, we propose the PBMOFA strategy outlined in Algorithm 2 for attacking PBM-UCB with exploration parameter $\alpha > 1$ (Remark A.1) and position bias $\mathcal{P} = (p_1, \ldots, p_K)$; we define $\lambda_p = p_1/p_K$.

At each round of the attack, the adversary generates $\hat{\mathrm{X}}_t \in \{0,1\}^K$ with $\hat{\mathrm{X}}_{i,t} \sim \text{Bernoulli}(p_i)$ to examine $\mathcal{L}_t$ according to PBM.

Similar to CascadeOFA, this attack also requires a parameter $\mathrm{w_m} \in (0, \min\{1/\lambda_p, w_{\min}\})$. Owing to the unique characteristics of position-based models, the computation of C_1 and C_2 differs markedly from that for the Cascade model, as detailed in Section 3.2.1. The consequences of the PBMOFA strategy on PBM-UCB are presented in Theorem 3.2.

Algorithm 2 PBMOFA

Input: Position bias $\mathcal{P}$, Horizon T, Item set $\mathcal{A}$, Target set $\tilde{\Gamma}$.
Initialize list $\tilde{\mathcal{L}}$ containing all elements in $\tilde{\Gamma}$.
Calculate C_1 and C_2 as per Section 3.2.1.
for $t = 1 \ldots T$ **do**
  $\hat{\mathrm{X}}_{i,t} \sim \text{Bernoulli}(p_i)\ \forall i = 1, \ldots, K$
  if $t \leqslant \mathrm{C}_1$ **then**
    $\hat{\mathrm{Z}}_{a,t} = 0\ \forall a \in \mathcal{A}$
  else if $\mathrm{C}_1 < t \leqslant \mathrm{C}_1 + \mathrm{C}_2$ **then**
    $\hat{\mathrm{Z}}_{a,t} = \sum_{i=1}^{K} \hat{\mathrm{X}}_{i,t} \cdot \mathbb{1}\{a = a_{i,t}, a \in \tilde{\mathcal{L}}\}$
  else
    $\hat{\mathrm{Z}}_t = \mathrm{Z}_t$
  end if
end for

3.2.1 PBMOFA: Skeleton and Details

Phase 1. Set $\hat{\mathrm{Z}}_{a,t} = 0\ \forall\ a \in \mathcal{A}$ for the first C_1 rounds where $\mathrm{C}_1 = \left\lceil \frac{L}{K} \left\{ \frac{\lambda_p^2 \alpha \log T}{\mathrm{w_m^2} p_K^2} + 1 \right\} \right\rceil$.

Phase 2. Enforce $\hat{\mathrm{Z}}_{a,t} = \sum_{i=1}^{K} \hat{\mathrm{X}}_{i,t} \cdot \mathbb{1}\left\{a = a_{i,t}, a \in \tilde{\mathcal{L}}\right\}$, for the next C_2 rounds where

$$\mathrm{C}_2 = \left\lceil \frac{L\left(\lambda_p^2 \gamma \log T + 1\right)}{K} \right\rceil, \text{with } \gamma = \frac{2\rho\eta + 1 + \sqrt{4\rho\eta + 1}}{2\eta^2},\ \rho = p_1 \mathrm{w_m} \left\{ \frac{K \mathrm{C}_1}{\log T} - \frac{(L-1)\alpha}{\mathrm{w_m^2} p_K^2} \right\},$$

and $\eta = p_K - p_1 \mathrm{w_m}$.

Phase 3. The third phase of PBMOFA proceeds without any further reward manipulation.

In contrast to the earlier reward-manipulation attacks on PBM-UCB [17, 18], PBMOFA explicitly incorporates the probabilistic examination structure of the Position-based model while generating $\hat{\mathrm{X}}_t$.

3.2.2 Theoretical Analysis

Theorem 3.2. *If a collective adversary attacks PBM-UCB using the* PBMOFA *strategy outlined in Algorithm 2, then with $O(\log T)$ reward manipulations, it can ensure that each item $a \in \tilde{\Gamma}$ is recommended to the users for at least $T - O(\log T)$ rounds, with probability exceeding $1 - \frac{2K}{T}$, thereby imposing $\mathcal{R}(T) = \Omega(T)$ (linear regret) on PBM-UCB.*

A detailed proof of Theorem 3.2 is provided in Appendix B.2.

4 Empirical Results

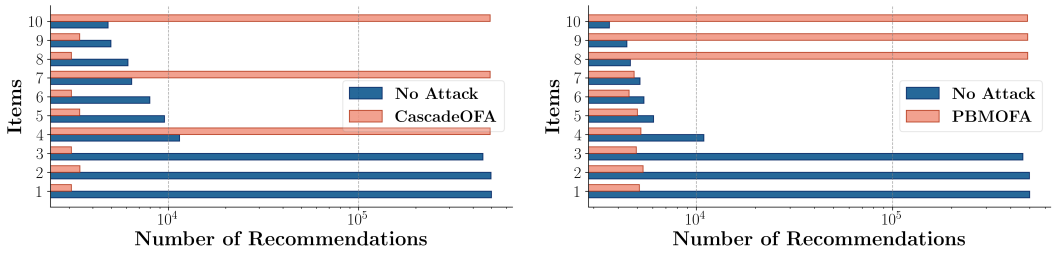
To support our theoretical analysis, we conduct experiments on the MovieLens dataset [21], containing ratings for about 3,900 movies. Following [8], we assign a reward of one when a user rates a movie above three stars. For a given L, $\mathcal{A}$ is created by arbitrarily choosing L movies from the dataset.

In all experiments, we set $T = 5 \times 10^5$, $K = 3$, $L = 10$, and average results over 50 runs. For PBM, we use $\mathcal{P} = (0.95, 0.90, 0.85)$. With $\tilde{\mathcal{L}} = (4, 7, 10)$ for CascadeOFA and $\tilde{\mathcal{L}} = (8, 9, 10)$ for PBMOFA, we obtain $\mathrm{w_m} \approx 0.08$ for both strategies. Under these parameters, the required reward manipulation $(\mathrm{C}_1 + \mathrm{C}_2)$ is 11265 for CascadeOFA and 12811 for PBMOFA, both well under 3% of the total number of rounds. To evaluate the effectiveness of our attacks, we compare the number of recommendations for the items in $\mathcal{L}^* = (1, 2, 3)$ and $\tilde{\mathcal{L}}$, with and without manipulation.

Figure 1 presents the results for CascadeOFA and PBMOFA; in the absence of any manipulation, both CascadeUCB1 and PBM-UCB recommend the items in $\mathcal{L}^*$ for most rounds. In contrast, with our attacks, the target items receive the maximum recommendations for both the OLTR algorithms.

Next, we compare the performance of CascadeOFA with the following baselines:

1. **No Attack**: The CascadeUCB1 algorithm without any external manipulation.
2. **CascadeATQ**: A naive attack-then-quit strategy, introduced by [18], with the same amount of reward manipulation as `CascadeOFA`. For $t \in \{1, \dots C_1 + C_2\}$, the adversary clicks $\left(\hat{Z}_{a,t} = 1\right)$ the top-most target item in $\mathcal{L}_t$; if no such items exist, it ignores $\left(\hat{Z}_{a,t} = 0\right)$ all the items in $\mathcal{L}_t$.
3. **CascadeAlphaAtk** (Algorithm 4 in [17]): First reward-manipulation attack on CascadeUCB1.



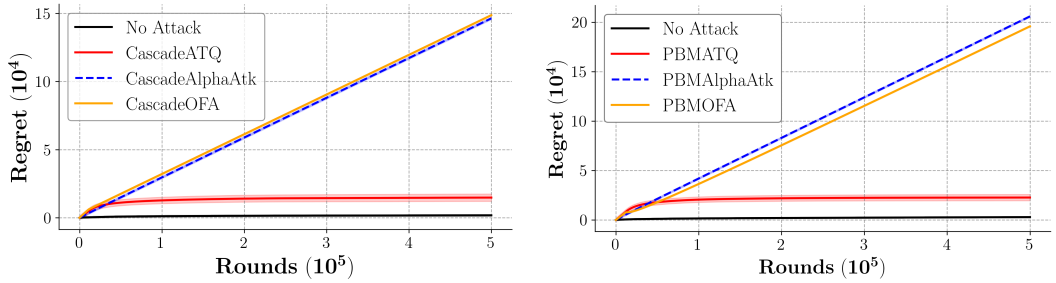
(a) `CascadeOFA` with $\tilde{\mathcal{L}} = (4, 7, 10)$. (b) `PBMOFA` with $\tilde{\mathcal{L}} = (8, 9, 10)$

Figure 1: Comparison of the number of recommendations for target items with and without attack.

We similarly compare the performance of `PBMOFA` with PBMAlphaAtk (Algorithm 2 in [17]) and a naive PBMATQ strategy. The naive strategy is synonymous with that of the collective adversary clicking all the examined target items while ignoring the rest, i.e., $\hat{Z}_{a,t} = \sum_{i=1}^{K} \hat{X}_{i,t} \cdot \mathbb{1}\left\{a = a_{i,t}, a \in \tilde{\mathcal{L}}\right\}$, in each round until $t = C_1 + C_2$. Additional details on the reward manipulations required for all the attack strategies discussed above are provided in Appendix D.

Figure 2 compares the regret enforced by the aforementioned attacks over CascadeUCB1, and PBM-UCB. Both CascadeAlphaAtk and `CascadeOFA` impose linear regret on CascadeUCB1, whereas CascadeATQ fails to do so despite having the same amount of reward manipulations as `CascadeOFA`. Similar trends follow for the Position-based model as well.

It is important to note that comparing CascadeAlphaAtk and PBMAlphaAtk with other strategies is inherently unfair, as they have access to significantly more information in the form of user feedback and rewards. Both attack strategies operate online, with actions determined by user feedback.



(a) Under CascadeUCB1 (b) Under PBM-UCB

Figure 2: Comparison of regret for attack strategies under different click feedback models.

Remark 4.1. We have conducted empirical experiments with some of the other popular OLTR algorithms as well, with results shown in Figure 3 in Appendix D. We observe that `CascadeOFA` successfully enforces the target choices for CascadeKL-UCB [6] and TS-Cascade [22] as well.

5 Conclusions

In this paper, we presented the first observation-free attack on OLTR algorithms across different click feedback models. We design two specific attacks: `CascadeOFA` for CascadeUCB1 and `PBMOFA` for PBM-UCB. Our analysis demonstrates that both attacks, can successfully promote their target items for $T - o(T)$ rounds with high probability, while requiring only $O(\log T)$ reward manipulations. We have supported our analysis through experiments on the real-world MovieLens dataset. In future work, we aim to develop attacks on OLTR algorithms under general stochastic click models and explore novel OLTR algorithms that would be inherently robust against such manipulations.

References

[1] A. Grotov and M. de Rijke, "Online learning to rank for information retrieval: Sigir 2016 tutorial," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1215–1218. [Online]. Available: https://doi.org/10.1145/2911451.2914798

[2] T.-Y. Liu *et al.*, "Learning to rank for information retrieval," *Foundations and Trends® in Information Retrieval*, vol. 3, no. 3, pp. 225–331, 2009.

[3] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey, "An experimental comparison of click position-bias models," in *Proceedings of the 2008 International Conference on Web Search and Data Mining*, ser. WSDM '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 87–94. [Online]. Available: https://doi.org/10.1145/1341531.1341545

[4] A. Chuklin, I. Markov, and M. de Rijke, *Click Models for Web Search*, ser. Synthesis Lectures on Information Concepts, Retrieval, and Services. Cham, Switzerland: Springer, 2015, vol. 43. [Online]. Available: https://link.springer.com/book/10.1007/978-3-031-02294-4

[5] T. Lattimore and C. Szepesvári, *Bandit algorithms*. Cambridge University Press, 2020.

[6] B. Kveton, Z. Wen, A. Ashkan, and C. Szepesvari, "Cascading bandits: Learning to rank in the cascade model," in *ICML*, 2015, pp. 767–776.

[7] B. Kveton, Z. Wen, A. Ashkan, and C. Szepesvári, "Combinatorial cascading bandits," in *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'15. Cambridge, MA, USA: MIT Press, 2015, p. 1450–1458.

[8] S. Zong, H. Ni, K. Sung, N. R. Ke, Z. Wen, and B. Kveton, "Cascading bandits for large-scale recommendation problems," *ArXiv*, vol. abs/1603.05359, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:2545548

[9] P. Lagrée, C. Vernade, and O. Cappé, "Multiple-play bandits in the position-based model," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 1605–1613.

[10] M. Zoghi, T. Tunys, M. Ghavamzadeh, B. Kveton, C. Szepesvari, and Z. Wen, "Online learning to rank in stochastic click models," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17. JMLR.org, 2017, p. 4199–4208.

[11] T. Lattimore, B. Kveton, S. Li, and C. Szepesvari, "Toprank: A practical algorithm for online stochastic ranking," *ArXiv*, vol. abs/1806.02248, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:46946360

[12] K.-S. Jun, L. Li, Y. Ma, and X. Zhu, "Adversarial attacks on stochastic bandits," in *Neural Information Processing Systems*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:53104043

[13] Y. Xu, B. Kumar, and J. D. Abernethy, "Observation-free attacks on stochastic bandits," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 22 550–22 561. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/be315e7f05e9f13629031915fe87ad44-Paper.pdf

[14] F. Liu and N. B. Shroff, "Data poisoning attacks on stochastic bandits," *ArXiv*, vol. abs/1905.06494, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:155100228

[15] Z. Wang, H. Wang, and H. Wang, "Stealthy adversarial attacks on stochastic multi-armed bandits," in *AAAI Conference on Artificial Intelligence*, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:267770319

[16] A. Rangi, L. Tran-Thanh, H. Xu, and M. Franceschetti, "Saving stochastic bandits from poisoning attacks via limited data verification," in *AAAI Conference on Artificial Intelligence*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:231925173

[17] J. Zuo, Z. Zhang, Z. Wang, S. Li, M. Hajiesmaili, and A. Wierman, "Adversarial attacks on online learning to rank with click feedback," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS '23. Red Hook, NY, USA: Curran Associates Inc., 2023.

[18] Z. Wang, R. Balasubramanian, H. Yuan, chenyu song, M. Wang, and H. Wang, "Adversarial attacks on online learning to rank with stochastic click models," *Transactions on Machine Learning Research*, 2024. [Online]. Available: https://openreview.net/forum?id=BKwGowR0Bt

[19] M. Hardt, E. Mazumdar, C. Mendler-Dünner, and T. Zrnic, "Algorithmic collective action in machine learning," in *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML'23. JMLR.org, 2023.

[20] J. Baumann and C. Mendler-Dünner, "Algorithmic collective action in recommender systems: promoting songs by reordering playlists," in *Proceedings of the 38th International Conference on Neural Information Processing Systems*, ser. NIPS '24. Red Hook, NY, USA: Curran Associates Inc., 2025.

[21] F. M. Harper, J. A. Konstan, and J. A., "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, pp. 19:1–19:19, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:16619709

[22] Z. Zhong, W. C. Chueng, and V. Y. F. Tan, "Thompson sampling algorithms for cascading bandits," *J. Mach. Learn. Res.*, vol. 22, no. 1, Jan. 2021.

[23] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, no. 2–3, p. 235–256, May 2002. [Online]. Available: https://doi.org/10.1023/A:1013689704352

A Algorithms for OLTR

In this section, we discuss two prominent OLTR algorithms: CascadeUCB1 (Algorithm 3) for the Cascade model and PBM-UCB (Algorithm 4) for PBM.

A.1 Definitions

Before presenting the algorithms in detail, we define the following terms:

- *Number of Recommendations*: $\mathcal{M}_a(t) = \sum_{\tau=1}^{t} \mathbb{1}\{a \in \mathcal{L}_\tau\}$ represents the number of times an OLTR algorithm has recommended an item a up to round t.
- *Number of Examinations*: $\mathcal{N}_a(t) = \sum_{\tau=1}^{t} \sum_{i=1}^{K} \mathbb{1}\{a_{i,\tau} = a\} \cdot \mathrm{X}_{i,\tau}$ denotes the number of times item a has been examined by users up to time t.
- *Number of Clicks*: $\mathcal{S}_a(t) = \sum_{\tau=1}^{t} \mathrm{Z}_{a,\tau}$ denotes the number of times item a has been clicked by users up to time t.
- *Empirical Mean*: The empirical mean reward for an item is given by $\hat{w}_{a,t} = \sum_{\tau=1}^{t} \mathrm{Z}_{a,\tau} / \mathcal{N}_a(t)$,and is defined as the average number of clicks per examination.
- *UCB Index*: The term $U_a(t)$ represents the upper confidence bound of the attractiveness of item a, as maintained by the OLTR algorithm.

Algorithm 3 CascadeUCB1

Input: item set $\mathcal{A}$, number of recommended items K, horizon T, exploration parameter α
Initialize: $t \leftarrow 1$, $\hat{w}_{a,t} \leftarrow 0$, $\mathcal{N}_a(t) \leftarrow 1$, for each $a \in \mathcal{A}$
while $t \leqslant T$ **do**
  Compute $U_a(t)$ according to (1)
  Recommend $\mathcal{L}_t = (a_{1,t}, \ldots, a_{K,t})$ to the user, where $a_{1,t} \ldots, a_{K,t} \in \mathcal{A}$ are the K items with largest $U_a(t-1)$
  Observe the user feedback - X_t and $\mathcal{C}_t$ (equivalently Z_t)
  for $a \in \mathcal{L}_t$ **do**
    if a is examined **then**
      $\mathcal{N}_a(t) \leftarrow \mathcal{N}_a(t-1) + 1$
      $\hat{w}_{a,t} \leftarrow \{\hat{w}_{a,t-1}\mathcal{N}_a(t-1) + \mathrm{Z}_{a,t}\}/\mathcal{N}_a(t)$
    else
      $\mathcal{N}_a(t) \leftarrow \mathcal{N}_a(t-1)$; $\hat{w}_{a,t} \leftarrow \hat{w}_{a,t-1}$
    end if
  end for
  $t \leftarrow t + 1$
end while

A.2 UCB-based OLTR Algorithms

In each round, both CascadeUCB1 and PBM-UCB compute the UCB index for each item based on the corresponding number of clicks, recommendations, and examinations, then select the K items with the highest UCB indices and recommend them as $\mathcal{L}_t$, ranking them in descending order of their UCB values. The computation of UCB differs significantly between the two algorithms; the computation in CascadeUCB1 is similar to that of the classical UCB1 algorithm [23], with a parameter $\alpha > 1$,

$$U_a(t) = \hat{w}_{a,t} + \sqrt{\frac{\alpha \log t}{\mathcal{N}_a(t)}}, \tag{1}$$

whereas PBM-UCB, where the learning agent cannot observe the examination feedback X_t, uses

$$U_a(t) = \frac{\mathcal{S}_a(t)}{\tilde{\mathcal{N}}_a(t)} + \sqrt{\frac{\alpha \mathcal{M}_a(t) \log t}{\tilde{\mathcal{N}}_a^2(t)}}, \tag{2}$$

where $\tilde{\mathcal{N}}_a(t)$ is an unbiased estimator for the number of examinations, given by $\tilde{\mathcal{N}}_a(t) = \sum_{\tau=1}^{T} \sum_{i=1}^{K} p_i \mathbb{1}\{a = a_{i,\tau}\}$.

The CascadeUCB1 algorithm was originally introduced in [6], which provided an $O(\log T)$ upper bound on its expected cumulative regret for a given horizon T. The corresponding PBM-UCB algorithm for position-based feedback was proposed by [9] with a similar $O(\log T)$ regret bound.

Algorithm 4 PBM-UCB

Input: item set $\mathcal{A}$, number of recommended items K, position bias $\mathcal{P}$, horizon T, exploration parameter α
Initialize: $t \leftarrow 1$, $\mathcal{S}_a(t) \leftarrow 0$, $\mathcal{M}_a(t) \leftarrow 1$, for each $a \in \mathcal{A}$
while $t \leqslant T$ **do**
  Compute $U_a(t)$ according to (2)
  Recommend $\mathcal{L}_t = (a_{1,t}, \ldots, a_{K,t})$ to the user, where $a_{1,t} \ldots, a_{K,t} \in \mathcal{A}$ are the K items with largest $U_a(t-1)$
  Observe the user feedback $\mathcal{C}_t$ and Z_t
  for $a \in \mathcal{L}_t$ **do**
    $\mathcal{M}_a(t) \leftarrow \mathcal{M}_a(t-1) + 1$
    $\mathcal{S}_a(t) \leftarrow \mathcal{S}_a(t-1) + \mathrm{Z}_{a,t}$
  end for
  $t \leftarrow t + 1$
end while

Remark A.1. The parameter α, present in both the OLTR algorithms (check (1) and (2)), is often referred to as the exploration parameter or the confidence parameter. The parameter is commonly used for the UCB-based online learning algorithms to define the confidence radii. Intuitively, a higher α ensures that the OLTR is more confident of its learned list being optimal, but it comes with a higher regret bound for the algorithms due to increased exploration of sub-optimal algorithms. Both the OLTR algorithms and their corresponding attack strategies are valid for any $\alpha > 1$.

B Key Proofs

This section provides the proofs for the theorems given in Section 3.

B.1 Proof of Theorem 3.1

Some of the consequences of the `CascadeOFA` attack on CascadeUCB1 are stated below in the following lemmas, which are used to prove Theorem 3.1. The proofs of these lemmas are discussed in the Appendix C.

Lemma B.1. *After the first phase of* `CascadeOFA` *with the value of* C_1 *given in Section 3.1.1,* $U_a(\mathrm{C}_1 + 1) \leqslant \mathrm{w_m} \ \forall\, a \in \mathcal{A}$.

From Lemma B.1, we conclude that, at the end of phase 1 of `CascadeOFA`, UCBs for all the items in $\mathcal{A}$ fall below $\mathrm{w_m}$.

Lemma B.2. *After the second phase of* `CascadeOFA` *with* C_1 *and* C_2 *given in Section 3.1.1,* $U_a(\mathrm{C}_1 + \mathrm{C}_2 + 1) > \mathrm{w_m}$ *for all* $a \in \tilde{\mathcal{L}}$ *and* $U_{a'}(\mathrm{C}_1 + \mathrm{C}_2 + 1) \leqslant \mathrm{w_m}$ *for all* $a' \notin \tilde{\mathcal{L}}$.

Thus, at the end of Phase 2 of `CascadeOFA`, the UCBs of all items in the target set $\tilde{\Gamma}$ exceed $\mathrm{w_m}$, while those for the remaining items in $\mathcal{A}$ stay below $\mathrm{w_m}$.

Lemma B.3. *Without any further manipulation in the third phase of* `CascadeOFA`*, an item* $a \in \tilde{\mathcal{L}}$ *maintains* $U_a(t) > \mathrm{w_m}$*, with probability*

$$\Pr\{U_a(t) > \mathrm{w_m} \ \forall \ \mathrm{C}_1 + \mathrm{C}_2 < t \leqslant T\} \geqslant 1 - 1/T, \tag{3}$$

Thus, with high probability, each item in $\tilde{\mathcal{L}}$ maintains its UCB above $\mathrm{w_m}$ throughout the third phase of `CascadeOFA`. This brings us to the Proof of Theorem 3.1 which uses the abovementioned lemmas.

Proof of Theorem 3.1. According to Lemma B.1 and Lemma B.2, the `CascadeOFA` strategy outlined in Algorithm 1 with the values of C_1 and C_2 given by Section 3.1.1 ensures that at the end of first

$$\mathrm{C} = \mathrm{C}_1 + \mathrm{C}_2$$

rounds, $U_a(\mathrm{C}+1) \geqslant \mathrm{w_m} \ \forall \ a \in \tilde{\mathcal{L}}$ and $U_{a'}(\mathrm{C}+1) \leqslant \mathrm{w_m} \ \forall \ a' \notin \tilde{\mathcal{L}}$. Additionally, Lemma B.3, guarantees that any $a \in \tilde{\mathcal{L}}$ maintains an UCB above $\mathrm{w_m}$ with a probability $\geqslant 1 - 1/T$.

Let Λ_i be the event that $U_{\tilde{\mathcal{L}}[i]}(t) > \mathrm{w_m}$ for all t in $\{C+1, \ldots, T\}$. Extending the analysis simultaneously to all the items in $\tilde{\mathcal{L}}$, for $t > \mathrm{C}$, we obtain

$$\Pr\{\Lambda_1 \cap \Lambda_2 \cdots \cap \Lambda_K\} \geqslant 1 - K/T.$$

With $U_{a'}(\mathrm{C}+1) \leqslant \mathrm{w_m}$ for all $a' \notin \tilde{\mathcal{L}}$ and $U_a(t) > \mathrm{w_m}$ for all $a \in \tilde{\mathcal{L}}$ in the third phase, none of the non-target items would receive any examinations after $t = \mathrm{C}$. Consequently, with high probability, $U_{a'}(t) = U_{a'}(\mathrm{C}+1) \leqslant \mathrm{w_m}$ for all $a' \notin \tilde{\mathcal{L}}$ and $\mathrm{C} < t \leqslant T$. This ensures that $\mathcal{L}_t$ is always a permutation of $\tilde{\mathcal{L}}$ during the final $T - \mathrm{C}$ rounds and all items in $\tilde{\Gamma}$ are consistently recommended by CascadeUCB1 in the third phase.

The probability of the *success event* σ, i.e., `CascadeOFA` successfully misleading CascadeUCB1 into recommending items from the target set $\tilde{\Gamma}$ as part of $\mathcal{L}_t$ for $T - o(T)$ rounds is given by $\Pr(\sigma) \geqslant 1 - K/T$.

According to Theorem 2 in [6], in the absence of any external manipulations, CascadeUCB1 incurs an expected regret of $O(\log T)$. In contrast, the cumulative T-step regret of CascadeUCB1 with `CascadeOFA` is lower-bounded by

$$\mathcal{R}(T) \geqslant \sum_{t=\mathrm{C}+1}^{T} \mathbb{E}\left[\mathrm{R}(\mathcal{L}_t, w)\right] \geqslant \Pr(\sigma) \cdot \sum_{t=\mathrm{C}+1}^{T} \min_{\Phi_{\mathrm{perm}}(\tilde{\mathcal{L}})} \mathrm{R}(\mathcal{L}, w)) \geqslant \left(1 - \frac{K}{T}\right) \cdot (T - \mathrm{C}) \cdot \mathrm{R}(\tilde{\mathcal{L}}, w).$$

$\mathcal{R}(T) = \Omega(T)$ if $\mathrm{R}(\tilde{\mathcal{L}}, w) > 0$, which holds true for any $\tilde{\mathcal{L}}$ containing at least one element not in $\mathcal{L}^*$. Thus, `CascadeOFA` is not only successful in recommending the target items for $T - o(T)$ rounds but also ensures a linear regret for CascadeUCB1. □

B.2 Proof of Theorem 3.2

Some of the major results for `PBMOFA` are stated below in the following lemmas, which are used to prove Theorem 3.1. The proofs of these lemmas are discussed in the Appendix C.

Lemma B.4. *After the first phase of* `PBMOFA` *with the value of* C_1 *given in Section 3.2.1, the probability* $\Pr\{\sigma^{(1)}\} = 1$*, where* $\sigma^{(1)} = \mathcal{E}_1 \cap \mathcal{E}_2 \cdots \cap \mathcal{E}_L$ *and* $\mathcal{E}_a$ *is the event that* $U_a(\mathrm{C}_1 + 1) \leqslant \mathrm{w_m}$.

At the end of the first phase, UCBs for all the items in $\mathcal{A}$ fall below $\mathrm{w_m}$ with a probability of one.

Lemma B.5. *After the second phase of* `PBMOFA` *with the value of* C_1 *and* C_2 *given in Section 3.2.1,* $\Pr\{\sigma^{(2)}\} \geqslant 1 - K/T$*, where* $\sigma^{(2)} = {\mathcal{E}'}_1 \cap {\mathcal{E}'}_2 \cdots \cap {\mathcal{E}'}_K$ *and* ${\mathcal{E}'}_i$ *is the event that* $U_{\tilde{\mathcal{L}}[i]}(\mathrm{C}_1 + \mathrm{C}_2 + 1) > \mathrm{w_m}$*. Additionally,* $U_{a'}(\mathrm{C}_1 + \mathrm{C}_2 + 1) \leqslant \mathrm{w_m}$ *for all* $a' \notin \tilde{\mathcal{L}}$.

At the end of Phase 2 of PBMOFA, the UCBs of all items in $\tilde{\Gamma}$ exceed $\mathrm{w_m}$, while those of the remaining items in $\mathcal{A}$ stay below $\mathrm{w_m}$ with a high probability.

Lemma B.6. *Given the success of the first two phases and in the absence of further manipulation in the third phase of* PBMOFA*, an item $a \in \tilde{\mathcal{L}}$ maintains UCB $U_a(t) > \mathrm{w_m}$ under PBM-UCB with a probability*

$$\Pr\{\sigma^{(3)} \mid \sigma^{(1)} \cap \sigma^{(2)}\} \geqslant 1 - K/T,$$

where $\sigma^{(3)} = {\mathcal{E}''}_1 \cap {\mathcal{E}''}_2 \cdots \cap {\mathcal{E}''}_K$ and ${\mathcal{E}''}_i$ is the event that $U_{\tilde{\mathcal{L}}[i]}(t) > \mathrm{w_m}$ for all t in $\{C+1, \ldots, T\}$.

If the first two phases of PBMOFA were successful, then, with high probability, each item in $\mathcal{A}$ maintains its UCB above $\mathrm{w_m}$ throughout the third phase of PBMOFA.

This brings us to the Proof of Theorem 3.2 which uses the abovementioned lemmas.

Proof of Theorem 3.2. The Lemmas B.4 and B.5 provide the guarantees of success for the first two phases of PBMOFA. In the final phase of the attack, Lemma B.6 asserts that if the earlier phases were successful, UCB $U_a(t) > \mathrm{w_m}$ for all $a \in \tilde{\mathcal{L}}$, with probability at least $1 - K/T$ while $\mathrm{C_1} + \mathrm{C_2} < t \leqslant T$.

Using a proof sketch similar to that of Theorem 3.1, the probability of PBMOFA successfully misleading the PBM-UCB into recommending items from $\tilde{\Gamma}$ as part of $\mathcal{L}_t$ for $T - o(T)$ rounds is given by

$$\Pr(\sigma) = 1 - \Pr(\text{failure}) \geqslant 1 - \Pr(\overline{\sigma^{(1)}}) - \Pr(\overline{\sigma^{(2)}}) - \Pr(\overline{\sigma^{(3)}} | \sigma^{(1)} \cap \sigma^{(2)}) \geqslant 1 - \frac{2K}{T}.$$

Thus, PBMOFA efficiently misleads PBM-UCB into recommending items from the target set $\tilde{\Gamma}$ for $T - o(T)$ rounds with high probability with a manipulation of the order $O(\log T)$.

Theorem 9 in [9] states that in the absence of any attack, the PBM-UCB algorithm incurs an $O(\log T)$ regret. We define $\tilde{\mathcal{L}}' = \underset{\Phi_{\mathrm{perm}}(\tilde{\mathcal{L}})}{\arg\min}(\mathrm{R}(\mathcal{L}, w))$.

The cumulative T-step regret for PBM-UCB under PBMOFA is lower-bounded by

$$\mathcal{R}(T) \geqslant \sum_{t=\mathrm{C}+1}^{T} \mathbb{E}\left[\mathrm{R}(\mathcal{L}_t, w)\right] \geqslant \Pr(\sigma) \cdot \sum_{t=\mathrm{C}+1}^{T} \min_{\Phi_{\mathrm{perm}}(\tilde{\mathcal{L}})} \mathrm{R}(\mathcal{L}, w)) \geqslant \left(1 - \frac{K}{T}\right) \cdot (T - \mathrm{C}) \cdot \mathrm{R}(\tilde{\mathcal{L}}', w).$$

Therefore, PBMOFA, along with successfully recommending the target items for $T - o(T)$ rounds, also forces an $\Omega(T)$ linear regret on the PBM-UCB algorithm. □

C Additional Proofs

C.1 Proof of Lemma B.1

We use the following lemma to prove Lemma B.1.

Lemma C.1. *Under CascadeUCB1, given a set Γ of items with equal empirical means and UCBs at the start of a round t. Any item $a \in \Gamma$ that is examined but not clicked in round t cannot be examined again until the remaining items in Γ get an examination. The items in Γ are recommended and examined in a round-robin fashion.*

Proof of Lemma C.1. Let there be two items $a, a' \in \Gamma$ with $\hat{w}_{a,t} = \hat{w}_{a',t}$ and $U_a(t) = U_a(t)$, such that a is examined but not clicked at round t, while a' is not examined in the same round. Let a get another examination in round $t_1 > t$, while a' has no examinations in rounds $\{t, \ldots, t_1\}$. Under CascadeUCB1, if item a gets an examination before a' in round t_1, it would imply that $U_a(t_1) \geqslant U_{a'}(t_1)$, i.e. $U_a(t_1) \geqslant U_a(t)$, which is contradictory to the fact that $\hat{w}_{a,t_1} < \hat{w}_{a,t}$ and $\mathcal{N}_a(t) > \mathcal{N}_a(t_1)$, as item a had been examined but not clicked at round t. Thus, an item $a \in \Gamma$ that is examined but not clicked in round t cannot be examined again until all the remaining items $a' \in \Gamma / \{a\}$ get an examination. This implies that the items in Γ will be recommended and examined in a round-robin manner under CascadeUCB1. □

This brings us to the Proof of Lemma B.1 which uses the abovementioned lemmas.

Proof of Lemma B.1. While $t \leqslant \mathrm{C}_1$, the rewards for all the items are set to 0. Therefore, in each round, the items are examined and ignored (not clicked) in a round-robin pattern in batches of K items according to Lemma C.1. After the completion of the first phase, the number of pulls for each item a is given by $\mathcal{N}_a\left(\mathrm{C}_1+1\right)=\frac{K\mathrm{C}_1}{L}$, and therefore with the value of C_1 given in Section 3.1,

$$U_a\left(\mathrm{C}_1+1\right) \leqslant \sqrt{\frac{L\alpha \log T}{K\mathrm{C}_1}} \leqslant \mathrm{w}_\mathrm{m} \; \forall \; a \in \mathcal{A}.$$

□

C.2 Proof of Lemma B.2

Proof of Lemma B.2. The second phase is split into K sub-phases with the i^{th} sub-phase lasting from round $\mathrm{C}_1+\frac{(i-1)\mathrm{C}_2}{K}+1$ to round $\mathrm{C}_1+\frac{i\mathrm{C}_2}{K}$ where $1 \leqslant i \leqslant K$. During the i^{th} sub-phase, the reward for $\tilde{\mathcal{L}}[i]$ is fixed as 1, while the rewards for the rest of the items are set to 0.

Assume that at the start of the i^{th} sub-phase, $\tilde{\mathcal{L}}[i] \notin \mathcal{L}_t$. At this point, the top $i-1$ positions of $\mathcal{L}_t$ are occupied by a permutation of the first $i-1$ members of $\tilde{\mathcal{L}}$. Until $\tilde{\mathcal{L}}[i] \in \mathcal{L}_t$, none of the items in $\mathcal{L}_t$ will be clicked. As a result, the lower $K-i+1$ positions in $\mathcal{L}_t$ are filled in a round-robin manner (as seen in Lemma C.1) from the remaining $L-i+1$ items in $\mathcal{A}$ with zero empirical mean. Once $\tilde{\mathcal{L}}[i] \in \mathcal{L}_t$, the item $\tilde{\mathcal{L}}[i]$ is clicked for the rest of the sub-phase. The number of rounds required to search for $\tilde{\mathcal{L}}[i]$ at the start of the i^{th} sub-phase is upper-bounded by

$$\left\lceil \frac{L-i+1}{K-i+1} \right\rceil \leqslant L-K+1, \quad 1 \leqslant i \leqslant K.$$

With each sub-phase lasting for an equal number of rounds, the empirical mean for all the items in $\tilde{\mathcal{L}}$ (and thus $\tilde{\Gamma}$) being greater than w_m at the end of round two is ensured by

$$\frac{\mathrm{C}_2/K-(L-K+1)}{K\mathrm{C}_1/L+\mathrm{C}_2} \geqslant \mathrm{w}_\mathrm{m}.$$

Thus, with C_2 equal to the value given in Section 3.1.1, $\hat{w}_{a,\mathrm{C}+1} \geqslant \mathrm{w}_\mathrm{m}$ and $U_a(\mathrm{C}+1) \geqslant \mathrm{w}_\mathrm{m}$ for all $a \in \tilde{\mathcal{L}}$. Additionally, the items not in $\tilde{\mathcal{L}}$ continue to be ignored (not clicked) during each of the recommendations, thus pulling down their UCBs further below w_m by the end of the second phase, i.e. $U_{a'}(\mathrm{C}+1) \leqslant \mathrm{w}_\mathrm{m}$ for all $a' \notin \tilde{\mathcal{L}}$. These two events occur simultaneously in the second phase of `CascadeOFA` proving Lemma B.2. Note that C_2 is a positive quantity as $\mathrm{w}_\mathrm{m} < 1/K$. □

Note. The decision to divide the second phase of `CascadeOFA` into K equal-length sub-phases is a design choice and alternative bounds for C_2 can be derived by using sub-phases of varying lengths. A possible suggestion can be to give $\left\lceil \frac{(K-i)\mathrm{C}_2}{K(K+1)} \right\rceil$ rounds to the i^{th} sub-phase. Intuitively, giving more rounds to the earlier sub-phases is beneficial, as their corresponding target items get observed and ignored for all the later sub-phases, thus pulling down their UCBs.

C.3 Proof of Lemma B.3

Proof of Lemma B.3. For the last $T-\mathrm{C}$ rounds, all the items in $\tilde{\mathcal{L}}$ should be recommended to the user with a high probability. At the beginning of the third phase, $U_a\left(\mathrm{C}+1\right) > \mathrm{w}_\mathrm{m}$ for all $a \in \tilde{\mathcal{L}}$.

Let an item $a \in \tilde{\mathcal{L}}$ exist, which receives m examinations in the initial two phases and further n examinations rest of the rounds. By Hoeffding's inequality, the number of clicks for the item a in the third phase until some time $t \leqslant T$ will be greater than $nw_a-\sqrt{n\log t}$ with probability at least $1-1/T$ (Check Section A.2 in [13]), and thus, the UCB for this item at time $\mathrm{C} < t \leqslant T$ is given by

$$U_a\left(t\right)=\hat{w}_a\left(t\right)+\sqrt{\frac{\alpha\log t}{n+m}} \geqslant \frac{m\mathrm{w}_\mathrm{m}+nw_a}{n+m}+\sqrt{\frac{\alpha\log t}{n+m}}-\frac{\sqrt{n\log t}}{n+m} > \mathrm{w}_\mathrm{m},$$

which follows from the fact that $w_a > \mathrm{w}_\mathrm{m}$ and $\alpha > 1$.Therefore, the UCB of any item in $\tilde{\mathcal{L}}$ will be greater than w_m for any round while $\mathrm{C} < t \leqslant T$ with the probability given in (3). □

C.4 Proof of Lemma B.4

Proof of Lemma B.4. We use the following lemma to prove Lemma B.4.

Lemma C.2. *Given two items $a_i, a_j \in \mathcal{A}$, $\mathcal{M}_{a_j}(t) \leqslant \lambda_p^2 \mathcal{M}_{a_i}(t) + 1$ for any $t \leqslant \mathrm{C}_1$ during the first phase of* PBMOFA.

Proof of Lemma C.2. We prove the lemma using inductive reasoning. Consider any two arbitrarily chosen items $a_i, a_j \in \mathcal{A}$.

Base Case ($t = 1$): At the time of initialization ($t \leftarrow 1$), we have $\mathcal{M}_{a_i}(t) = \mathcal{M}_{a_j}(t) = 1$. Thus, the lemma statement holds for $t = 1$ since $\mathcal{M}_{a_j}(t) \leqslant \lambda_p^2 \mathcal{M}_{a_i}(t) + 1$.

Inductive Step ($t = t_o + 1$): Assume that at the beginning of some round $t_o < \mathrm{C}_1$, the inductive hypothesis holds:

$$\mathcal{M}_{a_j}(t_o) \leqslant \lambda_p^2 \mathcal{M}_{a_i}(t_o) + 1.$$

We now show that the lemma statement holds for $t = t_o + 1$.

At $t = t_o$, one of the following mutually exclusive and exhaustive events must occur-

1. $a_i \in \mathcal{L}_{t_o}$ and $a_j \in \mathcal{L}_{t_o}$: In this case, at the start of round $t_o + 1$,

   $$\mathcal{M}_{a_i}(t_o + 1) = \mathcal{M}_{a_i}(t_o) + 1, \quad \mathcal{M}_{a_j}(t_o + 1) = \mathcal{M}_{a_j}(t_o) + 1.$$

   Thus, $\mathcal{M}_{a_j}(t_o + 1) \leqslant \lambda_p^2 \mathcal{M}_{a_i}(t_o) + 2 \leqslant \lambda_p^2 \mathcal{M}_{a_i}(t_o + 1) + 1$. The lemma holds in this case.

2. $a_i \in \mathcal{L}_{t_o}$ and $a_j \notin \mathcal{L}_{t_o}$: In this case,

   $$\mathcal{M}_{a_i}(t_o + 1) = \mathcal{M}_{a_i}(t_o) + 1, \quad \mathcal{M}_{a_j}(t_o + 1) = \mathcal{M}_{a_j}(t_o).$$

   Therefore, $\mathcal{M}_{a_j}(t_o + 1) \leqslant \lambda_p^2 \mathcal{M}_{a_i}(t_o) + 1 \leqslant \lambda_p^2 \mathcal{M}_a(t_o + 1) + 1$.

   The lemma holds in this case as well.

3. $a_i \notin \mathcal{L}_{t_o}$ and $a_j \notin \mathcal{L}_{t_o}$: In this scenario,

   $$\mathcal{M}_{a_i}(t_o + 1) = \mathcal{M}_{a_i}(t_o), \quad \mathcal{M}_{a_j}(t_o + 1) = \mathcal{M}_{a_j}(t_o).$$

   Hence, the lemma statement naturally holds since the quantities remain unchanged.

4. $a_i \notin \mathcal{L}_{t_o}$ and $a_j \in \mathcal{L}_{t_o}$: This occurs if and only if $U_a(t_o) \leqslant U_{a_j}(t_o)$. For $t_o < \mathrm{C}_1$ in PBMOFA, we know:

   $$\sqrt{\frac{\alpha \log t}{p_1^2 \mathcal{M}_{a_i}(t_o)}} \leqslant U_{a_i}(t_o) \leqslant \sqrt{\frac{\alpha \log t}{p_K^2 \mathcal{M}_a(t_o)}},$$

   $$\sqrt{\frac{\alpha \log t}{p_K^2 \mathcal{M}_{a_j}(t_o)}} \leqslant U_{a_j}(t_o) \leqslant \sqrt{\frac{\alpha \log t}{p_K^2 \mathcal{M}_{a_j}(t_o)}}.$$

   This is a consequence of the fact that

   $$p_K \mathcal{M}_a(t) \leqslant \tilde{\mathcal{N}}_a(t) \leqslant p_1 \mathcal{M}_a(t)$$

   for all $a \in \mathcal{A}$ (follows directly from the definition of $\tilde{\mathcal{N}}_a(t)$). Therefore, $U_{a_i}(t_o) \leqslant U_{a_j}(t_o)$ is possible iff

   $$\mathcal{M}_{a_j}(t_o) \leqslant \lambda_p^2 \mathcal{M}_{a_i}(t_o).$$

   Thus, $\mathcal{M}_{a_j}(t_o + 1) = \mathcal{M}_{a_j}(t_o) + 1 \leqslant \lambda_p^2 \mathcal{M}_a(t_o + 1) + 1$.The lemma holds in this case as well.

The lemma statement is always true for $t = t_o + 1$. Therefore, by induction, the lemma is true for all rounds $t \leqslant \mathrm{C}_1$. □

During the first phase of PBMOFA, the attack ensures that at $t = \mathrm{C}_1 + 1$, $U_a(t) \leqslant \sqrt{\frac{\alpha \log T}{p_K^2 \mathcal{M}_a(t)}} \leqslant \mathrm{w_m}$ for all $a \in \mathcal{A}$ and $t \leqslant \mathrm{C}_1$. For this to happen, we need

$$\mathcal{M}_a(\mathrm{C}_1 + 1) \geqslant \frac{\alpha \log T}{\mathrm{w_m^2}} \geqslant \frac{\alpha \log t}{\mathrm{w_m^2}} \; \forall \; a \in \mathcal{A}.$$

Let m denote the minimum number of recommendations received by an item in $\mathcal{A}$ during the first phase of PBMOFA. The corresponding maximum number of recommendations will be less than $\lambda_p^2 m + 1$, based on the Lemma C.2.

The total number of recommendations during the first phase is equal to $K\mathrm{C}_1$, which implies

$$\sum_{a=1}^{L} \mathcal{M}_a(\mathrm{C}_1) = K\mathrm{C}_1. \tag{4}$$

Since each item receives at least m and at most $\lambda_p^2 m + 1$ recommendations, we can write

$$L \cdot m \leqslant \sum_{a=1}^{L} \mathcal{M}_a(\mathrm{C}_1) \leqslant L \cdot (\lambda_p^2 m + 1).$$

Substituting $\sum_{a=1}^{L} \mathcal{M}_a(\mathrm{C}_1) = K\mathrm{C}_1$, we obtain

$$L \cdot (\lambda_p^2 m + 1) \geqslant K\mathrm{C}_1. \tag{5}$$

On substituting the value of C_1 from Section 3.2.1, we get $m \geqslant \alpha \log T / \mathrm{w_m^2}$ and therefore $U_a(\mathrm{C}_1 + 1) \leqslant \mathrm{w_m}$ for all $a \in \mathcal{A}$, thus proving Lemma B.4 □

C.5 Proof of Lemma B.5

Proof of Lemma B.5. Assume an item $a \in \tilde{\mathcal{L}}$ receives m_a recommendations in the first phase and n_a recommendations in the second phase of PBMOFA. Then at $t = \mathrm{C}_1 + \mathrm{C}_2 + 1$,

$$U_a(t) > \frac{\mathcal{S}_a(t)}{\tilde{\mathcal{N}}_a(t)} \geqslant \frac{\mathcal{S}_a(t)}{p_1 \mathcal{M}_a(t)} = \frac{\mathcal{S}_a(t)}{p_1(n_a + m_a)}.$$

During this phase, each examination for an item in $\tilde{\mathcal{L}}$ corresponds to a click.

With a probability $\geqslant 1 - 1/T$, $\mathcal{S}_a(t) \geqslant p_K n_a - \sqrt{n_a \log T}$ for all $t \in \{\mathrm{C}_1 + 1, \ldots, \mathrm{C}_1 + \mathrm{C}_2\}$, this follows from an argument similar to the one given in the Proof of B.3 (Appendix C.3). During the second phase of PBMOFA, the attack ensures that

$$\frac{p_K n_a - \sqrt{n_a \log T}}{p_1(n_a + m_a)} \geqslant \mathrm{w_m} \; \forall \; a \in \mathcal{A}. \tag{6}$$

Carrying forward the arguments given in the proof of Lemma B.4, m_a is upper-bounded by

$$m_a \leqslant K\mathrm{C}_1 - \frac{(L-1)\alpha \log T}{\mathrm{w_m^2} p_K^2} \; \forall \; a \in \mathcal{A}.$$

On setting m_a to the maximum possible value and solving (6), we obtain $n_a \geqslant \gamma \log T$. Looking back at (4) and (5), we find that they hold for any $t \leqslant \mathrm{C}_1$ as well. During phase 1, the minimum number of recommendations for any item between time t and $t + \Delta_t$ is lower-bounded by

$$m_{\Delta_t} = \frac{1}{\lambda_p^2} \left\{ \frac{K\Delta_t}{L} - 1 \right\}, \; t + \Delta_t \leqslant \mathrm{C}_1.$$

At the beginning of the second phase, the items in $\tilde{\mathcal{L}}$ are indistinguishable from the rest and receive at least an m_{Δ_t}/Δ_t fraction of recommendations between $t = \mathrm{C}_1$ and $t = \mathrm{C}_1 + \Delta_t$ for $\Delta_t << \mathrm{C}_2$. As the second phase progresses, the empirical means of the items in $\tilde{\mathcal{L}}$ increase, making them more distinguishable from the rest of the items in $\mathcal{A}$. Consequently, the instantaneous fraction of

recommendations for these items increases from m_{Δ_t}/Δ_t, converging asymptotically to 1. Therefore, $n_a \geqslant m_{\mathrm{C}_2}$; on solving

$$m_{\mathrm{C}_2} = \frac{1}{\lambda_p^2}\left\{\frac{K\mathrm{C}_2}{L} - 1\right\} = \gamma \log T,$$

we obtain the value of C_2 given in Section 3.2.1. Thus, on setting $\mathrm{C}_2 = \left\lceil \frac{L\left(\lambda_p^2 \gamma \log T + 1\right)}{K} \right\rceil$, with a probability $\geqslant 1 - 1/T$, $U_a(t) > \mathcal{S}_a(t)/\tilde{\mathcal{N}}_a(t) \geqslant \mathrm{w_m}$ for a given item a in $\mathcal{A}$ and $t = \mathrm{C}_1 + \mathrm{C}_2 + 1$. Extending this analysis to all the items in $\mathcal{A}$, we obtain $\Pr\{\sigma^{(2)}\} \geqslant 1 - K/T$, where $\sigma^{(2)} = \mathcal{E'}_1 \cap \mathcal{E'}_2 \cdots \cap \mathcal{E'}_K$ and $\mathcal{E'}_i$ is the event that $U_{\tilde{\mathcal{L}}[i]}\left(\mathrm{C}_1 + \mathrm{C}_2 + 1\right) > \mathrm{w_m}$. Additionally, $U_{a'}(\mathrm{C}_1 + \mathrm{C}_2 + 1) \leqslant \mathrm{w_m}$ for all $a' \notin \tilde{\mathcal{L}}$ as none of the other items outside of $\tilde{\mathcal{L}}$ receive any clicks during this phase; hence proving Lemma B.5. □

C.6 Proof of Lemma B.6

Proof of Lemma B.6. This proof proceeds similar to that of Lemma B.3. Let an item $a \in \tilde{\mathcal{L}}$ exist, which receives m recommendations in the initial two phases and further n recommendations during the third phase. For this item in $\mathcal{A}$, we define the following quantities:

$$p_{avg1}^{(a)} = \frac{\sum_{\tau=1}^{\mathrm{C}}\sum_{i=1}^{K} p_i \mathbb{1}\{a = a_{i,\tau}\}}{\sum_{\tau=1}^{\mathrm{C}} \mathbb{1}\{a \in \mathcal{L}_\tau\}},\ p_{avg2}^{(a)} = \frac{\sum_{\tau=\mathrm{C}+1}^{t}\sum_{i=1}^{K} p_i \mathbb{1}\{a = a_{i,\tau}\}}{\sum_{\tau=\mathrm{C}+1}^{t} \mathbb{1}\{a \in \mathcal{L}_\tau\}}.$$

By Hoeffding's inequality, the number of clicks for the item a at any time $t \leqslant T$ in the third phase will be greater than $nw_a p_{avg2}^{(a)} - \sqrt{n \log t}$ with probability at least $1 - 1/T$, and thus, assuming the success of phases 1 and 2, the UCB for this item at time $\mathrm{C} < t \leqslant T$ is given by

$$U_a\left(t\right) \geqslant \frac{\mathcal{S}_a(t)}{\tilde{\mathcal{N}}_a(t)} + \sqrt{\frac{\alpha \mathcal{M}_a(t) \log t}{\tilde{\mathcal{N}}_a^2(t)}} \geqslant \frac{m\mathrm{w_m} p_{avg1}^{(a)} + nw_a p_{avg2}^{(a)}}{np_{avg1}^{(a)} + mp_{avg2}^{(a)}} + \frac{\sqrt{\alpha(n+m)\log t} - \sqrt{n \log t}}{np_{avg1}^{(a)} + mp_{avg2}^{(a)}} > \mathrm{w_m},$$

which follows from the fact that $w_a > \mathrm{w_m}$ and $\alpha > 1$.Therefore, the UCB of any item in $\tilde{\mathcal{L}}$ will be greater than $\mathrm{w_m}$ for any round while $\mathrm{C} < t \leqslant T$ with the probability $\geqslant 1 - 1/T$. Extending this analysis to all the target items, we can say that all items in $\tilde{\mathcal{L}}$ maintain UCBs $U_a(t) > \mathrm{w_m}$ under PBM-UCB with a probability

$$\Pr\{\sigma^{(3)} \mid \sigma^{(1)} \cap \sigma^{(2)}\} \geqslant 1 - K/T,$$

where $\sigma^{(3)} = \mathcal{E''}_1 \cap \mathcal{E''}_2 \cdots \cap \mathcal{E''}_K$ and $\mathcal{E''}_i$ is the event that $U_{\tilde{\mathcal{L}}[i]}\left(t\right) > \mathrm{w_m}$ for all $C < t \leqslant T$. □

D Supplementary Empirical Analysis

This section offers additional details on the empirical analysis discussed in Section 4.

We conduct experiments on the real-world MovieLens dataset, processed in the same way as done in [8]. In all experiments, we set $T = 5 \times 10^5$, $K = 3$, $L = 10$, $\alpha = 1.5$, and average results over 50 runs. For PBM, we use $\mathcal{P} = (0.95, 0.90, 0.85)$. $\mathcal{A}$ is created by arbitrarily choosing 10 movies from the dataset. All the results in Section 4 are provided for a given $\mathcal{A}$, with the following attraction probabilities for the constituent items:

$$w = (0.336, 0.204, 0.163, 0.125, 0.112, 0.105, 0.099, 0.090, 0.086, 0.082).$$

The rewards are sampled in an I.I.D. manner from the dataset. With $\tilde{\mathcal{L}} = (4, 7, 10)$ for `CascadeOFA` and $\tilde{\mathcal{L}} = (8, 9, 10)$ for `PBMOFA`, we obtain $w_{\min} = 0.082$ for both the target sets, and thus, for both manipulation strategies, we use $w_m = 0.08$. For the given $\mathcal{A}$ and the abovementioned parameters, the phase durations (manipulations) of the attack strategies are as follows-

1. `CascadeOFA`: $\mathrm{C}_1 = 10260$ and $\mathrm{C}_2 = 1005$.
2. `PBMOFA`: $\mathrm{C}_1 = 11507$ and $\mathrm{C}_2 = 1304$.

Table 1: Comparison of costs and regrets for different attack strategies with horizon $T = 5 \times 10^5$.

(a) On CascadeUCB1

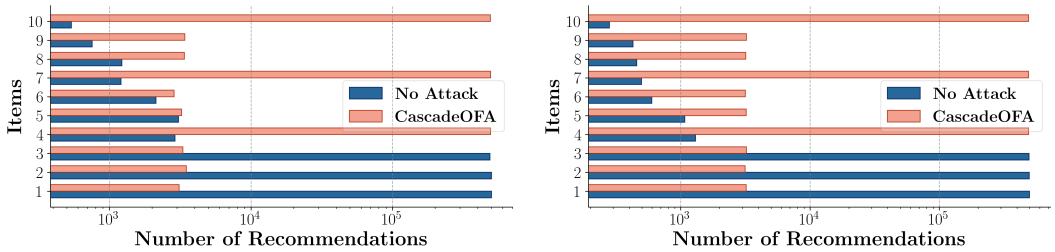
Attack	Manipulations	$\mathcal{R}(T)$
No Attack	-	2.233×10^3
CascadeATQ	11265	1.509×10^4
CascadeAlphaAtk	2927	1.457×10^5
`CascadeOFA`	11265	1.473×10^5

(b) On PBM-UCB

Attack	Manipulations	$\mathcal{R}(T)$
No Attack	-	2.507×10^3
PBMATQ	12811	2.153×10^4
PBMAlphaAtk	3432	1.989×10^5
`PBMOFA`	12811	1.922×10^5

The required reward manipulation for both the algorithms is well under 3% of the total number of rounds, showing that even a small group of adversarial users may greatly affect the outcome of the learning algorithm. Note that there is a significant gap between the attraction probability of the items in $\mathcal{L}^*$ and the rest, with $\approx 25\%$ difference in that of items 3 and 4. Therefore, the stated amount of reward manipulation does not significantly affect the overall distribution.

In Section 4, we presented the results for both the attack strategies and showcased how they are able to successfully mislead CascadeUCB1 and PBM-UCB to recommend their respective target items for a large majority of the rounds in the third phase.



(a) CascadeKL-UCB (b) TS-Cascade

Figure 3: The number of recommendations for the target items with and without `CascadeOFA`.

We benchmarked our strategies against existing attack techniques for OLTR, including the attack-then-quit strategies (CascadeATQ and PBMATQ), adapted from the one given in [18], which is effective against arm-elimination based algorithms but fails to mislead UCB-based algorithms. We also compared against the AlphaATk strategies (CascadeAlphaAtk and PBMAlphaAtk) from [17], which require access to the user feedback received by the algorithm in each round.

CascadeATQ and PBMATQ attacks were configured in a manner to require the same amount of reward manipulation as `CascadeOFA` and `PBMOFA`, respectively. For both the AlphaAtk strategies, parameters Δ_0 and δ (defined in [17]) are set to 0.1, following Appendix B.1 in [17].

We present the required reward manipulations and the corresponding regret implications for the aforementioned attack strategies in Tables 1a and 1b. While our strategies and the AlphaAtk strategies significantly increase the regret of OLTR algorithms, the regret induced by the ATQ strategies remains relatively limited compared to the amount of reward manipulation they require. These results are qualitatively consistent with Figure 2.

Having shown the efficacy of our attack strategies for UCB-based algorithms, we conducted empirical experiments with some of the other popular OLTR algorithms as well, using the same parameters as earlier. Figure 3 presents the results for CascadeKL-UCB and TS-Cascade in the presence of `CascadeOFA`, with $\mathrm{C}_1 = 10260$ and $\mathrm{C}_2 = 1005$ (same as earlier). `CascadeOFA` is able to mislead both the OLTR algorithms into recommending the target items for a large majority of rounds.

Motivated by the empirical results, we plan to extend our theoretical analysis of `CascadeOFA` to other Cascade model-based OLTR algorithms, with the goal of establishing regret bounds and recommendation guarantees for algorithms such as CascadeKL-UCB and TS-Cascade.