Stochastic Bandits with Non-stationary Rewards: Reward Attack and Defense

Chenye Yang, Guanlin Liu and Lifeng Lai

***Abstract*—In this paper, we investigate rewards attacks on stochastic multi-armed bandit algorithms with non-stationary environment. The attacker's goal is to force the victim algorithm to choose a suboptimal arm most of the time while incurring a small attack cost. We consider three increasingly general attack scenarios, each of which has different assumptions about the environment, victim algorithm and information available to the attacker. We propose three attack strategies, one for each considered scenario, and prove that they are successful in terms of expected target arm selection and attack cost. We also propose a defense non-stationary algorithm that is able to defend any attacker whose attack cost is bounded by a budget, and prove that it is robust to attacks. The simulation results validate our theoretical analysis.**

***Index Terms*—bandit, non-stationary reward, dynamic regret, attack and defense**

I. Introduction

Multi-armed bandit (MAB) problems is a class of sequential decision-making problems that have wide range of applications. This class of problems model the scenario where an agent algorithm must choose among multiple arms to maximize its cumulative reward. They have been applied to various fields, including online advertisement (optimizing ad selection), healthcare (personalized treatment strategies), and recommender systems (improving personalized recommendations). Existing works [**?**], [**?**], [**?**], [**?**] have identified potential security issues of existing MAB algorithms. In particular, these work show that an attacker can force the existing MAB algorithms to take unwanted actions, e.g., choose a suboptimal target arm, and may lead to severe real-world consequences (unfair business competition, health threats etc.).

Prior works investigate two attack methods: manipulate the reward signal [**?**], [**?**], [**?**] or manipulate the action signal [**?**] [**?**]. Most of these existing work focused on the traditional stationary random rewards setting, in which the distribution of reward of each arm does not change over time. Some [**?**] studied the adversarial setting, in which the reward given by the environment can be arbitrarily chosen. It is important to note that in many real-world applications, the reward distribution may change over time, but with a specific restriction on the extent of changes. For example, the best product recommendation may vary when the user's interest slightly changes on an e-commerce platform. The subtle changes of preference could be seasonal in a year, and could also be influenced by real world events like the Christmas holiday. In this case, it is more appropriate to model the problem as stochastic multi-armed bandit with non-stationary rewards [**?**]. In this paper, we study the reward attack on non-stationary MAB algorithms in the non-stationary reward setting with restriction on the extent of changes.

The non-stationary reward structure will introduce additional challenges on both the algorithm side and the attacker side. In addition to the exploitation and exploration trade-off, the algorithm also needs to handle trade-off between 'remembering' and 'forgetting' since the estimation of expected rewards is based on past rewards observations, and will have larger overall regret since the best arm is always changing. At the same time, this also creates additional challenges for the attack design as well, as it costs more for the attacker to perform a successful attack, since every time when the victim algorithm 'forgets' history it tends to 'explore' all possible arms instead of 'exploit' the target arm. To our knowledge, this is the first work to successfully attack those specifically designed non-stationary MAB algorithm with a variation budget, which models the temporal uncertainty and changes in the non-stationary reward environment.

In this paper, three attack scenarios targeting non-stationary MAB algorithm are considered. The first scenario only has very general assumption on the environment side, but assumes that the attacker has detailed knowledge of the victim algorithm's behavior. The second scenario removes the assumption on the attacker's knowledge of the victim algorithm. The third scenario then removes the restriction on the victim algorithm side, covering more victim algorithms in practice. For each scenario, we propose the corresponding attack strategy and prove them to be successful.

Attack and defense always go hand in hand. With the existence of the attack, it is important to design defense strategies to protect the algorithm from taking unwanted actions. Existing works focus on designing robust algorithms for stochastic bandit algorithms [**?**], [**?**], [**?**], [**?**] and adversarial bandit algorithms [**?**] under adversarial attacks. Some work also studies stochastic linear contextual bandits [**?**], stochastic bandits to strategic manipulations [**?**], and stochastic Lipschitz bandits [**?**]. It is important to study the defense algorithm in the non-stationary reward environment, which models the extent of changes in rewards. This is not only because the non-stationary environment is common in real-world applications, but also because we have already designed successful attack strategies against the current non-stationary MAB algorithms

C. Yang, G. Liu and L. Lai are with the Department of Electrical and Computer Engineering, University of California, Davis, CA. This work was supported by the National Science Foundation under Grants ECCS-2000415 and CCF-2232907. This work was presented in part at the 2023 Asilomar Conference on Signals, Systems, and Computers [**?**]. Email:{cyyyang, glnliu, lflai}@ucdavis.edu.

in this paper. In this paper, we propose a defense algorithm that is able to defend any attacker whose attack cost is bounded by a budget.

Compared with our conference paper [**?**], this journal paper has the following new contributions: 1) Besides the attack scenarios in [**?**], we consider a more general one which covers more victim algorithms and prove the attack strategy to be successful; 2) We propose a defense algorithm, *RexpRb*, and prove it to be robust to attacks; 3) We conduct more comprehensive simulations to validate our theoretical analysis.

The remainder of the paper is organized as follows. In Section II, we introduce the problem formulation. In Section III, we focus on the attack strategy design. In Section IV, we design the defense algorithm. In Section V, we provide simulation results. Finally, we conclude the paper in Section VI.

II. Problem Formulation

A. Multi-armed bandit problem

Let $\mathcal{K} = \{1, 2, \ldots, K\}$ be the set of arms to be pulled (decisions to be made), $\mathcal{T} = \{1, 2, \ldots, T\}$ be the sequence of decision steps for the decision maker (agent). The agent pulls an arm $a_t \in \mathcal{K}$ at step $t \in \mathcal{T}$ and receives a reward $X_t(a_t)$ which is generated by the environment. $X_t(a_t) \in [0, 1]$ is a random variable with expectation $\mathbb{E}[X_t(a_t)]$. The goal of the agent is to maximize the total expected reward over a long time, while balancing exploration and exploitation.

B. Non-stationary environment

In many practical cases, the reward distributions of the arms in an MAB problem may change over time. In the existing works, there are two popular approaches to model the changing reward: adversarial environments [**?**], [**?**] and non-stationary environments [**?**], [**?**], [**?**], [**?**], [**?**]. The adversarial environment allows the reward distribution to be arbitrarily chosen by the environment. The downside of this model is that it does not restrict the extent of changes and thus may not capture the temporal uncertainty of the reward distribution. Another popular approach is the non-stationary environment which models the temporal uncertainty in the following two ways: allow a finite number of changes in the expected reward [**?**], [**?**], or allow a bounded total variation of expected reward over the relevant time [**?**], [**?**], [**?**]. In this paper, we will perform attack within the non-stationary environment with a bounded total variation of the expected rewards.

Denote $\mu_t^k = \mathbb{E}[X_t(a_t = k)]$ and $\mu_t^* = \max_{k \in \mathcal{K}}\{\mu_t^k\}$, where $\mathbb{E}$ is taken with respect to reward $X_t(a_t)$ at step t. In this paper, we focus on the non-stationary environment with a bounded total variation of the expected rewards $\mathbb{E}[X_t(a_t)]$:

$$\sum_{t=1}^{T-1} \sup_{k \in \mathcal{K}} \left| \mu_t^k - \mu_{t+1}^k \right| \leq V_T,$$

where V_T is the variation budget for the entire horizon T of the problem. We define the temporal uncertainty set as the set of reward sequences that are subject to the variation budget V_T over the set of step epochs $\{1, \ldots, T\}$: $\mathcal{V} = \left\{ \mu \in [0,1]^{K \times T} : \sum_{t=1}^{T-1} \sup_k \left| \mu_t^k - \mu_{t+1}^k \right| \leq V_T \right\}$. Note that $V_T = 0$ corresponds to the stationary environment while $V_T = O(T)$ corresponds to the adversarial environment [**?**].

C. MAB algorithm performance metrics

The performance of a multi-armed bandit algorithm is measured by its regret. For the adversarial bandit problems [**?**], [**?**], the regret R_T is defined against the static oracle, which is the best arm in hindsight over the whole horizon.

$$R_T = \max_a \sum_{t=1}^{T} X_t(a) - \mathbb{E}^\pi \left[\sum_{t=1}^{T} X_t\left(a_t\right) \right].$$

One may use adaptive algorithms such as *Exp3* to handle adversarial environments, of which the static oracle regret is $O(\sqrt{KT \log K})$ [**?**].

In the non-stationary setting, the regret of the algorithm over the entire horizon is defined as the worst case difference between the expected performance of pulling the arm which has the highest expected reward at each epoch t and the expected performance under policy π, which is also known as the regret measured against the dynamic oracle [**?**], [**?**]:

$$R^\pi(\mathcal{V}, T) = \sup_{\mu \in \mathcal{V}} \left\{ \sum_{t=1}^{T} \mu_t^* - \mathbb{E}^\pi \left[\sum_{t=1}^{T} \mu_t^\pi \right] \right\},$$

where $\mathbb{E}^\pi$ is taken with respect to the noisy rewards and policy's actions, and μ_t^π means pulling arm according to π.

D. Reward attack

In this paper, as shown in Figure 1, we consider a setup where there is an attacker that can intercept the reward signal $X_t(a_t)$ from the environment and manipulate it to $\tilde{X}_t(a_t)$. The attacker's goal is to force the agent to choose a suboptimal target arm $a^\dagger$ [1] as often as possible while inducing an attack cost as low as possible. For example, in the e-commerce platform, the target arm $a^\dagger$ could be a specific product that one merchant wants to promote, and the attack cost could be the computational cost of hacking the system, manipulating the rewards from the user and injecting them to the system, as well as the potential legal risk and the loss of trust. All of these costs will increase with the extent of performing reward manipulation.

There are two metrics to measure the performance of the attacker: the expected attack cost $\mathbb{E}^\pi[C_T]$, and the expected number of target arm selection $\mathbb{E}^\pi[N_\mathcal{T}(a^\dagger)]$:

$$\mathbb{E}^\pi[C_T] = \mathbb{E}^\pi \left[\sum_{t=1}^{T} \left| \tilde{X}_t\left(a_t\right) - X_t\left(a_t\right) \right| \right],$$

$$\mathbb{E}^\pi[N_\mathcal{T}(a^\dagger)] = \mathbb{E}^\pi \left[\sum_{t=1}^{T} \mathbf{1}\left[a_t = a^\dagger\right] \right].$$

[1] In this paper, we assume that the target arm $a^\dagger$ is a specifically chosen arm and will not change, even though the reward distribution of $a^\dagger$ may change due to the non-stationarity. It is also possible to extend the problem to the case where $a^\dagger$ changes over time. Then, the entire horizon can be divided into several stages at the time points when $a^\dagger$ changes.

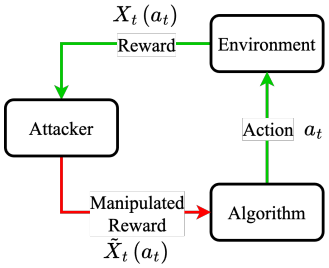


Fig. 1: Reward Attack

The goal of the attacker is to maximize the $\mathbb{E}^\pi[N_\mathcal{T}(a^\dagger)]$ while incurring a small $\mathbb{E}^\pi[C_T]$.

III. Attack

In this section, we consider three increasingly complex attack scenarios and propose three attack strategies, one for each scenario. Here, we assume that the non-stationary victim algorithm is unaware of the existence of the attacker.

A. Non-stationary Victim Algorithm

If there is no attacker, a widely used strategy for stochastic non-stationary MAB problem is *Rexp3*, proposed in [**?**], which is able to handle the temporal uncertainty by variation budget V_T. [**?**] proves that *Rexp3* is nearly minimax optimal with dynamic oracle regret of order $V_T{}^{1/3}T^{2/3}$. For completeness, this algorithm is listed in Algorithm 1.

Algorithm 1 *Rexp3 [?]*

Parameters: A learning rate η, and a batch size Δ_T.
for Batch $j = 1, 2, \ldots, m = \lceil \frac{T}{\Delta_T} \rceil$ **do**
  Initialization: $w_{t,a} = 1, \forall a \in \mathcal{K}$.
  for $t = (j-1)\Delta_T + 1 \le t \le \min\{j\Delta_T, T\}$ **do**
    Define $\pi_{t,a} = (1-\eta)\frac{w_{t,a}}{\sum_a w_{t,a}} + \frac{\eta}{K}$
    Draw $a_t \sim \{\pi_{t,a}\}$, and observe reward $X_t(a_t)$
    for $a = 1, 2, \ldots, K$ **do**

$$w_{t+1,a} = \begin{cases} w_{t,a} & , a \neq a_t \\ w_{t,a} \exp\left(\frac{\eta}{K}\frac{X_t(a_t)}{\pi_{t,a}}\right) & , a = a_t \end{cases}$$

    end for
  end for
end for

Figure 2 illustrates the *Rexp3* algorithm. In *Rexp3*, to handle the non-stationary environment, the total horizon $\mathcal{T}$ is split into many batches $(\mathcal{T}_1, \ldots, \mathcal{T}_m)$ with fixed size Δ_T each (except, possibly the last batch):

$$\mathcal{T}_j = \{t : (j-1)\Delta_T + 1 \le t \le \min\{j\Delta_T, T\}\}, \quad (1)$$

$\forall j = 1, \ldots, m$, where $m = \lceil \frac{T}{\Delta_T} \rceil$ is the number of batches. In each batch, one runs the *Exp3* algorithm. Furthermore, the *Exp3* algorithm restarts itself at the beginning of each batch, to forget all its memory and handle the changing environment.

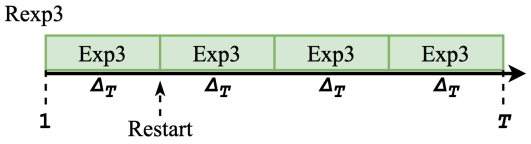


Fig. 2: Timeline of Rexp3

With the assumption that V_T is known to algorithm, *Rexp3* chooses the batch size $\Delta_T = \left\lceil (K\log K)^{1/3}(T/V_T)^{2/3} \right\rceil$, and achieves the dynamic oracle regret mentioned above.

Note that in the j-th batch $\mathcal{T}_j$, dynamic regret $R^\pi\left(\mathcal{V}, \Delta_{\mathcal{T}_j}\right)$ is defined as $\mathbb{E}^\pi\left[\sum_{t\in\mathcal{T}_j}\left(\mu_t^* - \mathbb{E}\left[X_t\left(a_t^\pi\right)\right]\right)\right]$, and static regret $R_{\Delta_{\mathcal{T}_j}}$ is defined as $\max_a \sum_{t\in\mathcal{T}_j} X_t(a) - \mathbb{E}^\pi\left[\sum_{t\in\mathcal{T}_j} X_t(a_t)\right]$. We assume the MAB algorithm running and restarting in each batch has static oracle regret $R_{\Delta_{\mathcal{T}_j}} = O\left({\Delta_{\mathcal{T}_j}}^\alpha\right)$ for some $\alpha \in [\frac{1}{2}, 1)$ in each batch $\mathcal{T}_j$, which holds for commonly used MAB algorithms such as *Exp3* with $\alpha = \frac{1}{2}$ [**?**].

Specifically, we will attack this *Rexp3* algorithm in the non-stationary reward setting with a variation budget V_T under three increasingly relaxed assumptions.

B. Attack Scenario I

In this scenario, there's no constraint on the environment side, which means that sometimes $X_t(a^\dagger) = 0$. We assume the attacker has information of when the victim algorithm restarts, i.e., the batch size Δ_T.

We present our attack scheme in Algorithm 2. The scheme keeps a diminish function $\tilde{t}^{\alpha+\epsilon-1}$ as the lower-bound of manipulated reward for the target arm, where $\tilde{t}$ is the relative step in each batch and $\epsilon < 1-\alpha$. Within each batch, if the victim algorithm chooses a non-target arm, the attacker will reduce the reward to 0. However, if the victim algorithm chooses the target arm, the attacker will manipulate the reward to the maximal value between the original reward and $\tilde{t}^{\alpha+\epsilon-1}$. The relative step $\tilde{t}$ will increase along with the absolute step $t \in \mathcal{T}$, but will be reset to 1 at the beginning of a new batch. The attacker is able to restart the diminish function simultaneously with the algorithm by the assumption of knowing Δ_T. In other words, the manipulated reward $\tilde{X}_t(a_t)$ will be:

$$\tilde{X}_t(a_t) = \begin{cases} \max\left\{\tilde{t}^{\alpha+\epsilon-1}, X_t(a_t)\right\} & a_t = a^\dagger \\ 0 & a_t \neq a^\dagger \end{cases}. \quad (2)$$

The purpose of using a diminishing function as the minimum for the manipulated reward $\tilde{X}_t\left(a^\dagger\right)$ of the target arm $a^\dagger$ is to: 1) give more reward to $a^\dagger$ for the first several times it is selected, which can make the victim algorithm more likely to exploit $a^\dagger$ in the future, instead of exploring other arms; 2) reduce the minimum of $\tilde{X}_t\left(a^\dagger\right)$ gradually, which helps to reduce the manipulations to reward when more and more times $a^\dagger$ is selected in the future; 3) make the attack successful even when the reward of target arm $a^\dagger$ is always 0, i.e., $X_t(a^\dagger) = 0$. Otherwise, the victim algorithm can not distinguish all the arms after attack and will explore them equally.

Algorithm 2 Attack Scenario I

Initialize total horizon T
Let absolute step $t = 1$, let relative step $\tilde{t} = 1$
while $t \leq T$ **do**
  Victim algorithm, e.g., *Rexp3*, chooses arm a_t
  Environment gives the reward $X_t(a_t)$
  Attacker manipulates the reward:
  if $a_t = a^\dagger$ **then**
    $\tilde{X}_t(a_t) = \max\{\tilde{t}^{\alpha+\epsilon-1}, X_t(a_t)\}$
  else
    $\tilde{X}_t(a_t) = 0$
  end if
  if $\tilde{t} < \Delta_T$ **then** ▷ $\tilde{t} < T - (m-1)\Delta_T$ for last batch
    $\tilde{t} = \tilde{t} + 1$ ▷ continue the diminish function
  else
    reset $\tilde{t} = 1$ ▷ restart the diminish function
  end if
  Victim updates itself based on $\tilde{X}_t(a_t)$
  $t = t + 1$
end while

This following theorem provides performance bounds of the proposed attack strategy for scenario I, with known fixed Δ_T.

Theorem III.1. *Suppose the victim algorithm has static oracle regret $R_{\Delta_T} = O\left({\Delta_T}^{\alpha}\right)$ for some $\alpha \in [\frac{1}{2}, 1)$ in each batch and follows the fixed batch strategy to handle non-stationary reward. Then the expected number of target arm selection and the expected attack cost of the attack scheme in Algorithm 2 satisfy:*

$$\mathbb{E}^{\pi}\left[N_{\mathcal{T}}\left(a^{\dagger}\right)\right] \geq T - \left(2MT{\Delta_T}^{-\epsilon} + \frac{2}{\alpha+\epsilon}T{\Delta_T}^{-\alpha-\epsilon}\right),$$
$$\mathbb{E}^{\pi}\left[C_T\right] \leq 2MT{\Delta_T}^{-\epsilon} + \frac{2}{\alpha+\epsilon}T{\Delta_T}^{-\alpha-\epsilon} + \frac{2}{\alpha+\epsilon}T{\Delta_T}^{\alpha+\epsilon-1}, \tag{3}$$

where M is a constant, and $\epsilon < 1 - \alpha$.

Theorem III.1 reveals that if the victim algorithm sets batch size as $\Delta_T = B\left(\frac{T}{V_T}\right)^{\beta}$, where $\beta \in [0, 1]$, and B is a constant, the expected attack cost $\mathbb{E}^{\pi}\left[C_T\right]$ will be:

$$\mathbb{E}^{\pi}\left[C_T\right] = O\left({V_T}^{\epsilon\beta}T^{1-\epsilon\beta} + {V_T}^{\beta(1-\alpha-\epsilon)}T^{1-\beta(1-\alpha-\epsilon)}\right).$$

When the variation budget V_T is sublinear in T, i.e., $V_T = O(T^{\xi})$ where $\xi \in (0, 1)$, the expected attack cost $\mathbb{E}^{\pi}[C_T]$ will be:

$$\mathbb{E}^{\pi}\left[C_T\right] = O\left(T^{1-(1-\xi)\epsilon\beta} + T^{1-(1-\xi)\beta(1-\alpha-\epsilon)}\right),$$

which is sublinear in T. Meanwhile, the target arm selection $\mathbb{E}^{\pi}\left[N_{\mathcal{T}}\left(a^{\dagger}\right)\right]$ will increase almost linear in T. This means that the attack is successful. Without knowing the existence of the attacker, the victim algorithm will pick the target arm most of the time. Meanwhile, the attack cost remains small. For example, when $\alpha = \frac{1}{2}$, $\epsilon = \frac{1}{5}$, $\beta = \frac{2}{3}$, and $\xi = \frac{1}{10}$, which are the same as those in experiments, the expected attack cost $\mathbb{E}^{\pi}\left[C_T\right] = O(T^{0.88})$.

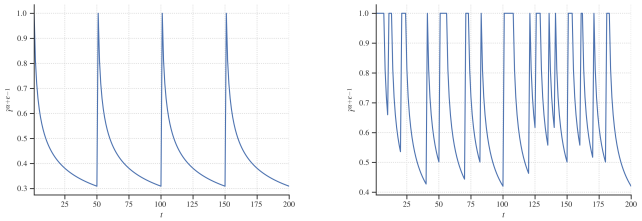
Note that in Theorem III.1, $\mathbb{E}^{\pi}\left[N_{\mathcal{T}}\left(a^{\dagger}\right)\right]$ and $\mathbb{E}^{\pi}\left[C_T\right]$ depend only on the batch size Δ_T and the order of static oracle regret α of the victim algorithm, rather than other properties on the algorithm side. This makes Theorem III.1 a general result for attacking any non-stationary MAB algorithm with $\alpha \in [\frac{1}{2}, 1)$ and fixed batch size Δ_T strategy. It will cover more victim algorithms than *Rexp3* in Section III-A.

C. Attack Scenario II

In this section, we relax the assumption on the attacker's knowledge about Δ_T, and assume that the attacker does NOT know when the algorithm restarts, i.e., the attacker does not know the batch size Δ_T.

In this scenario, the attacker follows the scheme in Algorithm 3. It also keeps a diminish function $\tilde{t}^{\alpha+\epsilon-1}$. If the victim algorithm chooses a non-target arm, the attacker will reduce the reward to 0, and reset the relative step $\tilde{t}$ to 1 (including for this step although $\tilde{t}$ not used). However, when the target arm is chosen, the attacker will manipulate the reward to the maximal value between the original reward and $\tilde{t}^{\alpha+\epsilon-1}$, and then increase the relative step $\tilde{t}$ by 1. Similarly, the manipulated reward $\tilde{X}_t\left(a_t\right)$ will be as (2).

The main difference of the attacker's behavior is the reset timing of the diminishing function. In scenario I, the attacker is able to reset the diminishing function at the beginning of each batch simultaneously with the victim algorithm. However, in scenario II, the attacker does not know Δ_T, and thus will reset the diminishing function anytime when a non-target arm is pulled. Figure 3 illustrates how the diminish functions behave in the two scenarios.



(a) Attack Scenario I (b) Attack Scenario II

Fig. 3: Comparisons of the diminishing functions

When the diminishing function is reset, the manipulated reward of the next target arm pull will be $\max\left\{\tilde{t}^{\alpha+\epsilon-1}, X_t(a_t)\right\} = 1$. Consider that $X_t(a_t) \leq 1\ \forall a, t$, the above strategy helps to give more reward to the target arm when it is selected after any non-target arm pull. There's always a possibility for any arm been selected because of the 'exploration' of bandit algorithm. Thus, the diminishing function will be reset more frequently, instead of resetting only at the beginning of each batch. This attack strategy ensures that the victim algorithm receives larger target arm rewards and thus tends to choose $a^{\dagger}$ to minimize its regret, especially

for those whose probability of choosing an arm is weighted by its reward, e.g., *Rexp3*. This will make the victim algorithm choose the target arm with higher probability compared with that in scenario I in Section III-B.

Algorithm 3 Attack Scenario II & III

Initialize total horizon T
Let absolute step $t = 1$, let relative step $\tilde{t} = 1$
while $t \leq T$ **do**
  Victim algorithm, e.g., *Rexp3*, chooses arm a_t
  Environment gives the reward $X_t(a_t)$
  Attacker manipulates the reward $\tilde{X}_t(a_t)$:
  if the chosen arm is the target arm: $a_t = a^\dagger$ **then**
    $\tilde{X}_t(a_t) = \max\{\tilde{t}^{\alpha+\epsilon-1}, X_t(a_t)\}$
    $\tilde{t} = \tilde{t} + 1$ ▷ continue the diminish function
  else if $a_t \neq a^\dagger$ **then**
    $\tilde{X}_t(a_t) = 0$
    reset $\tilde{t} = 1$ ▷ restart the diminish function
  end if
  Victim updates itself based on $\tilde{X}_t(a_t)$
  $t = t + 1$
end while

The following theorem illustrates the performance of the proposed attack strategy for scenario II, with unknown fixed Δ_T to the attacker.

Theorem III.2. *Under the same assumption as in Theorem III.1, the expected number of target arm selection and the expected attack cost of the attack scheme in Algorithm 3 satisfy*

$$\mathbb{E}^\pi\left[N_\mathcal{T}\left(a^\dagger\right)\right] \geq T - 2MT{\Delta_T}^{\alpha-1},$$
$$\mathbb{E}^\pi\left[C_T\right] \leq 2MT{\Delta_T}^{\alpha-1} + \frac{(2M)^{1-\alpha-\epsilon}}{\alpha+\epsilon}T{\Delta_T}^{(1-\alpha)(\alpha+\epsilon-1)},$$

where M is a constant, and $\epsilon < 1 - \alpha$.

With the same setting for Δ_T and V_T as in Section III-B, Theorem III.2 reveals that the expected attack cost:

$$\mathbb{E}^\pi\left[C_T\right] = O\left(T^{1-(1-\xi)\beta(1-\alpha)} + T^{1-(1-\xi)\beta(1-\alpha)(1-\alpha-\epsilon)}\right),$$

which is sublinear in T. Meanwhile, the target arm selection $\mathbb{E}^\pi\left[N_\mathcal{T}\left(a^\dagger\right)\right]$ will increase almost linear in T, meaning that the attack is successful. For example, when the parameters are the same as those in experiments, the expected attack cost $\mathbb{E}^\pi\left[C_T\right] = O(T^{0.91})$.

Theorem III.2 also covers a wide range of non-stationary victim algorithms, similar to Theorem III.1.

Note that one can not conclude that the attack cost in scenario II is larger than that in scenario I for arbitrary parameters. The reason is that, for the big-O expression, in scenario II, the first term has a smaller order, while the second term has a larger order, compared with the corresponding terms in scenario I. The relation between these two terms depends on the choice of β, ξ, and ϵ. Moreover, these theorems provide only bounds rather than exact function forms of the expected attack cost, and the constant factors are also different.

D. Attack Scenario III

In this section, we further relax the restriction on the algorithm side and assume that the batch size Δ_{T_j}s are no longer the same, making the problem more general and closer to the real-world scenarios, in which the non-stationary algorithm may restart at any specific time to learn the changing environment. In particular, we assume the victim algorithm like *Rexp3* can restart at any time, as illustrated in Figure 4. Moreover, the attacker does not know when the algorithm restarts, i.e., the batch sizes Δ_{T_j}. In this scenario, the attacker follows the same scheme in Algorithm 3. Similarly, the manipulated reward $\tilde{X}_t\left(a_t\right)$ will be as (2).

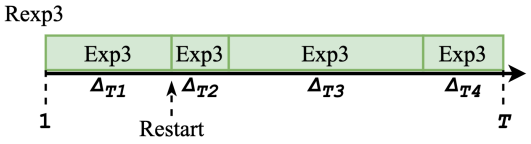


Fig. 4: Timeline of Rexp3 with non-fixed batch size

The following theorem provides performance bounds of the proposed attack strategy for scenario III, with unknown non-fixed Δ_{T_j} to the attacker. Although the attack strategy is the same as in previous cases, the analysis of the attack cost and the target arm selection is different and more challenging.

Theorem III.3. *Assume the victim algorithm has static oracle regret $R_{\Delta_{T_j}} = O\left({\Delta_{T_j}}^\alpha\right)$ for some $\alpha \in [\frac{1}{2}, 1)$ in each batch and follows the non-fixed batch strategy to handle non-stationary reward. The expected number of target arm selection and the expected attack cost of the attack scheme specified in Algorithm 3 satisfies:*

$$\mathbb{E}^\pi\left[N_\mathcal{T}\left(a^\dagger\right)\right] \geq T - MT{\delta_T}^{\alpha-1},$$
$$\mathbb{E}^\pi\left[C_T\right] \leq MT{\delta_T}^{\alpha-1} + \frac{M^{1-\alpha-\epsilon}}{\alpha+\epsilon}T{\delta_T}^{(1-\alpha)(\alpha+\epsilon-1)},$$

where M is a constant, $\epsilon < 1 - \alpha$, and $\delta_T = \min_j\left\{\Delta_{T_j}\right\}$ is the minimal batch size.

With $\delta_T = B\left(\frac{T}{V_T}\right)^\beta$ where $\beta \in [0, 1]$ and B is a constant, and the same setting for V_T as in Section III-B, Theorem III.3 reveals that the expected attack cost:

$$\mathbb{E}^\pi\left[C_T\right] = O\left(T^{1-(1-\xi)\beta(1-\alpha)} + T^{1-(1-\xi)\beta(1-\alpha)(1-\alpha-\epsilon)}\right),$$

which is sublinear in T. Meanwhile, the target arm selection $\mathbb{E}^\pi\left[N_\mathcal{T}\left(a^\dagger\right)\right]$ will increase almost linear in T, meaning that the attack is successful. For example, when the parameters are the same as those in experiments, the expected attack cost $\mathbb{E}^\pi\left[C_T\right] = O(T^{0.91})$.

In scenario III, we relax the restriction on the algorithm side, and no longer require the batch size to be fixed. Thus, the results in Theorem III.3 are more general and similar to real-world applications, and can be applied to a larger class of non-stationary victim algorithms than those in Theorem III.2.

In the special non-stationary case when all the m batches have the same size Δ_T, i.e., $\Delta_{T_1} = \Delta_{T_2} = \ldots = \Delta_{T_m} = \Delta_T$,

Theorem III.2 and Theorem III.3 are identical. This can be seen by using $m = \frac{T}{\Delta_T}$ in the proof of Theorem III.2 and using $\delta_T = \Delta_T$ in the proof of Theorem III.3.

E. Lower-bound of the Expected Attack Cost for Fixed Δ_T

We have shown the upper-bound of the expected attack cost and lower-bound of the expected target arm selection of our attack strategies in Theorem III.1 and III.2, proving that our attackers can successfully control the victim's behavior and induce a small cost. In this section, we show that our attack strategies are near optimal. In particular, we show that if an attacker achieves $T - o(T)$ expected target arm selection, and it is also victim-agnostic to non-stationary bandit algorithm, then the attacker must induce at least expected attack cost $\Omega\left(T{\Delta_T}^{\alpha-1}\right)$. Here, victim-agnostic means that the attacker does not know what is exactly the victim algorithm, but only knows that the non-stationary algorithm has sublinear static oracle regret in each batch and follows the batch strategy.

Since we are looking for the victim-agnostic lower-bound, it is sufficient to pick a particular victim non-stationary algorithm that guarantees $O({\Delta_T}^{\alpha})$ static regret in each batch, under one bandit environment. Then we need to show that any victim-agnostic attacker must induce at least some attack cost to achieve $T - o(T)$ expected target arm selection on this particular victim algorithm. The main result for lower-bound of the expected attack cost is provided in Theorem III.4.

Theorem III.4. *Assume some victim-agnostic attack algorithm achieves $\mathbb{E}^{\pi}\left[N_T(a^{\dagger})\right] = T - o(T)$ on all victim bandit algorithms that has static oracle regret $O\left({\Delta_T}^{\alpha}\right)$ in each batch and follows the fixed batch strategy to handle non-stationary reward, where $\alpha \in \left[\frac{1}{2}, 1\right)$. Then there exists a bandit task such that the attacker must induce at least expected attack cost $\mathbb{E}^{\pi}\left[C_T\right] = \Omega\left(T{\Delta_T}^{\alpha-1}\right)$ on some victim algorithm, where Δ_T is the fixed batch size.*

Theorem III.4 reveals that the best achievable performance of attacker is $\Omega\left(T{\Delta_T}^{\alpha-1}\right)$ in fixed batch size cases. Thus, for scenarios considered in Section III-B and Section III-C, our methods are near optimal except with a small additional cost depending on the choice of parameters β and ϵ.

IV. Defense

In this section, we first summarize different types of regret, which are used as the performance metric of robust MAB algorithms under reward attack, and propose a new one: the dynamic oracle regret with original rewards. Then, we propose a defense non-stationary MAB algorithm which is robust to attackers with bounded attack cost. We assume that the defense algorithm is aware of the existence of the attacker. However, it does not know the attacker's attack strategy.

Even with attack, the regret of the defense algorithm should still be evaluated against the original rewards μ_t, instead of the manipulated rewards $\tilde{\mu}_t$, to minimize the difference between the original best action's reward and the original reward of the chosen action (based on the manipulated policy $\tilde{\pi}$ after attack). In this way, a small regret represents that the defense algorithm is less affected by the attack, and robust to the attack. Thus, the dynamic oracle regret under attack becomes:

$$R^{\tilde{\pi}}\left(\mathcal{V}, T\right) = \sup_{\mu \in \mathcal{V}} \left\{ \sum_{t=1}^{T} \mu_t^* - \mathbb{E}^{\pi}\left[\sum_{t=1}^{T} \mu_t^{\tilde{\pi}}\right] \right\}.$$

Here's the explanation: As we assume the agent is aware of the existence of the attack, the MAB algorithm will have a different policy $\tilde{\pi}$. However, we are anticipating that there's a defense algorithm that can still take the original best actions as much as possible, leading to larger cumulative original reward sent out by the environment (not exactly the reward received by the defense algorithm due to attack). For example, in the medical recommender system, it is vital to base decisions on real feedback from the patient, not on the manipulated feedback. The recommender wants to take the action that is best for the patient and has the largest real feedback, instead of the action with largest manipulated feedback since it may actually be dangerous to the patient.

The goal of defense algorithm is to minimize the dynamic oracle regret with original rewards $R^{\tilde{\pi}}\left(\mathcal{V}, T\right)$.

Note that this regret is different from $\tilde{R}_T$ defined in [**?**], which should be named as the static oracle regret evaluated with the manipulated rewards $\tilde{X}_t$ and against the original rewards X_t, with our notation: $\tilde{R}_T = \max_a \sum_{t=1}^{T} X_t(a) - \mathbb{E}^{\pi}\left[\sum_{t=1}^{T} \tilde{X}_t\left(a_t\right)\right]$. The results in [**?**] are not applicable to our defense algorithm, due to the different regret definitions and [**?**]'s adversarial bandit settings.

To make the notations clear, we summarize all the commonly used regrets in Table I. The dynamic oracle regret with original rewards [1] is the performance metric of the defense algorithm under attack considered in this paper. The dynamic oracle regret with manipulated rewards [2] represents that the victim algorithm is not aware of the existence of the attacker, and thus trying to behave the best as it can under the manipulated reward environment. This regret is indirectly used in the design of the attacker for victim algorithms in Section III, and can be found in the corresponding proofs. The static oracle regret with original rewards [3] is the performance metric of adversarial algorithms. The static oracle regret with the manipulated rewards $\tilde{X}_t$ and against the original rewards X_t [4] is that considered when designing the defense adversarial algorithm in the existing work [**?**].

In practical applications, the attacker may have a budget on the attack cost. Denote $\Phi(T)$ as the maximum attack cost allowed by the attacker over the entire horizon T of the problem. We have that $\mathbb{E}^{\pi}[C_T] \leq \Phi(T)$. These attackers with bounded attack cost could be defended by particularly designed defense algorithms. On the contrary, if an attacker has unbounded attack cost, there's no algorithm can defend it, since the reward can always be arbitrarily changed.

Here, we propose a defense bandit algorithm, *RexpRb*, in the non-stationary reward environment, which is robust to reward attacks. The *RexpRb* algorithm is described in Algorithm 4.

In *RexpRb*, to handle the non-stationary environment, the total horizon $\mathcal{T}$ is split into $m = \lceil \frac{T}{\Delta_T} \rceil$ batches $(\mathcal{T}_1, \ldots, \mathcal{T}_m)$

TABLE I: Regrets under attack, with the manipulated policy $\tilde{\pi}$ (W/ means With, A/ means Against)

	Dynamic Oracle	**Static Oracle**
W/ & A/ Original Rewards	$\sup_{\mu\in\mathcal{V}}\left\{\sum_{t=1}^{T}\mu_t^* - \mathbb{E}^{\pi}\left[\sum_{t=1}^{T}\mu_t^{\tilde{\pi}}\right]\right\}$ [1]	$\max_a \sum_{t=1}^{T} X_t(a) - \mathbb{E}^{\pi}\left[\sum_{t=1}^{T} X_t\left(a_t\right)\right]$ [3]
W/ Manipulated & A/ Original Rewards		$\max_a \sum_{t=1}^{T} X_t(a) - \mathbb{E}^{\pi}\left[\sum_{t=1}^{T} \tilde{X}_t\left(a_t\right)\right]$ [4]
W/ & A/ Manipulated Rewards	$\sup_{\mu\in\mathcal{V}}\left\{\sum_{t=1}^{T}\tilde{\mu}_t^* - \mathbb{E}^{\pi}\left[\sum_{t=1}^{T}\tilde{\mu}_t^{\tilde{\pi}}\right]\right\}$ [2]	

Algorithm 4 *RexpRb*

Parameters: Learning rate η, robustness parameter γ, and batch size Δ_T.
for Batch $j = 1, 2, \ldots, m = \lceil \frac{T}{\Delta_T} \rceil$ **do**
  Initialization: $w_{0,a} = 1, q_{0,a} = 1, \forall a \in \mathcal{K}$.
  for $t = (j-1)\Delta_T + 1 \le t \le \min\{j\Delta_T, T\}$ **do**
    Define $\pi_{t,a} = (1-\eta)\frac{w_{t,a}}{\sum_a w_{t,a}} + \frac{\eta}{K}$
    Draw $a_t \sim \{\pi_{t,a}\}$, and observe reward $X_t(a_t)$
    Set $\delta_t = 0$
    if $\pi_{t,a_t} < q_{t-1,a_t}$ **then**
      Set $\delta_t = \min\{\gamma(1 - \pi_{t,a_t}/q_{t-1,a_t}), 1\}$
      Update $q_{t,a}$ for $a = 1, 2, \ldots, K$:

$$q_{t,a} = \begin{cases} \max\{\pi_{t,a}, (1-1/\gamma)q_{t-1,a}\} & , a = a_t \\ q_{t-1,a} & , a \neq a_t \end{cases}$$

    end if
    Update $w_{t,a}$ for $a = 1, 2, \ldots, K$:

$$w_{t,a} = \begin{cases} w_{t-1,a} & , a \neq a_t \\ w_{t-1,a}\exp\left(\frac{\eta}{K}\frac{X_t(a_t)+\delta_t}{\pi_{t,a}}\right) & , a = a_t \end{cases}$$

  end for
end for

with fixed size Δ_T each, similar to (1). The *ExpRb* algorithm [?] will restart itself at the beginning of each batch, as illustrated in Figure 5. This batched behavior helps to forget the memory of past and adapt to the changing environment. Note that the variation budget V_T of the environment can be utilized to determine the batch size.

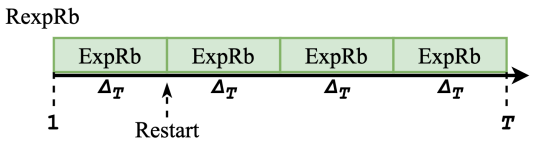


Fig. 5: Timeline of RexpRb

To defend the attacker, similar to *ExpRb* [?] but unlike *Rexp3* [?], the variable $\delta(t)$ is introduced to augment the reward estimation, and a robustness parameter γ is used to connect $\delta(t)$ and the attack cost budget $\Phi(T)$. The design of $\delta(t)$ is motivated by the following intuitions: 1) encourage *RexpRb* to explore more when the reward estimation is not accurate; 2) use the history of reward estimation to handle the potential corrupted rewards in the future; 3) when there's no attacker, *RexpRb* will behave like *Rexp3* by setting robustness parameter $\gamma = 0$.

We now analyze the performance of the defense non-stationary MAB algorithm *RexpRb*.

Theorem IV.1. *Assume that the attacker has attack cost budget $\Phi(T)$, i.e., $\mathbb{E}^{\pi}[C_T] \le \Phi(T)$. The dynamic oracle regret with original rewards $R^{\tilde{\pi}}(\mathcal{V}, T)$ of RexpRb, Algorithm 4, when it is run with parameters $\gamma = \Phi(T)$ and $\eta = O\left(\sqrt{K\log K/\Delta_T}\right)$, is bounded as:*

$$R^{\tilde{\pi}}(\mathcal{V}, T) = O\left(V_T\Delta_T + T\Delta_T^{-\frac{1}{2}} + \Phi(T)\frac{T}{\Delta_T}\log(\Delta_T) + \Phi(T)\right).$$

With the same setting for Δ_T and V_T as in Section III-B, Theorem IV.1 reveals that the dynamic oracle regret will be:

$$R^{\tilde{\pi}}(\mathcal{V}, T) = O\left(T^{\xi-\xi\beta+\beta} + T^{1-\frac{1}{2}\beta+\frac{1}{2}\beta\xi} + \Phi(T)T^{1-\beta+\xi\beta}\log(T) + \Phi(T)\right).$$

It is worth noting that once the ξ in variation budget and $\Phi(T)$ of attack cost budget are given, the defense algorithm *RexpRb* can achieve sublinear dynamic oracle regret $R^{\tilde{\pi}}(\mathcal{V}, T)$ with a specific choice of β in batch size. For example, when $\xi = 0.001$ and $\Phi(T) = O(T^{0.861})$, which are the same as those in experiments, the parameter β can be chosen as $\beta = 0.931$. Then, the regret $R^{\tilde{\pi}}(\mathcal{V}, T) = O\left(T^{0.931}\log(T)\right)$.

V. Experimental Data and Results

A. Attack

We first present experimental results for the attack cases. We consider a bandit problem environment with $K = 5$ arms. The target arm $a^{\dagger} = 1$. The initial expected reward is:

$$\mathbb{E}[X_t(a)] = \begin{cases} 0.1, & a = 1 \\ 0.5, & a = 2, 3, 4 \\ 0.8, & a = 5 \end{cases}.$$

The non-stationary reward structure is simulated by the random walk, which changes the expected reward at each step, and the total variation of expected reward is bounded by $V_T = (T/K)^{1/10}$. The reward signal in $[0, 1]$ given by environment at each step t is sampled from a Beta distribution.

We attack the popular strategy for stochastic non-stationary MAB problem, *Rexp3* as described in Section III-A. The

diminishing function parameters are $\alpha = \frac{1}{2}$ and $\epsilon = \frac{1}{5}$. Expectation is taken over 5 independent runs.

The scenario 1, 2 and 3 correspond to three attack scenarios: Section III-B, Section III-C and Section III-D, respectively. The batch sizes are set as $\Delta_T = \left\lceil 5(T/V_T)^{2/3} \right\rceil$ (scenario 1 2) and $\delta_T = \left\lceil 5(T/V_T)^{2/3} \right\rceil$ (scenario 3) for different horizons.

The attack result is shown in Figure 6a and 6b. Figure 6a shows that the number of target arm selection $\mathbb{E}^{\pi}\left[N_{\mathcal{T}}\left(a^{\dagger}\right)\right]$ increases significantly with attack. For example, as shown in Table II, the percentage of target arm selection increased from 0.483% to 89.5% in the scenario 1, from 0.485% to 97.9% in the scenario 2, and from 0.160% to 99.4% in the scenario 3. Note that, in the scenario 2, compared with the scenarios 1, the victim algorithm tends to select $a^{\dagger}$ more often since the attacker gives more reward to the target arm every time after a non-target arm is chosen. However, as shown in Figure 6b, Algorithm 3 also has disadvantage of having a larger attack cost. In particular, the expected attack cost for 10^7 steps are 6.14×10^5, 2.60×10^6 and 2.38×10^6 for the scenario 1, 2, and 3, respectively. In the scenario 3, even without attack, the victim algorithm tends to select $a^{\dagger}$ less often, since the batch size is larger, and the algorithm 'forgets' the history less often.

TABLE II: Expected target arm pulls for 10^7 steps

	With Attack		Without Attack	
Scenario	**Pulls**	**%**	**Pulls**	**%**
Scenario 1	8.95×10^6	89.5	4.83×10^4	0.483
Scenario 2	9.79×10^6	97.9	4.85×10^4	0.485
Scenario 3	9.94×10^6	99.4	1.60×10^4	0.160

The reason for more target arm selection while less attack cost in the scenario 3, compared with the scenario 2, will be discussed later as the influences of non-fixed batch size.

In general, our three attack strategies are successful: the expected attack cost, shown in Figure 6b, is sublinear in T, when the victim algorithm is forced to select one suboptimal arm mostly and the number of selection increases almost linear in T, shown in Figure 6a.

For the influences of the fixed batch size, Figure 6c shows the expected attack cost in the general attack scenario for different choice of parameter β for batch size $\Delta_T = \left\lceil 5(T/V_T)^{\beta} \right\rceil$ (shown in Figure 6d). The results verify Theorem III.1 that if the order of Δ_T is larger, the attack cost will be smaller, since the power of Δ_T-s in (3) is negative.

Note that the attack cost in the scenario 3 with non-fixed Δ_{T_j} could be larger or smaller than that in the scenario 2 with fixed Δ_T. Every time the victim algorithm restarts, it has a much higher probability to choose a non-target arm. As a result, the attacker will need to manipulate the reward to 0 more frequently. Meanwhile, the diminish function will be reset more frequently, introducing more cost when a successive target arm is pulled. Therefore, the attack cost and target arm selection depend on how frequently victim algorithm restarts.

The influences of the non-fixed batch size in the scenario 3 can be seen from Figure 6e and 6f. The batch size in the scenario 2 is fixed as $\Delta_T = \left\lceil 5(T/V_T)^{2/3} \right\rceil$. The scenario 3 is independently run for two rounds, with $\min \Delta_{T_j} = \Delta_T$ and $\max \Delta_{T_j} = \Delta_T$, respectively. As shown in Figure 6e, the target arm selection of the scenario 3 with $\max \Delta_{T_j} = \Delta_T$ is less than that of the scenario 2 and less than that of the scenario 3 with $\min \Delta_{T_j} = \Delta_T$. As shown in Figure 6f, the attack cost of the scenario 3 with $\max \Delta_{T_j} = \Delta_T$ is larger than that of the scenario 2 and larger than that of the scenario 3 with $\min \Delta_{T_j} = \Delta_T$. These results are consistent with the discussion above.

B. Defense

We now present results for the defense case. The same multiarmed bandit problem environment as in Section V-A is considered in these experiments. We attack our defense algorithm *RexpRb* with two different attack strategies:
1) *Unbounded attackers*: as Algorithm 2 and 3;
2) *Bounded attackers*: only perform Algorithm 2 and 3 attacks when the cumulative attack cost is less than the budget $\Phi(T)$.

We set $\Phi(T) = T^{0.861}$, $\Delta_T = \left\lceil (T/V_T)^{0.931} \right\rceil$, $V_T = (T/K)^{0.001}$ for bounded attack. Other parameters and parameters in unbounded attack are the same as in Section V-A.

The results with unbounded attackers are shown in Figure 7a, 7b and 7c. If the attacker can perform attacks with no budget on attack cost, the attacker will force both the *Rexp3* and *RexpRb* algorithms to pull the target arm most of the time, and the dynamic oracle regret is almost linear in T after attack. However, when attacking the *Rexp3*, the attack cost is sublinear in T, which is the same as the theoretical analysis in Section III. When attacking *RexpRb*, the attack cost is larger than that of *Rexp3*, and is almost linear in T. In other words, *RexpRb* makes the attacker more expensive to attack, and the attack can not be considered successful in terms of the attack cost. This fact shows that *RexpRb* is robust to attackers, even when the attacker has unbounded attack cost.

The results with bounded attackers are shown in Figure 7d, 7e and 7f. In this case, the attacker can only perform attacks when the cumulative attack cost is less than the budget $\Phi(T) = T^{0.861}$. As shown in Figure 7d, the dynamic regret of *Rexp3* increases greatly after attack, and is almost linear in T. However, as shown in Figure 7e, *RexpRb* has less regret, and is also sublinear in T. Without attack, the regret of *RexpRb* is same as that of *Rexp3*. The experiment results are consistent with the theoretical analysis in Section IV. In other words, *RexpRb* is robust to attackers, when the attack cost is bounded.

VI. Conclusion

In this paper, we have proposed three reward attacks scenarios and corresponding attack methods for the stochastic non-stationary multi-armed bandit problem. We have proved that our attack methods are successful. We have also proposed a defense algorithm *RexpRb* that is able to defend against any attacker whose attack cost is bounded. The experimental results verify our theoretical analysis. Moreover, we have derived a lower-bound of the expected attack cost when the

(a) $\mathbb{E}^{\pi}[N_{\mathcal{T}}(a^{\dagger})]$ (b) $\mathbb{E}^{\pi}[C_T]$ (c) $\mathbb{E}^{\pi}[C_T]$, fixed batch

(d) Δ_T, fixed batch (e) $\mathbb{E}^{\pi}[N_{\mathcal{T}}(a^{\dagger})]$, non-fixed batch (f) $\mathbb{E}^{\pi}[C_T]$, non-fixed batch

Fig. 6: Attack: (a-b) target arm pulls and attack cost; (c-d) fixed batch size influences, attack cost and batch size in scenario 1 for different β; (e-f) non-fixed batch size influences, target arm pulls and attack cost for different non-fixed batch

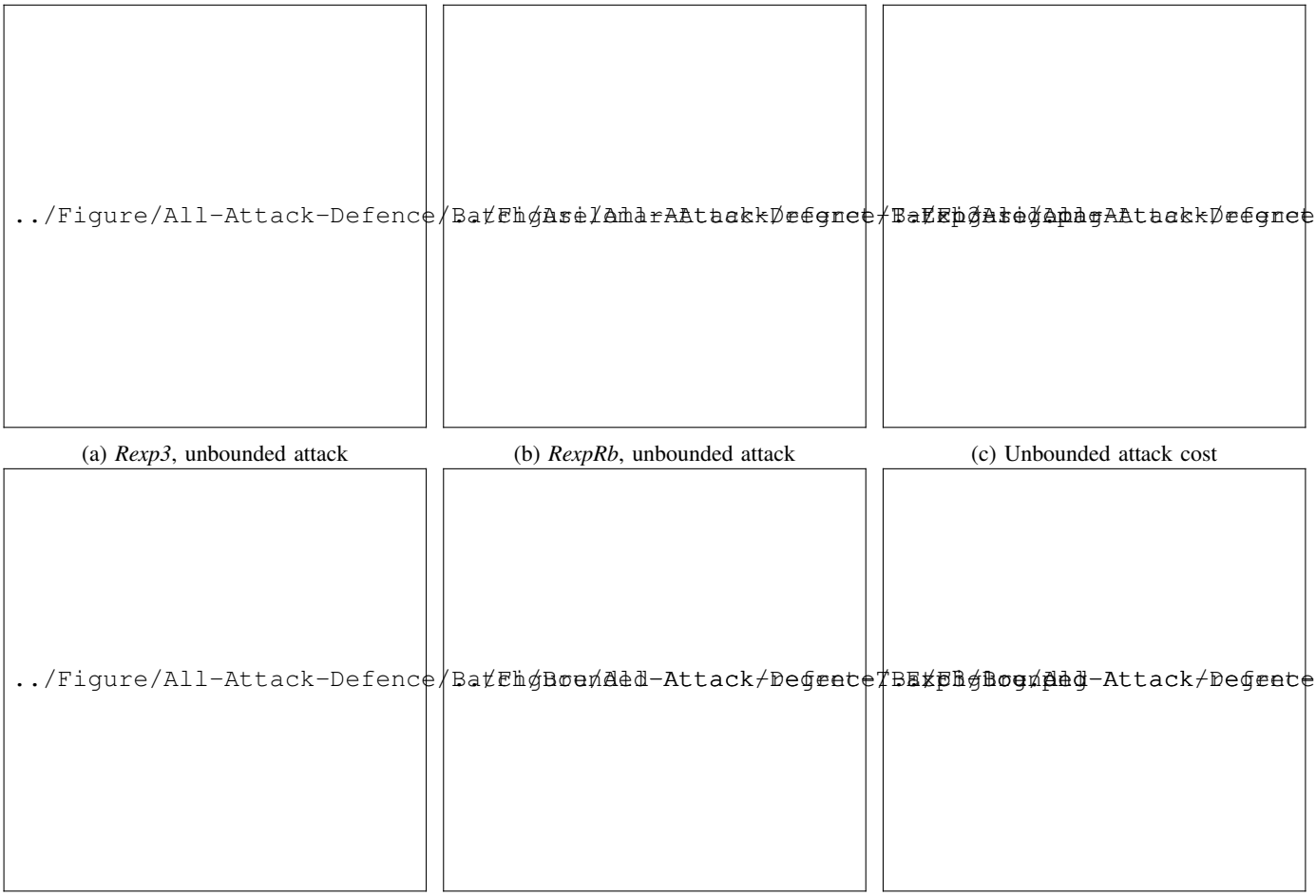
attack is successful in fixed batch scenarios. This lower bound shows that our attack methods are near optimal.

APPENDIX

A. Proof of Theorem III.1

Lower-bound the target arm selection $\mathbb{E}^{\pi}[N_{\mathcal{T}}(a^{\dagger})]$.

In batch $\mathcal{T}_j$, note that since the diminish function restarts at the beginning of each batch, the relative step $\tilde{t}$ in one batch which is used to calculate $\tilde{t}^{\alpha+\epsilon-1}$ will be: $\left\{\tilde{t} : 1, 2, \ldots, \Delta_{T_j}\right\}$. From the way batches are split, we have $\Delta_{T_j} \leq \Delta_T$. In batch



(a) *Rexp3*, unbounded attack (b) *RexpRb*, unbounded attack (c) Unbounded attack cost

(d) *Rexp3*, bounded attack (e) *RexpRb*, bounded attack (f) Bounded attack cost

Fig. 7: Defense: (a-c) dynamic regret of *Rexp3* and *RexpRb* facing unbounded attackers, alongside the corresponding attack cost; (d-f) dynamic regret of *Rexp3* and *RexpRb* facing bounded attackers, alongside the corresponding attack cost

$\mathcal{T}_j$, consider the dynamic oracle regret after attack:

$$\begin{aligned}
&\mathbb{E}^{\pi}\left[\sum_{t\in\mathcal{T}_j}(\tilde{\mu}_t^{*}-\tilde{\mu}_t^{\pi})\right]\\
&=\mathbb{E}^{\pi}\left[\sum_{t\in\mathcal{T}_j}\left(\max_{a_t}\left\{\mathbb{E}\left[\tilde{X}_t(a_t)\right]\right\}-\mathbb{E}\left[\tilde{X}_t(a_t^{\pi})\right]\right)\right]\\
&\overset{\text{(i)}}{=}\mathbb{E}^{\pi}\left[\sum_{t\in\mathcal{T}_j}\left(\mathbb{E}\left[\max\left\{\tilde{t}^{\alpha+\epsilon-1},X_t(a^{\dagger})\right\}\right]-\mathbb{E}\left[\tilde{X}_t\left(a_t^{\pi}\right)\right]\right)\right]\\
&\overset{\text{(ii)}}{=}\mathbb{E}^{\pi}\left[\sum_{t\in\mathcal{T}_j}\mathbf{1}\left[a_t^{\pi}\neq a^{\dagger}\right]\cdot\mathbb{E}\left[\max\left\{\tilde{t}^{\alpha+\epsilon-1},X_t(a^{\dagger})\right\}\right]\right]\\
&\geq\mathbb{E}^{\pi}\left[\sum_{t\in\mathcal{T}_j}\mathbf{1}\left[a_t^{\pi}\neq a^{\dagger}\right]\cdot\tilde{t}^{\alpha+\epsilon-1}\right].
\end{aligned}\tag{4}$$

(i) and (ii) are from manipulated rewards as in (2).

Note that $\epsilon<1-\alpha,\ \tilde{t}^{\alpha+\epsilon-1}$ is monotonically decreasing as $\tilde{t}$ increases, then we have:

$$\begin{aligned}
\sum_{t\in\mathcal{T}_j}\mathbf{1}\left[a_t^{\pi}\neq a^{\dagger}\right]\cdot\tilde{t}^{\alpha+\epsilon-1}&=\sum_{\tilde{t}=1}^{\Delta_{\mathcal{T}_j}}\mathbf{1}\left[a_t^{\pi}\neq a^{\dagger}\right]\cdot\tilde{t}^{\alpha+\epsilon-1}\\
&\overset{\text{(i)}}{\geq}\sum_{\tilde{t}=N_{\mathcal{T}_j}(a^{\dagger})+1}^{\Delta_{\mathcal{T}_j}}\tilde{t}^{\alpha+\epsilon-1}\overset{\text{(ii)}}{=}\sum_{\tilde{t}=1}^{\Delta_{\mathcal{T}_j}}\tilde{t}^{\alpha+\epsilon-1}-\sum_{\tilde{t}=1}^{N_{\mathcal{T}_j}(a^{\dagger})}\tilde{t}^{\alpha+\epsilon-1}.
\end{aligned}$$

Here (i) means that the inside of the summation is non-zero only for $\Delta_{T_j}-N_{\mathcal{T}_j}(a^{\dagger})$ steps in $\mathcal{T}_j$. The minimal value of the summation occurs if all the $a_t^{\pi}\neq a^{\dagger}$ concentrate in the tail of $\mathcal{T}_j$, since $\tilde{t}^{\alpha+\epsilon-1}$ is decreasing. (ii) means that the whole batch ($\mathcal{T}_j$) is split into tail ($a_t^{\pi}\neq a^{\dagger}$) and head ($a_t^{\pi}=a^{\dagger}$).

Since $\tilde{t}^{\alpha+\epsilon-1}$ is a monotonically decreasing non-negative

function, we have the integral and summation relations:

$$\sum_{\tilde{t}=1}^{\Delta_{T_j}} \tilde{t}^{\alpha+\epsilon-1} \geq \int_1^{\Delta_{T_j}} \tilde{t}^{\alpha+\epsilon-1} d\tilde{t} = \frac{{\Delta_{T_j}}^{\alpha+\epsilon} - 1}{\alpha+\epsilon},$$

$$\sum_{\tilde{t}=1}^{N_{\mathcal{T}_j}(a^\dagger)} \tilde{t}^{\alpha+\epsilon-1} \leq \int_0^{N_{\mathcal{T}_j}(a^\dagger)} \tilde{t}^{\alpha+\epsilon-1} = \frac{\left(N_{\mathcal{T}_j}(a^\dagger)\right)^{\alpha+\epsilon}}{\alpha+\epsilon}.$$

Therefore, we have:

$$\begin{aligned}
&\sum_{t\in\mathcal{T}_j} \mathbf{1}\left[a_t^\pi \neq a^\dagger\right] \cdot \tilde{t}^{\alpha+\epsilon-1} \\
&\geq \frac{1}{\alpha+\epsilon}\left({\Delta_{T_j}}^{\alpha+\epsilon} - N_{\mathcal{T}_j}(a^\dagger)^{\alpha+\epsilon}\right) - \frac{1}{\alpha+\epsilon} \\
&= \frac{{\Delta_{T_j}}^{\alpha+\epsilon}}{\alpha+\epsilon}\left(1 - \left(1 - \frac{\Delta_{T_j} - N_{\mathcal{T}_j}(a^\dagger)}{\Delta_{T_j}}\right)^{\alpha+\epsilon}\right) - \frac{1}{\alpha+\epsilon} \\
&\overset{\text{(i)}}{\geq} \frac{{\Delta_{T_j}}^{\alpha+\epsilon}}{\alpha+\epsilon}\frac{\Delta_{T_j} - N_{\mathcal{T}_j}(a^\dagger)}{\Delta_{T_j}}(\alpha+\epsilon) - \frac{1}{\alpha+\epsilon} \\
&= {\Delta_{T_j}}^{\alpha+\epsilon} - {\Delta_{T_j}}^{\alpha+\epsilon-1} N_{\mathcal{T}_j}(a^\dagger) - \frac{1}{\alpha+\epsilon}.
\end{aligned} \tag{5}$$

(i) comes from that $(1-x)^c \leq 1 - cx$ for $x, c \in (0,1)$.

Combine (4) and (5), we have

$$\begin{aligned}
&\mathbb{E}^\pi\left[\sum_{t\in\mathcal{T}_j}(\tilde{\mu}_t^* - \tilde{\mu}_t^\pi)\right] \\
&\geq {\Delta_{T_j}}^{\alpha+\epsilon} - {\Delta_{T_j}}^{\alpha+\epsilon-1}\mathbb{E}^\pi\left[N_{\mathcal{T}_j}(a^\dagger)\right] - \frac{1}{\alpha+\epsilon}.
\end{aligned}$$

Next we find the upper-bound of the regret. From the Section 5 of [**?**], we have:

$$\begin{aligned}
\mathbb{E}^\pi\left[\sum_{t\in\mathcal{T}_j}(\tilde{\mu}_t^* - \tilde{\mu}_t^\pi)\right] &= \underbrace{\sum_{t\in\mathcal{T}_j}\tilde{\mu}_t^* - \mathbb{E}\left[\max_{k\in\mathcal{K}}\left\{\sum_{t\in\mathcal{T}_j}\tilde{X}_t^k\right\}\right]}_{J_{1,j}} \\
&\quad + \underbrace{\mathbb{E}\left[\max_{k\in\mathcal{K}}\left\{\sum_{t\in\mathcal{T}_j}\tilde{X}_t^k\right\}\right] - \mathbb{E}^\pi\left[\sum_{t\in\mathcal{T}_j}\tilde{\mu}_t^\pi\right]}_{J_{2,j}},
\end{aligned}$$

where the first component, $J_{1,j}$, is the expected loss associated with using a single action over batch $\mathcal{T}_j$, under the manipulated environment. The second component, $J_{2,j}$, is the expected regret relative to the best static action in batch $\mathcal{T}_j$, which is also known as the static oracle regret.

Note that the first component $J_{1,j} = 0$, since after attack, only the target arm has non-zero rewards for all steps. Therefore, from the assumption that static oracle regret $R_T = O\left({\Delta_T}^\alpha\right)$ for some $\alpha \in [\frac{1}{2}, 1)$ in each batch, we have:

$$\mathbb{E}^\pi\left[\sum_{t\in\mathcal{T}_j}(\tilde{\mu}_t^* - \tilde{\mu}_t^\pi)\right] = J_{2,j} \leq M{\Delta_{T_j}}^\alpha, \tag{6}$$

where M is a constant, and Δ_{T_j} is the batch size of $\mathcal{T}_j$.

Then, combine the upper-bound and lower-bound, we have:

$$\mathbb{E}^\pi\left[N_{\mathcal{T}_j}(a^\dagger)\right] \geq \Delta_{T_j} - \frac{M}{{\Delta_{T_j}}^{\epsilon-1}} - \frac{1}{(\alpha+\epsilon){\Delta_{T_j}}^{\alpha+\epsilon-1}}.$$

With the bounded $N_{\mathcal{T}_j}(a^\dagger)$, next is to bound $N_{\mathcal{T}}(a^\dagger)$:

$$\begin{aligned}
\mathbb{E}^\pi\left[N_{\mathcal{T}}(a^\dagger)\right] &= \mathbb{E}^\pi\left[\sum_{j=1}^m N_{\mathcal{T}_j}(a^\dagger)\right] \\
&\geq \sum_{j=1}^m \Delta_{T_j} - \sum_{j=1}^m\left(\frac{M}{{\Delta_{T_j}}^{\epsilon-1}} + \frac{1}{(\alpha+\epsilon){\Delta_{T_j}}^{\alpha+\epsilon-1}}\right) \\
&= T - \sum_{j=1}^m \frac{1}{{\Delta_{T_j}}^{\alpha+\epsilon-1}}\left(M{\Delta_{T_j}}^\alpha + \frac{1}{\alpha+\epsilon}\right) \\
&\geq T - \frac{1}{{\Delta_T}^{\alpha+\epsilon-1}}\sum_{j=1}^m\left(M{\Delta_{T_j}}^\alpha + \frac{1}{\alpha+\epsilon}\right).
\end{aligned}$$

Since $m = \lceil \frac{T}{\Delta_T} \rceil \leq \frac{2T}{\Delta_T}$ with $T \geq \Delta_T \geq 1$, we have:

$$\sum_{j=1}^m M{\Delta_{T_j}}^\alpha \leq \sum_{j=1}^m M{\Delta_T}^\alpha \leq 2TM{\Delta_T}^{\alpha-1}. \tag{7}$$

Thus, with $m \leq \frac{2T}{\Delta_T}$, we have:

$$\mathbb{E}^\pi\left[N_{\mathcal{T}}(a^\dagger)\right] \geq T - \left(2TM{\Delta_T}^{-\epsilon} + \frac{2T{\Delta_T}^{-\alpha-\epsilon}}{\alpha+\epsilon}\right). \tag{8}$$

Upper-bound the attack cost $\mathbb{E}^\pi[C_T]$.

By the attack design, when $a_t \neq a^\dagger$, the attack cost of this step is $\left|\tilde{X}_t(a_t) - X_t(a_t)\right| \leq 1$. On the other hand, when $a_t = a^\dagger$, the attack cost of this step is $\left|\tilde{X}_t(a_t) - X_t(a_t)\right| \leq \tilde{t}^{\alpha+\epsilon-1}$. Therefore, the expected attack cost is:

$$\begin{aligned}
\mathbb{E}^\pi[C_T] &= \mathbb{E}^\pi\left[\sum_{t=1}^T\left|\tilde{X}_t(a_t) - X_t(a_t)\right|\right] \\
&\overset{\text{(i)}}{\leq} \underbrace{\mathbb{E}^\pi\left[\sum_{t=1}^T \mathbf{1}\left[a_t \neq a^\dagger\right]\cdot 1\right]}_{(a)} + \underbrace{\mathbb{E}^\pi\left[\sum_{t=1}^T \mathbf{1}\left[a_t = a^\dagger\right]\cdot\tilde{t}^{\alpha+\epsilon-1}\right]}_{(b)} \\
&\overset{\text{(ii)}}{\leq} \underbrace{T - \mathbb{E}^\pi\left[N_{\mathcal{T}}(a^\dagger)\right]}_{(a)} + \underbrace{\mathbb{E}^\pi\left[m\sum_{\tilde{t}=1}^{\Delta_T}\tilde{t}^{\alpha+\epsilon-1}\right]}_{(b)} \\
&\overset{\text{(iii)}}{\leq} 2TM{\Delta_T}^{-\epsilon} + \frac{2T{\Delta_T}^{-\alpha-\epsilon}}{\alpha+\epsilon} + m\sum_{\tilde{t}=1}^{\Delta_T}\tilde{t}^{\alpha+\epsilon-1}.
\end{aligned} \tag{9}$$

In (i), the entire horizon of the problem is split into two parts: $a_t \neq a^\dagger$ and $a_t = a^\dagger$, and the inequality follows from the attack design stated above.

Part (a) of (ii) comes from the fact that: $\sum_{t=1}^T \mathbf{1}\left[a_t \neq a^\dagger\right]$ means the number count of steps when $a_t \neq a^\dagger$ in the entire horizon of the problem, and $N_{\mathcal{T}}(a^\dagger)$ is the number count of steps when $a_t = a^\dagger$ in the entire horizon of the problem.

For the part (b) of (ii), because the m-th batch $\mathcal{T}_m$ might be shorter than Δ_T, and not all steps in a batch is $a_t = a^\dagger$, and $\tilde{t}^{\alpha+\epsilon-1} > 0$, then this part comes from the fact that:

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbf{1}\left[a_t = a^\dagger\right] \cdot \tilde{t}^{\alpha+\epsilon-1} \\
&= (m-1)\sum_{\tilde{t}=1}^{\Delta_T} \mathbf{1}\left[a^\dagger\right] \cdot \tilde{t}^{\alpha+\epsilon-1} + \sum_{\tilde{t}=1}^{T-(m-1)\Delta_T} \mathbf{1}\left[a^\dagger\right] \cdot \tilde{t}^{\alpha+\epsilon-1} \\
&\leq (m-1)\sum_{\tilde{t}=1}^{\Delta_T} \tilde{t}^{\alpha+\epsilon-1} + \sum_{\tilde{t}=1}^{T-(m-1)\Delta_T} \tilde{t}^{\alpha+\epsilon-1} \leq m\sum_{\tilde{t}=1}^{\Delta_T} \tilde{t}^{\alpha+\epsilon-1}.
\end{aligned}$$

(iii) comes from (8).

For the decreasing non-negative function $\tilde{t}^{\alpha+\epsilon-1}$, we have:

$$\sum_{\tilde{t}=1}^{\Delta_T} \tilde{t}^{\alpha+\epsilon-1} \leq \int_0^{\Delta_T} \tilde{t}^{\alpha+\epsilon-1} d\tilde{t} = \frac{{\Delta_T}^{\alpha+\epsilon}}{\alpha+\epsilon}.$$

Thus, with $m \leq 2T/\Delta_T$, the expected attack cost can be upper bounded as:

$$\mathbb{E}^\pi\left[C_T\right] \leq 2TM{\Delta_T}^{-\epsilon} + \frac{2T{\Delta_T}^{-\alpha-\epsilon}}{\alpha+\epsilon} + \frac{2T{\Delta_T}^{\alpha+\epsilon-1}}{\alpha+\epsilon}. \quad (10)$$

B. Proof of Theorem III.2

Lower-bound the target arm selection $\mathbb{E}^\pi[N_{\mathcal{T}}(a^\dagger)]$.

Similar as (4) in Section A but replacing $\mathcal{T}_j$ by $\mathcal{T}$, the dynamic regret after attack over entire horizon $\mathcal{T}$ is:

$$\mathbb{E}^\pi\left[\sum_{t\in\mathcal{T}} (\tilde{\mu}_t^* - \tilde{\mu}_t^\pi)\right] \geq \mathbb{E}^\pi\left[\sum_{t\in\mathcal{T}} \mathbf{1}\left[a_t^\pi \neq a^\dagger\right] \cdot \tilde{t}^{\alpha+\epsilon-1}\right].$$

Note that $\tilde{t}$ will be reset to 1 once any non-target arm is pulled, according to Algorithm 3. As a result, the $\tilde{t}^{\alpha+\epsilon-1}$ term will always be 1, controlled by the indicator function $\mathbf{1}\left[a_t^\pi \neq a^\dagger\right]$. Then, the dynamic regret is lower-bounded as:

$$\mathbb{E}^\pi\left[\sum_{t\in\mathcal{T}} (\tilde{\mu}_t^* - \tilde{\mu}_t^\pi)\right] \geq \mathbb{E}^\pi\left[\sum_{t\in\mathcal{T}} \mathbf{1}\left[a_t^\pi \neq a^\dagger\right]\right] = \mathbb{E}^\pi\left[N_{\mathcal{T}}(a')\right],$$

where a' represents any non-target arm, and $\mathbb{E}^\pi\left[N_{\mathcal{T}}(a')\right]$ is the expected number of non-target arm selection.

Then, we find the upper-bound of the dynamic regret after attack from the periodical behavior of the victim algorithm.

Similar to (6), in victim algorithm's view, we have $\mathbb{E}^\pi\left[\sum_{t\in\mathcal{T}_j} (\tilde{\mu}_t^* - \tilde{\mu}_t^\pi)\right] \leq M{\Delta_{T_j}}^\alpha$.

Note that in this attack method, when the batch starts is not necessarily the same as when the $\tilde{t}$ is reset to 1. There may be multiple times of $\tilde{t}$ reset in one batch, but the victim algorithm won't know that. Also, the batch information is only available to the victim algorithm.

Then by (7), we have the upper-bound for the dynamic regret after attack over the entire horizon $\mathcal{T}$ as:

$$\begin{aligned}
\mathbb{E}^\pi\left[\sum_{t\in\mathcal{T}} (\tilde{\mu}_t^* - \tilde{\mu}_t^\pi)\right] &= \mathbb{E}^\pi\left[\sum_{j=1}^{m}\sum_{t\in\mathcal{T}_j} (\tilde{\mu}_t^* - \tilde{\mu}_t^\pi)\right] \\
&\leq \sum_{j=1}^{m} M{\Delta_{T_j}}^\alpha \leq 2TM{\Delta_T}^{\alpha-1}.
\end{aligned}$$

Combining the lower-bound and upper-bound, we have:

$$\mathbb{E}^\pi\left[N_{\mathcal{T}}(a')\right] \leq 2MT{\Delta_T}^{\alpha-1}. \quad (11)$$

Thus, the expected number of target arm selection has:

$$\mathbb{E}^\pi\left[N_{\mathcal{T}}(a^\dagger)\right] \geq T - 2MT{\Delta_T}^{\alpha-1}. \quad (12)$$

Upper-bound the attack cost $\mathbb{E}^\pi[C_T]$.

Similar as (9) in Section A, by the attack design, the expected attack cost is:

$$\begin{aligned}
&\mathbb{E}^\pi\left[C_T\right] = \mathbb{E}^\pi\left[\sum_{t=1}^{T}\left|\tilde{X}_t\left(a_t\right) - X_t\left(a_t\right)\right|\right] \\
&\leq \underbrace{\mathbb{E}^\pi\left[\sum_{t=1}^{T} \mathbf{1}\left[a_t \neq a^\dagger\right] \cdot 1\right]}_{(a)} + \underbrace{\mathbb{E}^\pi\left[\sum_{t=1}^{T} \mathbf{1}\left[a_t = a^\dagger\right] \cdot \tilde{t}^{\alpha+\epsilon-1}\right]}_{(b)} \\
&\overset{\text{(i)}}{\leq} \underbrace{T - \mathbb{E}^\pi\left[N_{\mathcal{T}}\left(a^\dagger\right)\right]}_{(a)} + \underbrace{\mathbb{E}^\pi\left[N_{\mathcal{T}}\left(a'\right)\right] \cdot \sum_{\tilde{t}=1}^{n} \tilde{t}^{\alpha+\epsilon-1}}_{(b)} \\
&\overset{\text{(ii)}}{\leq} 2MT{\Delta_T}^{\alpha-1} + \mathbb{E}^\pi\left[N_{\mathcal{T}}\left(a'\right)\right] \cdot \sum_{\tilde{t}=1}^{n} \tilde{t}^{\alpha+\epsilon-1}.
\end{aligned}$$

Part (a) of (i) comes from the fact that: $\sum_{t=1}^{T} \mathbf{1}\left[a_t \neq a^\dagger\right]$ means the number count of steps when $a_t \neq a^\dagger$ in the entire horizon of the problem, and $N_{\mathcal{T}}\left(a^\dagger\right)$ is the number count of steps when $a_t = a^\dagger$ in the entire horizon of the problem.

For part (b) of (i), since $\tilde{t}^{\alpha+\epsilon-1}$ is decreasing, and $\tilde{t}$ is reset to 1 whenever a non-target arm a' is pulled, and $\tilde{t}^{\alpha+\epsilon-1}$ decreases from 1 for all the successive target arms $a^\dagger$. Therefore, the upper-bound for (b) of (i) can be found by evenly splitting the target arm pulls steps using non-target arm pulls steps. In other words, one non-target arm pull is followed by multiple target arm pulls. Here we assume the target arm selections $\mathbb{E}^\pi\left[N_{\mathcal{T}}\left(a^\dagger\right)\right]$ is greater than non-target arm selections $\mathbb{E}^\pi\left[N_{\mathcal{T}}\left(a'\right)\right]$, otherwise, the attack is not successful. If the target arm pulls steps are not grouped in this way, the summation will be smaller since $\tilde{t}^{\alpha+\epsilon-1}$ decreases.

Thus, the target arm pulls steps in horizon are split into $\mathbb{E}^\pi\left[N_{\mathcal{T}}\left(a'\right)\right]$ groups, within each group there are $n = \frac{\mathbb{E}^\pi\left[N_{\mathcal{T}}\left(a^\dagger\right)\right]}{\mathbb{E}^\pi\left[N_{\mathcal{T}}\left(a'\right)\right]}$ target arm pulls. Note that even though $\mathbb{E}^\pi\left[N_{\mathcal{T}}\left(a^\dagger\right)\right]$ may not be divisible by $\mathbb{E}^\pi\left[N_{\mathcal{T}}\left(a'\right)\right]$, the n can still be used here for the upper-bound. Because the remainders will be placed to each group one by one and $\tilde{n}^{\alpha+\epsilon-1} \geq \lceil\tilde{n}\rceil^{\alpha+\epsilon-1}$. In this case, $\sum_{\tilde{t}=1}^{n}$ means that the summation for $\tilde{t} = 1, 2, \ldots, \lfloor n\rfloor, n$.

(ii) comes from (12).

For the decreasing non-negative function $\tilde{t}^{\alpha+\epsilon-1}$, we have:

$$\sum_{\tilde{t}=1}^{n} \tilde{t}^{\alpha+\epsilon-1} \leq \int_0^{n} \tilde{t}^{\alpha+\epsilon-1} d\tilde{t} = \frac{1}{\alpha+\epsilon}\left(\frac{\mathbb{E}^\pi\left[N_{\mathcal{T}}\left(a^\dagger\right)\right]}{\mathbb{E}^\pi\left[N_{\mathcal{T}}\left(a'\right)\right]}\right)^{\alpha+\epsilon}.$$

With (11) and $\mathbb{E}^\pi\left[N_{\mathcal{T}}\left(a^\dagger\right)\right] \leq T$, it follows that:

$$\mathbb{E}^\pi\left[N_{\mathcal{T}}\left(a'\right)\right] \cdot \sum_{\tilde{t}=1}^{n} \tilde{t}^{\alpha+\epsilon-1} \leq \frac{(2M)^{1-\alpha-\epsilon}}{\alpha+\epsilon} T{\Delta_T}^{2\alpha-\alpha^2-1+\epsilon-\alpha\epsilon}.$$

Therefore, the expected attack cost is upper-bounded as:

$$\mathbb{E}^{\pi}\left[C_T\right] \leq 2MT{\Delta_T}^{\alpha-1} + \frac{(2M)^{1-\alpha-\epsilon}}{\alpha+\epsilon} T{\Delta_T}^{2\alpha-\alpha^2-1+\epsilon-\alpha\epsilon}.$$

C. Proof of Theorem III.3

Lower-bound the target arm selection $\mathbb{E}^{\pi}[N_{\mathcal{T}}(a^{\dagger})]$.

Similar as the proof of Theorem III.2 in Appendix B, the dynamic regret after attack is lower-bounded as: $\mathbb{E}^{\pi}\left[\sum_{t\in\mathcal{T}}(\tilde{\mu}_t^* - \tilde{\mu}_t^{\pi})\right] \geq \mathbb{E}^{\pi}\left[N_{\mathcal{T}}(a')\right]$.

The upper-bound for the dynamic regret after attack is: $\mathbb{E}^{\pi}\left[\sum_{t\in\mathcal{T}}(\tilde{\mu}_t^* - \tilde{\mu}_t^{\pi})\right] \leq \sum_{j=1}^{m} M{\Delta_{T_j}}^{\alpha}$.

Since $f(x) = x^{\alpha}$ is concave for $\alpha \in [\frac{1}{2}, 1)$, by Jensen's Inequality, we have: $M\sum_{j=1}^{m}{\Delta_{T_j}}^{\alpha} \leq Mm\left(\frac{\sum_{j=1}^{m}\Delta_{T_j}}{m}\right)^{\alpha} = Mm^{1-\alpha}T^{\alpha}$.

Since the number of batches $m \leq \frac{T}{\min_j\{\Delta_{T_j}\}}$, denote $\delta_T = \min_j\{\Delta_{T_j}\}$ as the minimal batch size, we have:

$$\mathbb{E}^{\pi}\left[\sum_{t\in\mathcal{T}}(\tilde{\mu}_t^* - \tilde{\mu}_t^{\pi})\right] \leq M\left(\frac{T}{\delta_T}\right)^{1-\alpha} T^{\alpha} = MT{\delta_T}^{\alpha-1}.$$

Then, combine the lower-bound and upper-bound, we have:

$$\mathbb{E}^{\pi}\left[N_{\mathcal{T}}(a')\right] \leq MT{\delta_T}^{\alpha-1}. \tag{13}$$

Thus, for the expected number of target arm selection, we have:

$$\mathbb{E}^{\pi}\left[N_{\mathcal{T}}(a^{\dagger})\right] \geq T - MT{\delta_T}^{\alpha-1}. \tag{14}$$

Upper-bound the attack cost $\mathbb{E}^{\pi}[C_T]$.

Similar as the proof of Theorem III.2 in Appendix B, the attack cost is upper-bounded as:

$$\begin{aligned}\mathbb{E}^{\pi}\left[C_T\right] &\leq \underbrace{T - \mathbb{E}^{\pi}\left[N_{\mathcal{T}}\left(a^{\dagger}\right)\right]}_{(a)} + \underbrace{\mathbb{E}^{\pi}\left[N_{\mathcal{T}}\left(a'\right)\right]\sum_{\tilde{t}=1}^{n}\tilde{t}^{\alpha+\epsilon-1}}_{(b)} \\ &\leq \underbrace{MT{\delta_T}^{\alpha-1}}_{(a)} + \underbrace{\frac{M^{1-\alpha-\epsilon}}{\alpha+\epsilon}T{\delta_T}^{2\alpha-\alpha^2-1+\epsilon-\alpha\epsilon}}_{(b)}.\end{aligned}$$

Part (a) comes from (14).

Part (b) comes from the similar proof procedure that:

$$\begin{aligned}\mathbb{E}^{\pi}\left[N_{\mathcal{T}}\left(a'\right)\right]\sum_{\tilde{t}=1}^{n}\tilde{t}^{\alpha+\epsilon-1} &\leq \frac{\mathbb{E}^{\pi}\left[N_{\mathcal{T}}\left(a'\right)\right]^{1-\alpha-\epsilon}\mathbb{E}^{\pi}\left[N_{\mathcal{T}}\left(a^{\dagger}\right)\right]^{\alpha+\epsilon}}{\alpha+\epsilon} \\ &\overset{\text{(i)}}{\leq} \frac{M^{1-\alpha-\epsilon}}{\alpha+\epsilon}T{\delta_T}^{2\alpha-\alpha^2-1+\epsilon-\alpha\epsilon}.\end{aligned}$$

(i) comes from (13) and $\mathbb{E}^{\pi}\left[N_{\mathcal{T}}\left(a^{\dagger}\right)\right] \leq T$.

D. Proof of Theorem III.4

To show the lower-bound of attack cost for a victim-agnostic attack algorithm, it is sufficient to find one specific victim algorithm that has the regret as in Theorem III.4 and follows the batch strategy, under one bandit environment.

Algorithm 5 *Exp3*

1: **Parameters:** $w_1 = (1, \ldots, 1)$, total horizon T, and a constant learning rate η.
2: **for** $t = 1, 2, \ldots, T$ **do**
3: Define $\pi_t = \frac{w_t}{\|w_t\|_1}$
4: Draw $a_t \sim \pi_t$, and observe loss $\ell_t = \mathcal{L}_t\left(a_t\right)$
5: Update $w_{t+1,a}$ for each arm $a = 1, 2, \ldots, K$ as:

$$w_{t+1,a} = \begin{cases} w_{t,a}\exp\left(-\eta\frac{\ell_t}{\pi_{t,a}}\right) & a = a_t \\ w_{t,a} & a \neq a_t \end{cases}$$

6: **end for**

First we construct the special bandit environment, in which there are two arms, $a^{\dagger}$ and a', and the reward of $a^{\dagger}$ is always 0.5, while the reward of a' is always 1:

$$X_t(a) = \begin{cases} 0.5 & a = a^{\dagger} \\ 1 & a = a' \end{cases} \quad \forall t \in \mathcal{T}.$$

The reward signal can also be expressed as loss signal $\mathcal{L}_t(a) = 1 - X_t(a)$. Note that this is a special case for non-stationary environment, and the original optimal arm is a'.

For the victim algorithm, we consider the case when *Exp3* algorithm restarts every Δ_T steps. The *Exp3* algorithm is defined with loss signal $\mathcal{L}_t(a_t) \in [0, 1]$ in Algorithm 5. The static regret of *Exp3* is $O({\Delta_T}^{\alpha})$, $\alpha = \frac{1}{2}$. There are m batches in total, i.e., $T = m\Delta_T$. Each batch has the same fixed size Δ_T, also for the last batch.

By Lemma 5.1 in [**?**], within batch $\mathcal{T}_j$, the expected number of rounds where arm a is selected, $\mathbb{E}^{\pi}\left[N_{\mathcal{T}_j}(a)\right]$, satisfies:

$$\mathbb{E}^{\pi}\left[N_{\mathcal{T}_j}(a)\right] \geq \Delta_T\pi_1(a) - \eta\Delta_T\sum_{t'=1}^{\Delta_T}\mathbb{E}\left[\pi_{t'}(a)\mathcal{L}_{t'}(a)\right], \tag{15}$$

where $\pi_{t'}(a)$ is the probability of selecting a at relative time step t' within this batch, and η is the learning rate. Note that:

- $\pi_1(a)$ is the initial probability at the beginning of each batch. Since the *Exp3* algorithm restarts periodically, $\pi_1(a) = \frac{1}{K}$ $\forall a \in \mathcal{K}$ is the same for all batches.
- $\eta = \beta{\Delta_T}^{-\alpha}$ is chosen as the learning rate for *Exp3* algorithm within each batch, where β is a constant.

Thus, the expected total number of arm selection $\mathbb{E}^{\pi}\left[N_{\mathcal{T}}(a)\right]$ in the entire horizon T satisfies:

$$\begin{aligned}\mathbb{E}^{\pi}\left[N_{\mathcal{T}}(a)\right] &= \sum_{j=1}^{m}\mathbb{E}^{\pi}\left[N_{\mathcal{T}_j}(a)\right] \\ &\overset{\text{(i)}}{\geq} \sum_{j=1}^{m}\Delta_T\pi_1(a) - \sum_{j=1}^{m}\eta\Delta_T\sum_{t'=1}^{\Delta_T}\mathbb{E}\left[\pi_{t'}(a)\mathcal{L}_{t'}(a)\right] \\ &\overset{\text{(ii)}}{=} \pi_1(a)T - \beta\left(\frac{T}{m}\right)^{1-\alpha}\sum_{j=1}^{m}\sum_{t'=1}^{\Delta_T}\mathbb{E}\left[\pi_{t'}(a)\mathcal{L}_{t'}(a)\right] \\ &= \pi_1(a)T - \beta\left(\frac{T}{m}\right)^{1-\alpha}\sum_{t=1}^{T}\mathbb{E}\left[\pi_t(a)\mathcal{L}_t(a)\right].\end{aligned} \tag{16}$$

(i) comes from (15).

(ii) comes from the fact that $\pi_1(a)$ is the initial probability at the beginning of each batch and the definition of η.

Now we introduce the attack. The attacker's target arm is $a^\dagger$, and the manipulated rewards are $\tilde{X}_t(a_t)$. Let the loss signal manipulated by the attacker be $\tilde{\mathcal{L}}_t = 1 - \tilde{X}_t$, and the expected total number of arm selection under attack be $\mathbb{E}^\pi\left[\tilde{N}_\mathcal{T}(a)\right]$. Suppose the attack is successful, i.e., $\mathbb{E}^\pi\left[\tilde{N}_\mathcal{T}(a^\dagger)\right] = T - o(T)$. Then, we must have $\mathbb{E}^\pi\left[\tilde{N}_\mathcal{T}(a')\right] = o(T)$, i.e., for all constant $C > 0$, $\lim_{T\to\infty} \frac{\mathbb{E}^\pi\left[\tilde{N}_\mathcal{T}(a')\right]}{T} \leq C$.

Then, by the lower-bound in (16), under attack, we have that as $T \to \infty$, for arm a': $CT \geq \tilde{\pi}_1(a')T - \beta\left(\frac{T}{m}\right)^{1-\alpha}\sum_{t=1}^T \mathbb{E}\left[\tilde{\pi}_t(a')\tilde{\mathcal{L}}_t(a')\right]$. It follows that:

$$\sum_{t=1}^T \mathbb{E}\left[\tilde{\pi}_t(a')\tilde{\mathcal{L}}_t(a')\right] \geq \frac{\tilde{\pi}_1(a')T}{\beta\left(\frac{T}{m}\right)^{1-\alpha}} - \frac{CT}{\beta\left(\frac{T}{m}\right)^{1-\alpha}} = \left[\frac{\tilde{\pi}_1(a')}{\beta} - \frac{C}{\beta}\right] T{\Delta_T}^{\alpha-1}.$$

Note that the positive constant C can be arbitrarily small, we can treat it as $C < \frac{1}{2}$. As a result, the constant $\frac{\tilde{\pi}_1(a')}{\beta} - \frac{C}{\beta} > 0$, since $\tilde{\pi}_1(a') = \frac{1}{2}$ in our bandit problem.

Now we prove the lower-bound of the expected attack cost $\mathbb{E}^\pi[C_T]$, note that $\mathcal{L}_t(a') = 0$:

$$\begin{aligned}
\mathbb{E}^\pi[C_T] &= \mathbb{E}\left[\sum_{t=1}^T \sum_a \tilde{\pi}_t(a)\left|\tilde{\mathcal{L}}_t(a) - \mathcal{L}_t(a)\right|\right] \\
&\geq \mathbb{E}\left[\sum_{t=1}^T \tilde{\pi}_t(a')\left|\tilde{\mathcal{L}}_t(a') - \mathcal{L}_t(a')\right|\right] \\
&= \mathbb{E}\left[\sum_{t=1}^T \tilde{\pi}_t(a')\tilde{\mathcal{L}}_t(a')\right] \geq \left[\frac{\tilde{\pi}_1(a')}{\beta} - \frac{C}{\beta}\right] T{\Delta_T}^{\alpha-1}.
\end{aligned}$$

Therefore, the expected attack cost is:

$$\mathbb{E}^\pi[C_T] = \Omega\left(T{\Delta_T}^{\alpha-1}\right).$$

E. Proof of Theorem IV.1

In batch $\mathcal{T}_j$, decompose the dynamic oracle regret with original reward as:

$$\begin{aligned}
\mathbb{E}^\pi\left[\sum_{t\in\mathcal{T}_j}\left(\mu_t^* - \mu_t^{\tilde{\pi}}\right)\right] &= \underbrace{\sum_{t\in\mathcal{T}_j}\mu_t^* - \mathbb{E}\left[\max_{k\in\mathcal{K}}\left\{\sum_{t\in\mathcal{T}_j} X_t^k\right\}\right]}_{J_{1,j}} \\
&+ \underbrace{\mathbb{E}\left[\max_{k\in\mathcal{K}}\left\{\sum_{t\in\mathcal{T}_j} X_t^k\right\}\right] - \mathbb{E}^\pi\left[\sum_{t\in\mathcal{T}_j}\tilde{X}_t^{\tilde{\pi}}\right]}_{J_{2,j}} \\
&+ \underbrace{\mathbb{E}^\pi\left[\sum_{t\in\mathcal{T}_j}\tilde{X}_t^{\tilde{\pi}}\right] - \mathbb{E}^\pi\left[\sum_{t\in\mathcal{T}_j}\mu_t^{\tilde{\pi}}\right]}_{J_{3,j}},
\end{aligned}$$

where $J_{1,j}$ is the expected loss associated with using a single action over batch $\mathcal{T}_j$, under the original environment. $J_{2,j}$ is the static oracle regret [4] with the manipulated rewards and against the original rewards over batch $\mathcal{T}_j$, as in Table I and in the existing work [**?**]. $J_{3,j}$ represents the extent of manipulations on rewards made by the attacker.

From the Equation (6) in Section 5 of [**?**], we have that $J_{1,j} \leq 2V_j\Delta_{T_j}$. From the Theorem 3 in Section 5 of [**?**], we have that $J_{2,j} \leq D_1\sqrt{\Delta_{T_j}} + D_2\Phi(T)\log(\Delta_{T_j})$, where D_1 and D_2 are constants.

For the third component $J_{3,j}$, we have:

$$\begin{aligned}
J_{3,j} &= \mathbb{E}^\pi\left[\sum_{t\in\mathcal{T}_j}\tilde{X}_t^{\tilde{\pi}}\right] - \mathbb{E}^\pi\left[\sum_{t\in\mathcal{T}_j}\mu_t^{\tilde{\pi}}\right] \\
&= \mathbb{E}^\pi\left[\sum_{t\in\mathcal{T}_j}\left(\tilde{X}_t^{\tilde{\pi}} - X_t^{\tilde{\pi}}\right)\right] \leq \mathbb{E}^\pi\left[\sum_{t\in\mathcal{T}_j}\left|\tilde{X}_t^{\tilde{\pi}} - X_t^{\tilde{\pi}}\right|\right] = C_{T_j},
\end{aligned}$$

where C_{T_j} is the attack cost in batch $\mathcal{T}_j$.

For the dynamic oracle regret with original reward in the entire horizon T, we have:

$$\begin{aligned}
R^{\tilde{\pi}}(\mathcal{V}, T) &= \mathbb{E}^\pi\left[\sum_{t\in\mathcal{T}}\left(\mu_t^* - \mu_t^{\tilde{\pi}}\right)\right] = \sum_{j=1}^m \mathbb{E}^\pi\left[\sum_{t\in\mathcal{T}_j}\left(\mu_t^* - \mu_t^{\tilde{\pi}}\right)\right] \\
&\leq \sum_{j=1}^m\left(2V_j\Delta_{T_j} + D_1\sqrt{\Delta_{T_j}} + D_2\Phi(T)\log(\Delta_{T_j}) + C_{T_j}\right) \\
&\overset{\text{(i)}}{\leq} 2V_T\Delta_T + D_1\sum_{j=1}^m\sqrt{\Delta_{T_j}} + D_2\Phi(T)\sum_{j=1}^m\log(\Delta_{T_j}) + \Phi(T) \\
&\overset{\text{(ii)}}{\leq} 2V_T\Delta_T + D_1' T\Delta_T^{-\frac{1}{2}} + D_2'\Phi(T)\frac{T}{\Delta_T}\log(\Delta_T) + \Phi(T),
\end{aligned}$$

where D_1' and D_2' are constants.

(i) comes from the Equation (4) in Section 5 of [**?**], and the facts that $\Delta_{T_j} \leq \Delta_T$ and:

$$\begin{aligned}
\sum_{j=1}^m C_{T_j} &= \mathbb{E}^\pi\left[\sum_{t=1}^T\left|\tilde{X}_t(a_t) - X_t(a_t)\right|\right] \\
&= \mathbb{E}^\pi[C_T] \leq \Phi(T)
\end{aligned}$$

Considering the fact that $m \leq 2T/\Delta_T$, (ii) is from that:

$$\sum_{j=1}^m\sqrt{\Delta_{T_j}} \leq \sum_{j=1}^m\sqrt{\Delta_T} = m\sqrt{\Delta_T} \leq \frac{2T}{\Delta_T}\sqrt{\Delta_T} = 2T\Delta_T^{-\frac{1}{2}}$$

$$\sum_{j=1}^m\log(\Delta_{T_j}) \leq \sum_{j=1}^m\log(\Delta_T) = m\log(\Delta_T) \leq \frac{2T}{\Delta_T}\log(\Delta_T).$$

Therefore, the dynamic oracle regret with original reward:

$$\begin{aligned}
&R^{\tilde{\pi}}(\mathcal{V}, T) \\
&\quad = O\left(V_T\Delta_T + T\Delta_T^{-\frac{1}{2}} + \Phi(T)\frac{T}{\Delta_T}\log(\Delta_T) + \Phi(T)\right).
\end{aligned}$$

This completes the proof.