
Adversarial Attacks on Stochastic Bandits

Kwang-Sung Jun
Boston University
kwangsung.jun@gmail.com

Lihong Li
Google Brain
lihong@google.com

Yuzhe Ma
UW-Madison
ma234@wisc.edu

Xiaojin Zhu
UW-Madison
jerryzhu@cs.wisc.edu

Abstract

We study adversarial attacks that manipulate the reward signals to control the actions chosen by a stochastic multi-armed bandit algorithm. We propose the first attack against two popular bandit algorithms: ϵ -greedy and UCB, *without* knowledge of the mean rewards. The attacker is able to spend only logarithmic effort, multiplied by a problem-specific parameter that becomes smaller as the bandit problem gets easier to attack. The result means the attacker can easily hijack the behavior of the bandit algorithm to promote or obstruct certain actions, say, a particular medical treatment. As bandits are seeing increasingly wide use in practice, our study exposes a significant security threat.

1 Introduction

Designing trustworthy machine learning systems requires understanding how they may be attacked. There has been a surge of interest on adversarial attacks against supervised learning [12, 15]. In contrast, little is known on adversarial attacks against stochastic multi-armed bandits (MABs), a form of online learning with limited feedback. This is potentially hazardous since stochastic MABs are widely used in the industry to recommend news articles [18], display advertisements [9], improve search results [17], allocate medical treatment [16], and promote users’ well-being [13], among many others. Indeed, as we show, an adversarial attacker can modify the reward signal to manipulate the MAB for nefarious goals.

Our main contribution is an analysis on reward-manipulation attacks. We distinguish three agents in this setting: “the world,” “Bob” the bandit algorithm, and “Alice” the attacker. As in standard stochastic bandit problems, the world consists of K arms with sub-Gaussian rewards centered at μ_1, \dots, μ_K . Note that we do *not* assume $\{\mu_i\}$ are sorted. Neither Bob nor Alice knows $\{\mu_i\}$. Bob pulls selected arms in rounds and attempts to minimize his regret. When Bob pulls arm $I_t \in [K]$ in round t , the world generates a random reward r_t^0 drawn from a sub-Gaussian distribution with expectation μ_{I_t} . However, Alice sits in-between the world and Bob and manipulates the reward into $r_t = r_t^0 - \alpha_t$. We call $\alpha_t \in \mathbb{R}$ the attack. If Alice decides not to attack in this round, she simply lets $\alpha_t = 0$. Bob then receives r_t , without knowing the presence of Alice. Without loss of generality, assume arm K is a suboptimal “attack target” arm: $\mu_K < \max_{i=1\dots K} \mu_i$. Alice’s goal is to manipulate Bob into pulling arm K very often while making small attacks. Specifically, we show Alice can force Bob to pull the target arm $T - o(T)$ number of times with a cumulative attack cost of $\sum_{t=1}^T |\alpha_t| = O(\log(T))$.

The assumption that Alice does not know $\{\mu_i\}$ is significant because otherwise Alice can perform the attack trivially. To see this, with the knowledge of $\{\mu_i\}$ Alice would be able to compute the truncated reward gap $\Delta_i^\epsilon = \max\{\mu_i - \mu_K + \epsilon, 0\} \geq 0$ for all non-target arms $i \neq K$ for some small parameter

$\epsilon > 0$. Alice can perform the following *oracle attack*: in any round where a non-target arm $I_t \neq K$ is pulled, attack with $\alpha_t = \Delta_{I_t}^\epsilon$. This oracle attack transforms the original bandit problem into one where all non-target arms have expected reward less than μ_K . It is well-known that if Bob runs a sublinear-regret algorithm (e.g., UCB [6, 8]), almost all arm pulls will concentrate on the now-best target arm K in the transformed bandit problem. Furthermore, Alice’s cumulative attack cost will be sublinear in time, because the total number of non-target arm pulls is sublinear in the transformed problem. In practice, however, it is almost never the case that Alice knows μ_1, \dots, μ_K and hence the Δ_i^ϵ ’s. Thus the oracle attack is impractical. Our focus in this paper is to design an attack that nearly matches the oracle attack, but for Alice who does not know $\{\mu_i\}$. We do so for two popular bandit algorithms, ϵ -greedy [7] and UCB [8].

What damage can Alice do in practice? She can largely control the arms pulled by Bob. She can also control which arm appears to Bob as the best arm at the end. As an example, consider the news-delivering contextual bandit problem [18]. The arms are available news articles, and Bob selects which arm to pull (i.e., which article to show to a user at the news site). In normal operation, Bob shows news articles to users to maximize the click-through rate. However, Alice can attack Bob to change his behavior. For instance, Alice can manipulate the rewards so that users from a particular political base are always shown particular news articles that can reinforce or convert their opinion. Conversely, Alice can coerce the bandit to not show an important article to certain users. As another example, Alice may interfere with clinical trials [16] to funnel most patients toward certain treatment, or make researchers draw wrong conclusions on whether treatment is better than control. Therefore, adversarial attacks on MAB deserve our attention. Insights gained from our study can be used to build defense in the future.

Finally, we note that our setting is motivated by modern industry-scale applications of contextual bandits, where arm selection, reward signal collection, and policy updates are done in a distributed way [3, 18]. Attacks can happen when the reward signal is joined with the selected arm, or when the arm-reward data is sent to another module for Bob to update his policy. In either case, Alice has access to both I_t and r_t^0 for the present and previous rounds.

The rest of the paper is organized as follows. In Section 2, we introduce notations and straightforward attack algorithms that serve as baseline. We then propose our two attack algorithms for ϵ -greedy and UCB in Section 3 and 4 respectively, along with their theoretical attack guarantees. In Section 5, we empirically confirm our findings with toy experiments. Finally, we conclude our paper with related work (Section 6) and a discussion of future work (Section 7) that will enrich our understanding of security vulnerability and defense mechanisms for secure MAB deployment.

2 Preliminaries

Before presenting our main attack algorithms, in this section we first discuss a simple heuristic attack algorithm which serves to illustrate the intrinsic difficulty of attacks. Throughout, we assume Bob runs a bandit algorithm with sublinear pseudo-regret $\mathbb{E} \sum_{t=1}^T (\max_{j=1}^K \mu_j - \mu_{I_t})$. As Alice does not know $\{\mu_i\}$ she must rely on the empirical rewards up to round $t - 1$ to decide the appropriate attack α_t . The attack is online since α_t is computed on-the-fly as I_t and r_t^0 are revealed. The attacking protocol is summarized in Algorithm 1.

Algorithm 1 Alice’s attack against a bandit algorithm

- 1: **Input:** Bob’s bandit algorithm, target arm K
 - 2: **for** $t = 1, 2, \dots$ **do**
 - 3: Bob chooses arm I_t to pull.
 - 4: World generates pre-attack reward r_t^0 .
 - 5: Alice observes I_t and r_t^0 , and then decides the attack α_t .
 - 6: Alice gives $r_t = r_t^0 - \alpha_t$ to Bob.
 - 7: **end for**
-

We assume all arm rewards are σ^2 -sub-Gaussian where σ^2 is known to both Alice and Bob. Let $N_i(t)$ be the number of pulls of arm i up to round t . We say the attack is *successful* after T rounds if the

number of target-arm pulls is $N_K(T) = T - o(T)$ in expectation or with high probability, while minimizing the cumulative attack cost $\sum_{t=1}^T |\alpha_t|$. Other attack settings are discussed in Section 7.

For convenience we define the following quantities:

- $\tau_i(t) := \{s : s \leq t, I_s = i\}$, the set of rounds up to t where arm i is chosen,
- $\hat{\mu}_i^0(t) := N_i(t)^{-1} \sum_{s \in \tau_i(t)} r_s^0$, the pre-attack average reward of arm i up to round t , and
- $\hat{\mu}_i(t) := N_i(t)^{-1} \sum_{s \in \tau_i(t)} r_s$, the corresponding post-attack average reward.

The oracle attack, revisited While the oracle attack was impractical, it gives us a baseline for comparison. The oracle attack drags down the reward of all non-target arms,¹ and can be written as

$$\alpha_t = \mathbb{1}\{I_t \neq K\} \cdot \Delta_{I_t}^\epsilon.$$

Proposition 1 shows that the oracle attack succeeds and requires only a logarithmic attack cost. While more general statements for sublinear-regret algorithms can be made, we focus on logarithmic-regret bandit algorithms for simplicity. Throughout, omitted proofs can be found in our supplementary material.

Proposition 1. *Assume that Bob’s bandit algorithm achieves an $O(\log T)$ regret bound. Then, Alice’s oracle attack with $\epsilon > 0$ succeeds; i.e., $\mathbb{E}N_K(T) = T - o(T)$. Furthermore, the expected total attack cost is $O\left(\sum_{i=1}^{K-1} \Delta_i^\epsilon \log T\right)$.²*

The heuristic constant attack A slight variant of the oracle attack is to attack all the non-target arms with a single constant amount $A > 0$, regardless of the actual μ_i ’s:

$$\alpha_t = \mathbb{1}\{I_t \neq K\} \cdot A.$$

Let $\Delta_i := \Delta_i^0$. Unfortunately, this heuristic constant attack depends critically on the value of A compared to the unknown maximum gap $\max_i \Delta_i$. Proposition 2 states the condition under which the attack succeeds:

Proposition 2. *Assume that Bob’s bandit algorithm achieves an $O(\log T)$ regret bound. Then, Alice’s heuristic constant attack with A succeeds if and only if $A > \max_i \Delta_i$. If the attack succeeds, then the expected attack cost is $O(AK \log T)$.*

Conversely, if $A < \max_i \Delta_i$ the attack fails. This is because in the transformed bandit problem, there exists an arm that has a higher expected reward than arm K , and Bob will mostly pull that arm. Therefore, the heuristic constant attack has to know an unknown quantity to guarantee a successful attack. Moreover, the attack is non-adaptive to the problem difficulty since some Δ_i ’s can be much smaller than A , in which case Alice pays an unnecessarily large attack cost.

We therefore ask the following question:

Does there exist an attacker Alice that guarantees a successful attack with cost adaptive to the problem difficulty?

The answer is yes. We present attack strategies against two popular bandit algorithms of Bob: ϵ -greedy and UCB. We show that Alice can indeed succeed in her attacks and incur cost as small as that of the oracle with an additive term due to the sub-Gaussian noise level σ .

3 Alice’s Attack on ϵ -Greedy Bob

The ϵ -greedy strategy initially pulls each arm once in the first K rounds. For convenience, we assume that the target arm is pulled first: $I_1 = K$. Our results in this section can be adapted to any order of initialization with more complicated notation.

¹The opposite strategy is to push up the target arm: $\alpha_t = \mathbb{1}\{I_t = K\} \cdot (\mu_K - \max_j \mu_j - \epsilon)$ to make arm K the best arm in post-attack rewards. However, a successful attack means that Alice pulls the target arm $T - o(T)$ times; the attack cost is necessarily linear in T , which is inefficient. Simulations that support “drag down” instead of “push up” are presented in Appendix D.

²For near-optimal algorithms like UCB [6], one can find the optimal choice of ϵ . See our supplementary material for detail.

Bob's ϵ -greedy strategy has the following arm-selection rule for $t > K$ [7]:

$$I_t = \begin{cases} \text{draw uniform}[K], & \text{w.p. } \epsilon_t \quad (\text{exploration}) \\ \arg \max_i \hat{\mu}_i(t-1), & \text{otherwise (exploitation)} \end{cases}.$$

The strategy uses an exploration scheme $\{\epsilon_t\}$ over t . Alice's attack algorithm is not aware of $\{\epsilon_t\}$ though her cumulative attack cost $\sum |\alpha_t|$ will implicitly depend on it. Later in Corollary 1 we show that, for the typical decaying scheme $\epsilon_t \propto 1/t$, the cumulative attack cost is mild: $O(\log(t))$.

Alice wants to make Bob *always* pull the target arm during *exploitation* rounds. Since Alice has no influence on which arm is pulled during exploration, this attack goal is the strongest she can achieve. Here, Algorithm 1 is specialized to ensure the following condition:

$$\hat{\mu}_{I_t}(t) \leq \hat{\mu}_K(t) - 2\beta(N_K(t)), \quad (1)$$

where we define $\beta(N)$ as

$$\beta(N) := \sqrt{\frac{2\sigma^2}{N} \log \frac{\pi^2 K N^2}{3\delta}}. \quad (2)$$

From this condition, we derive the actual attack α_t . Since

$$\hat{\mu}_{I_t}(t) = \frac{\hat{\mu}_{I_t}(t-1)N_{I_t}(t-1) + r_t^0 - \alpha_t}{N_{I_t}(t)}, \quad (3)$$

we set the attack in Algorithm 1 as

$$\alpha_t = [\hat{\mu}_{I_t}(t-1)N_{I_t}(t-1) + r_t^0 - (\hat{\mu}_K(t) - 2\beta(N_K(t)))N_{I_t}(t)]_+, \quad (4)$$

where $[z]_+ = \max(0, z)$. Note α is always non-negative, thus the cumulative attack cost can be written without absolute value: $\sum_{t=1}^T \alpha_t$.

With this α_t , we claim that (i) Alice forces Bob to pull the target arm in all exploitation rounds as shown in Lemma 2, and (ii) the cumulative attack cost is logarithmic in t for standard ϵ -greedy learner exploration scheme $\epsilon_t = O(1/t)$ as shown in Corollary 1. Our main result is the following general upper bound on the cumulative attack cost.

Theorem 1. *Let $\delta \leq 1/2$. With probability at least $1 - 2\delta$, for any T satisfying $\sum_{t=1}^T \epsilon_t \geq \frac{K}{e-2} \log(K/\delta)$,³ Alice forces Bob running ϵ -greedy to choose the target arm in at least $\tilde{N}_K(T)$ rounds, using a cumulative attack cost at most*

$$\sum_{t=1}^T |\alpha_t| < \left(\sum_{i=1}^K \Delta_i \right) \tilde{N}(T) + (K-1) \cdot \left(\tilde{N}(T)\beta(\tilde{N}(T)) + 3\tilde{N}(T)\beta(\tilde{N}_K(T)) \right)$$

where

$$\begin{aligned} \tilde{N}(T) &= \left(\frac{\sum_{t=1}^T \epsilon_t}{K} \right) + \sqrt{3 \log \left(\frac{K}{\delta} \right) \left(\frac{\sum_{t=1}^T \epsilon_t}{K} \right)}, \\ \tilde{N}_K(T) &= T - \left(\sum_{t=1}^T \epsilon_t \right) - \sqrt{3 \log \left(\frac{K}{\delta} \right) \left(\sum_{t=1}^T \epsilon_t \right)}. \end{aligned}$$

Before proving the theorem, we first look at its consequence. If Bob's ϵ_t decay scheme is $\epsilon_t = \min\{1, cK/t\}$ for some $c > 0$ as recommended in Auer et al. [7], Alice's cumulative attack cost is $O(\sum_{i=1}^K \Delta_i \log T)$ for large enough T , as the following corollary shows:

Corollary 1. *Inherit the assumptions in Theorem 1. Fix K and δ . If $\epsilon_t = cK/t$ for some constant $c > 0$, then*

$$\sum_{t=1}^T |\alpha_t| = \hat{O} \left(\left(\sum_{i=1}^K \Delta_i \right) \log T + \sigma K \sqrt{\log T} \right), \quad (5)$$

where \hat{O} ignores log log factors.

³ One can drop this condition by considering slightly larger $\tilde{N}(t)$ and smaller $\tilde{N}_K(t)$. However, we keep the condition as it simplifies $\tilde{N}(t)$ and $\tilde{N}_K(t)$. We refer to the proof of Lemma 4 for detail.

Note that the two important constants are $\sum_i \Delta_i$ and σ . While a large σ can increase the cost significantly, the term with $\sum_i \Delta_i$ dominates the cost for large enough T . Specifically, $\sum_i \Delta_i$ is multiplied by $\log T$ that is of higher order than $\sqrt{\log T}$. We empirically verify the scaling of cost with T in Section 5.

To prove Theorem 1, we first show that β in (2) is a high-probability bound on the pre-attack empirical mean of all arms on all rounds. Define the event

$$E := \{\forall i, \forall t > K : |\hat{\mu}_i^0(t) - \mu_i| < \beta(N_i(t))\}. \quad (6)$$

Lemma 1. For $\delta \in (0, 1)$, $\mathbb{P}(E) > 1 - \delta$.

The following lemma proves the first half of our claim.

Lemma 2. For $\delta \leq 1/2$ and under event E , attacks (4) force Bob to always pull the target arm K in exploitation rounds.

We now show that on average each attack on a non-target arm i is not much bigger than Δ_i .

Lemma 3. For $\delta \leq 1/2$ and under event E , we have for all arm $i < K$ and all t that

$$\sum_{s \in \tau_i(t)} |\alpha_s| < (\Delta_i + \beta(N_i(t)) + 3\beta(N_K(t))) N_i(t).$$

Finally, we upper bound the number of non-target arm i pulls $N_i(T)$ for $i < K$. Recall the arm i pulls are only the result of exploration rounds. In round t the exploration probability is ϵ_t ; if Bob explores, he chooses an arm uniformly at random. We also lower bound the target arm pulls $N_K(T)$.

Lemma 4. Let $\delta < 1/2$. Suppose T satisfy $\sum_{t=1}^T \epsilon_t \geq \frac{K}{e-2} \log(K/\delta)$. With probability at least $1 - \delta$, for all non-target arms $i < K$,

$$N_i(T) < \sum_{t=1}^T \frac{\epsilon_t}{K} + \sqrt{3 \sum_{s=1}^T \frac{\epsilon_t}{K} \log \frac{K}{\delta}}.$$

and for the target arm K ,

$$N_K(T) > T - \sum_{t=1}^T \epsilon_t - \sqrt{3 \sum_{s=1}^T \epsilon_t \log \frac{K}{\delta}}.$$

We are now ready to prove Theorem 1.

Proof. The theorem follows immediately from a union bound over Lemma 3 and Lemma 4 below. We add up the attack costs over $K - 1$ non-target arms. Then, we note that $N\beta(N)$ is increasing in N so $N_i(T)\beta(N_i(T)) \leq \tilde{N}(T)\beta(\tilde{N}(T))$. Finally, by Lemma 8 in our supplementary material $\beta(N)$ is decreasing in N , so $\beta(N_K(T)) \leq \beta(\tilde{N}_K(T))$. \square

4 Alice's Attack on UCB Bob

Recall that we assume rewards are σ^2 -sub-Gaussian. Bob's UCB algorithm in its basic form often assumes rewards are bounded in $[0, 1]$; we need to modify the algorithm to handle the more general sub-Gaussian rewards. By choosing $\alpha = 4.5$ and $\psi : \lambda \mapsto \frac{\sigma^2 \lambda^2}{2}$ in the (α, ψ) -UCB algorithm of Bubeck & Cesa-Bianchi [8, Section 2.2], we obtain the following arm-selection rule:

$$I_t = \begin{cases} t, & \text{if } t \leq K \\ \arg \max_i \left\{ \hat{\mu}_i(t-1) + 3\sigma \sqrt{\frac{\log t}{N_i(t-1)}} \right\}, & \text{otherwise.} \end{cases}$$

For the first K rounds where Bob plays each of the K arms once in an arbitrary order, Alice does not attack: $\alpha_t = 0$ for $t \leq K$. After that, attack happens only when $I_t \neq K$. Specifically, consider any round $t > K$ where Bob pulls arm $i \neq K$. It follows from the UCB algorithm that

$$\hat{\mu}_i(t-1) + 3\sigma \sqrt{\frac{\log t}{N_i(t-1)}} \geq \hat{\mu}_K(t-1) + 3\sigma \sqrt{\frac{\log t}{N_K(t-1)}}.$$

Alice attacks as follows. She computes an attack α_t with the smallest absolute value, such that

$$\hat{\mu}_i(t) \leq \hat{\mu}_K(t-1) - 2\beta(N_K(t-1)) - \Delta_0,$$

where $\Delta_0 \geq 0$ is a parameter of Alice. Since the post-attack empirical mean can be computed recursively by the following

$$\hat{\mu}_i(t) = \frac{N_i(t-1)\hat{\mu}_i(t-1) + r_t^0 - \alpha_t}{N_i(t-1) + 1},$$

where r_t^0 is the pre-attack reward; this enables us to write down in closed form Alice's attack:

$$\alpha_t = \left[N_i(t)\hat{\mu}_i^0(t) - \sum_{s \in \tau_i(t-1)} \alpha_s - N_i(t) \cdot (\hat{\mu}_K(t-1) - 2\beta(N_K(t-1)) - \Delta_0) \right]_+. \quad (7)$$

For convenience, define $\alpha_t = 0$ if $I_t = K$. We now present the main theorem on Alice's cumulative attack cost against Bob who runs UCB.

Theorem 2. *Suppose $T \geq 2K$ and $\delta \leq 1/2$. Then, with probability at least $1 - \delta$, Alice forces Bob to choose the target arm in at least*

$$T - (K - 1) \left(2 + \frac{9\sigma^2}{\Delta_0^2} \log T \right),$$

rounds, using a cumulative attack cost at most

$$\sum_{t=1}^T \alpha_t \leq \left(2 + \frac{9\sigma^2}{\Delta_0^2} \log T \right) \sum_{i < K} (\Delta_i + \Delta_0) + \sigma(K-1) \sqrt{32 \left(2 + \frac{9\sigma^2}{\Delta_0^2} \log T \right) \log \frac{\pi^2 K \left(2 + \frac{9\sigma^2}{\Delta_0^2} \log T \right)^2}{3\delta}}.$$

While the bounds in the theorem are somewhat complicated, the next corollary is more interpretable and follows from a straightforward calculation.

Corollary 2. *Inherit the assumptions in Theorem 2 and fix δ . Then, the total number of non-target arm pulls is*

$$O \left(K + \frac{K\sigma^2}{\Delta_0^2} \log T \right),$$

and the cumulative attack cost is

$$\widehat{O} \left(\left(1 + \frac{\sigma^2}{\Delta_0^2} \log T \right) \sum_{i < K} (\Delta_i + \Delta_0) + \sigma K \cdot \left(1 + \frac{\sigma}{\Delta_0} \sqrt{\log T} \right) \sqrt{\log \left(1 + \frac{K\sigma}{\Delta_0} \right)} \right),$$

where \widehat{O} ignores $\log \log(T)$ factors.

We observe that a larger Δ_0 decreases non-target arm pulls (i.e. a more effective attack). The effect diminishes when $\Delta_0 > \sigma \sqrt{\log T}$ since $\frac{K\sigma^2}{\Delta_0^2} \log T < K$. Thus there is no need for Alice to choose a larger Δ_0 . By choosing $\Delta_0 = \Theta(\sigma)$, the cost is $\widehat{O}(\sum_{i < K} \Delta_i \log T + \sigma K \log T)$. This is slightly worse than the cost of attacking ϵ -greedy where σ is multiplied by $\sqrt{\log T}$ rather than $\log T$. However, we find that a stronger attack is possible when the time horizon T is fixed and known to Alice ahead of time (i.e., the fixed budget setting). One can show that the choice $\Delta_0 = \Theta(\sigma \sqrt{\log T})$ minimizes the cumulative attack cost, which is $\widehat{O}(K\sigma \sqrt{\log T})$. This is a very efficient attack since the dominating term w.r.t. T does not depend on $\sum_{i < K} \Delta_i$; in fact the cost associated with $\sum_{i < K} \Delta_i$ does not grow with T at all. Furthermore, such a cost matches the cost of the oracle attack up to doubly-logarithmic factors, which is shown in our supplementary material A.

For the proof of Theorem 2 we use the following two lemmas.

Lemma 5. *Assume event E holds and $\delta \leq 1/2$. Then, for any $i < K$ and any $t \geq 2K$, we have*

$$N_i(t) \leq \min\{N_K(t), 2 + \frac{9\sigma^2}{\Delta_0^2} \log t\}. \quad (8)$$

Lemma 6. *Assume event E holds and $\delta \leq 1/2$. Then, at any round $t \geq 2K$, the cumulative attack cost to any fixed arm $i < K$ can be bounded as:*

$$\sum_{s \in \tau_i(t)} \alpha_s \leq N_i(t) \left(\Delta_i + \Delta_0 + 4\beta(N_i(t)) \right).$$

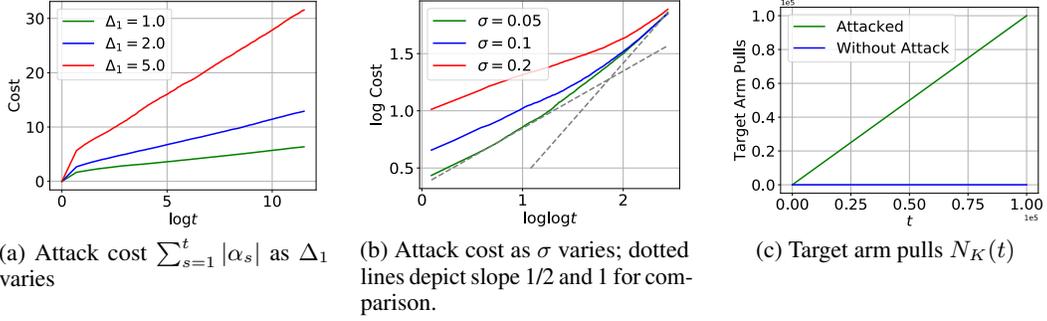


Figure 1: Attack on ϵ -greedy bandit.

Proof of Theorem 2. Suppose event E holds. The bounds are direct consequences of Lemmas 6 and 5 below, by summing the corresponding upper bounds over all non-target arms i . Specifically, the number of target arm pulls is $T - \sum_{i < K} N_i(T)$, and the cumulative attack cost is $\sum_{t=1}^T \alpha_t = \sum_{i < K} \sum_{t \in \tau_i(T)} \alpha_t$. Since event E is true with probability at least $1 - \delta$ (Lemma 1), the bounds also hold with probability at least $1 - \delta$. \square

5 Simulations

In this section, we run simulations on attacking ϵ -greedy and UCB algorithms to illustrate our theoretical findings.

Attacking ϵ -greedy The bandit has two arms. The reward distributions of arms 1 and 2 are $\mathcal{N}(\Delta_1, \sigma^2)$ and $\mathcal{N}(0, \sigma^2)$, respectively, with $\Delta_1 > 0$. Alice’s target arm is arm 2. We let $\delta = 0.025$. Bob’s exploration probability decays as $\epsilon_t = \frac{1}{t}$. We run Alice and Bob for $T = 10^5$ rounds; this forms one trial. We repeat 1000 trials.

In Figure 1(a), we fix $\sigma = 0.1$ and show Alice’s cumulative attack cost $\sum_{s=1}^t |\alpha_s|$ for different Δ_1 values. Each curve is the average over 1000 trials. These curves demonstrate that Alice’s attack cost is proportional to $\log t$ as predicted by Corollary 1. As the reward gap Δ_1 becomes larger, more attack is needed to reduce the reward of arm 1, and the slope increases.

Furthermore, note that $\sum_{t=1}^T |\alpha_t| = \widehat{O}(\Delta_1 \log T + \sigma \sqrt{\log T})$. Ignoring $\log \log T$ terms, we have $\sum_{t=1}^T |\alpha_t| \leq C(\Delta_1 \log T + \sigma \sqrt{\log T})$ for some constant $C > 0$ and large enough T . Therefore, $\log\left(\sum_{t=1}^T |\alpha_t|\right) \leq \max\{\log \log T + \log \Delta_1, \frac{1}{2} \log \log T + \log \sigma\} + \log C$. We thus expect the log-cost curve as a function of $\log \log T$ to behave like the maximum of two lines, one with slope 1/2 and the other with slope 1. Indeed, we observe such a curve in Figure 1(b) where we fix $\Delta_1 = 1$ and vary σ . All the slopes eventually approach 1, though larger σ ’s take a longer time. This implies that the effect of σ diminishes for large enough T , which was predicted by Corollary 1.

In Figure 1(c), we compare the number of target arm (the suboptimal arm 2) pulls with and without attack. This experiment is with $\Delta_1 = 0.1$ and $\sigma = 0.1$. Alice’s attack dramatically forces Bob to pull the target arm. In 10000 rounds, Bob is forced to pull the target arm 9994 rounds with the attack, compared to only 6 rounds if Alice was not present.

Attacking UCB The bandit has two arms. The reward distributions are the same as the ϵ -greedy experiment. We let $\delta = 0.05$. To study how σ and Δ_0 affects the cumulative attack cost, we perform two groups of experiments. In the first group, we fix $\sigma = 0.1$ and vary Alice’s free parameter Δ_0 while in the second group, we fix $\Delta_0 = 0.1$ and vary σ . We perform 100 trials with $T = 10^7$ rounds.

Figure 2(a) shows Alice’s cumulative attack cost as Δ_0 varies. As Δ_0 increases, the cumulative attack cost decreases. In Figure 2(b), we show the cost as σ varies. Note that for large enough t , the cost grows almost linearly with $\log t$, which is implied by Corollary 2. In both figures, there is a large attack near the beginning, after which the cost grows slowly. This is because the initial attacks drag down the empirical average of non-target arms by a large amount, such that the target arm appears to have the best UCB for many subsequent rounds. Figure 2(c) again shows that Alice’s attack forces

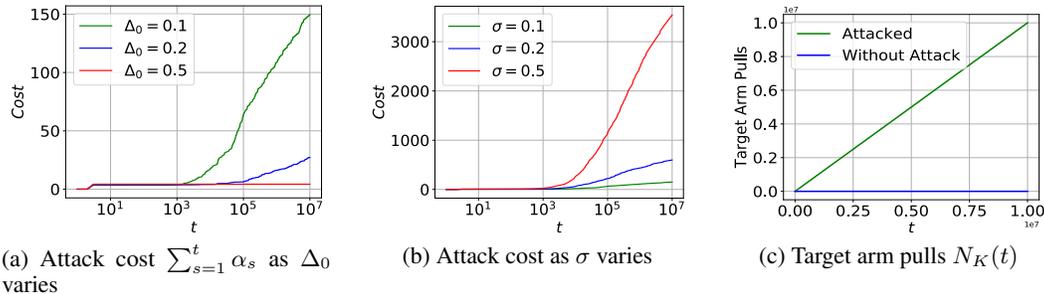


Figure 2: Attack on UCB learner.

Bob to pull the target arm: with attack Bob is forced to pull the target arm $10^7 - 2$ times, compared to only 156 times without attack.

6 Related Work

The literature on general adversarial learning is vast and covers ethics, safety, fairness, and legal concerns; see e.g. Joseph et al. [15] and Goodfellow et al. [12]. Related to MAB, there has been empirical evidence that suggests adversarial attacks can be quite effective, even in the more general multi-step reinforcement learning problems, as opposed to the bandit case considered in this paper. The learned policy may be lured to visit certain target states when adversarial examples are driven [19], or have inferior generalization ability when training examples are corrupted [14]. There are differences, though. In the first, non-stochastic setting [7, 11], the reward is generated by an adversary instead of a stationary, stochastic process. However, the reward observed by the learner is still a *real* reward, in that the learner is still interested in maximizing it, or more precisely, minimizing some notion of regret in reference to some reference policy [8]. Another related problem is reward shaping (e.g., Dorigo & Colombetti [10]), where the reward received by the learner is modified, as in our paper. However, those changes are typically done to *help* the learner in various ways (such as promoting exploration), and are designed in a way not to change the optimal policy the learner eventually converges to [22].

A concurrent work by Lykouris et al. [20] considers a complementary problem to ours. They propose a randomized bandit algorithm that is robust to adversarial attacks on the stochastic rewards. In contrast, our work shows that the existing stochastic algorithms are vulnerable to adversarial attacks. Note that their attack protocol is slightly different in that the attacker has to prepare attacks for all the arms before the learner chooses an arm. Furthermore, they have a different attack cost definition where the cost in a round is the *largest* manipulation over the arms, regardless of which arm the learner selects afterwards.

Another concurrent work by Ma et al. [21] considers attacking stochastic contextual bandit algorithms. The authors show that for a contextual bandit algorithm which periodically updates the arm selection policy, an attacker can perform offline attack to force the contextual bandit algorithm to pull some pre-specified target arm for a given target context vector. Our work differs in that we consider online attack, which is performed on the fly rather than offline.

7 Conclusions and Future Work

We presented a reward-manipulating attack on stochastic MABs. We analyzed the attack against ϵ -greedy and a generalization of the UCB algorithm, and proved that the attacker can force the algorithms to almost always pull a suboptimal target arm. The cost of the attack is only logarithmic in time. Given the wide use of MABs in practice, this is a significant security threat.

Our analysis is only the beginning. We targeted ϵ -greedy and UCB learners for their simplicity and popularity. Future work may look into attacking Thompson sampling [23, 4], linear bandits [1, 5], and contextual bandits [18, 2], etc. We assumed the reward attacks α_t are unbounded from above; new analysis is needed if an application’s reward space is bounded or discrete. It will also be useful

to establish lower bounds on the cumulative attack cost. Specifically, it would be interesting to study pareto optimality w.r.t. the number of target arm pulls and the cumulative attack cost.

Beyond the attack studied in this paper, there is a wide range of possible attacks on MABs. We may organize them along several dimensions:

- Optimal control viewpoint: Our ‘reward shaping’ attack model can be formulated as optimal control [24]. We can define the control cost as $\alpha_t^2 + \lambda \mathbb{1}\{I_t \neq K\}$ and design optimal control strategies.
- The attack goal: The attacker may force the learner into pulling or avoiding target arms, or worsen the learner’s regret, or make the learner identify the wrong best-arm, etc.
- The attack action: The attacker can manipulate the rewards or corrupt the context for contextual bandits, etc.
- Online vs. offline: An online attacker must choose the attack action in real time; An offline attacker poisons a dataset of historical action-reward pairs in batch mode, then the learner learns from the poisoned dataset.

The combination of these attack dimensions presents fertile ground for future research into both bandit-algorithm attacks and the corresponding defense mechanisms.

Acknowledgments

This work is supported in part by NSF 1837132, 1545481, 1704117, 1623605, 1561512, and the MADLab AF Center of Excellence FA9550-18-1-0166.

References

- [1] Abbasi-Yadkori, Yasin, Pál, Dávid, and Szepesvári, Csaba. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2312–2320, 2011.
- [2] Agarwal, Alekh, Hsu, Daniel, Kale, Satyen, Langford, John, Li, Lihong, and Schapire, Robert E. Taming the monster: A fast and simple algorithm for contextual bandits. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1638–1646, 2014.
- [3] Agarwal, Alekh, Bird, Sarah, Cozowicz, Markus, Hoang, Luong, Langford, John, Lee, Stephen, Li, Jiaji, Melamed, Dan, Oshri, Gal, Ribas, Oswaldo, Sen, Siddhartha, and Slivkins, Alex. Making contextual decisions with low technical debt. CoRR abs/1606.03966, 2016.
- [4] Agrawal, Shipra and Goyal, Navin. Analysis of Thompson Sampling for the Multi-armed Bandit Problem. In *In Proceedings of the Conference on Learning Theory (COLT)*, volume 23, pp. 39.1–39.26, 2012.
- [5] Agrawal, Shipra and Goyal, Navin. Thompson Sampling for Contextual Bandits with Linear Payoffs. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 127–135, 2013.
- [6] Auer, Peter, Cesa-Bianchi, Nicolò, and Fischer, Paul. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2–3):235–256, 2002.
- [7] Auer, Peter, Cesa-Bianchi, Nicolò, Freund, Yoav, and Schapire, Robert E. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.
- [8] Bubeck, Sébastien and Cesa-Bianchi, Nicolò. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. *Foundations and Trends in Machine Learning*, 5:1–122, 2012.
- [9] Chapelle, Olivier, Manavoglu, Eren, and Rosales, Romer. Simple and scalable response prediction for display advertising. *ACM Transactions on Intelligent Systems and Technology*, 5(4):61:1–61:34, 2014.
- [10] Dorigo, Marco and Colombetti, Luca Marco. *Robot Shaping: An Experiment in Behavior Engineering*. MIT Press, 1997. ISBN 0-262-04164-2.

- [11] Even-Dar, Eyal, Kakade, Sham M., and Mansour, Yishay. Online Markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009.
- [12] Goodfellow, Ian J, Shlens, Jonathon, and Szegedy, Christian. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.
- [13] Greenewald, Kristjan, Tewari, Ambuj, Murphy, Susan A., and Klasnja, Predrag V. Action centered contextual bandits. In *Advances in Neural Information Processing Systems 30 (NIPS)*, pp. 5979–5987, 2017.
- [14] Huang, Sandy, Papernot, Nicolas, Goodfellow, Ian, Duan, Yan, and Abbeel, Pieter. Adversarial attacks on neural network policies, 2017. arXiv:1702.02284.
- [15] Joseph, Anthony D., Nelson, Blaine, Rubinstein, Benjamin I. P., and Tygar, J.D. *Adversarial Machine Learning*. Cambridge University Press, 2018.
- [16] Kuleshov, Volodymyr and Precup, Doina. Algorithms for multi-armed bandit problems. CoRR abs/1402.6028, 2014.
- [17] Kveton, Branislav, Szepesvári, Csaba, Wen, Zheng, and Ashkan, Azin. Cascading bandits: Learning to rank in the cascade model. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pp. 767–776, 2015.
- [18] Li, Lihong, Chu, Wei, Langford, John, and Schapire, Robert E. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the Nineteenth International Conference on World Wide Web (WWW)*, pp. 661–670, 2010.
- [19] Lin, Yen-Chen, Hong, Zhang-Wei, Liao, Yuan-Hong, Shih, Meng-Li, Liu, Ming-Yu, and Sun, Min. Tactics of adversarial attack on deep reinforcement learning agents. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3756–3762, 2017.
- [20] Lykouris, Thodoris, Mirrokni, Vahab, and Paes Leme, Renato. Stochastic bandits robust to adversarial corruptions. In *Proceedings of the Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pp. 114–122, 2018.
- [21] Ma, Yuzhe, Jun, Kwang-Sung, Li, Lihong, and Zhu, Xiaojin. Data poisoning attacks in contextual bandits. In *Conference on Decision and Game Theory for Security (GameSec)*, 2018.
- [22] Ng, Andrew Y., Harada, Daishi, and Russell, Stuart J. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the 16th International Conference on Machine Learning (ICML)*, pp. 278–287, 1999.
- [23] Thompson, William R. On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika*, 25(3/4):285, 1933.
- [24] Zhu, Xiaojin. An optimal control view of adversarial machine learning, 2018. arXiv:1811.04422.

Supplementary Material

A Details on the oracle and constant attack

Logarithmic regret and the suboptimal arm pull counts. For simplicity, denote by i^* the *unique* best arm; that is, $i^* = \arg \max_{i=1, \dots, K} \mu_i$. We show that a logarithmic regret bound implies that the arm pull count of arm $i \neq i^*$ is at most logarithmic in T .

Lemma 7. *Assume that a bandit algorithm enjoys a regret bound of $O(\log(T))$. Then, $\mathbb{E}N_i(T) = O(\log(T))$, $\forall i \neq i^*$.*

Proof. The logarithmic regret bound implies that for a large enough T there exists $C > 0$ such that $\sum_{i=1}^K \mathbb{E}N_i(T)(\mu_{i^*} - \mu_i) \leq C \log T$. Therefore, for any $i \neq i^*$, we have $\mathbb{E}N_i(T)(\mu_{i^*} - \mu_i) \leq C \log T$, which implies that

$$\mathbb{E}N_i(T) \leq \frac{C}{\mu_{i^*} - \mu_i} \log T = O(\log T).$$

□

Proof of Proposition 1 By Lemma 7, a logarithmic regret bound implies that the bandit algorithm satisfies $\mathbb{E}N_i(T) = O(\log(T))$. That is, for a large enough T , $\mathbb{E}N_i(T) \leq C_i \log(T)$ for some $C_i > 0$. Based on the view that the oracle attack effectively shifts the means μ_1, \dots, μ_K , the best arm is now the K -th arm. Then, $\mathbb{E}N_K(T) = T - \sum_{i \neq K} \mathbb{E}N_i(T) \geq T - \sum_{i \neq K} C_i \log T = T - o(T)$, which proves the first statement.

For the second statement, we notice that $\mathbb{E}N_i(T) = C_i \log T$ for any $i \neq K$ and that we do not attack the K -th arm. Therefore,

$$\mathbb{E} \left[\sum_{t=1}^T |\alpha_t| \right] = \sum_{i=1}^{K-1} \mathbb{E}N_i(T) \cdot \Delta_i^\epsilon \leq \sum_{i=1}^{K-1} C_i \Delta_i^\epsilon \log T = O \left(\sum_{i=1}^{K-1} \Delta_i^\epsilon \log T \right).$$

Proof of Proposition 2 By Lemma 7, a logarithmic regret bound implies that the bandit algorithm satisfies $\mathbb{E}N_i(T) = O(\log(T))$. Note that the constant attack effectively shifts the means of all the arms by A except for the K -th arm. Since $A > \max_i \Delta_i$, the best arm is now the K -th arm. Then, $\mathbb{E}N_K(T) = T - \sum_{i=1}^{K-1} \mathbb{E}N_i(T) \geq T - \sum_{i=1}^{K-1} C_i \log T = T - o(T)$, which proves the first statement.

For the second statement, we notice that $\mathbb{E}N_i(T) = C_i \log T$ for any $i \neq K$, and we do not attack the K -th arm. Therefore,

$$\mathbb{E} \left[\sum_{t=1}^T |\alpha_t| \right] = \sum_{i=1}^{K-1} \mathbb{E}N_i(T) \cdot A \leq A \sum_{i=1}^{K-1} C_i \log T = O(AK \cdot \log T).$$

The best ϵ for Alice's oracle attack Consider the case where Bob employs a near-optimal bandit algorithm such as UCB [6], which enjoys $\mathbb{E}N_i(T) = \Theta(1 + \sigma^2 \Delta_i^{-2} \log T)$. When the time horizon T is known ahead of time, one can compute the best ϵ ahead of time. Hereafter, we omit unimportant constant factors for simplicity. Since Alice employs the oracle attack, Bob pulls each arm $1 + \sigma^2 \epsilon^{-2} \log(T)$ times for some $C > 0$ in expectation. Assuming that the target arm is K , the attack cost is

$$\sum_{i=1}^{K-1} \Delta_i^\epsilon \cdot (1 + \sigma^2 \epsilon^{-2} \log(T)) = \sum_{i=1}^{K-1} \Delta_i + (K-1) \cdot \epsilon + \sum_{i=1}^{K-1} \left(\frac{\Delta_i}{\epsilon^2} + \frac{1}{\epsilon} \right) \sigma^2 \log T$$

To balance the two terms, one can see that ϵ has to grow with T and the term Δ_i/ϵ^2 is soon dominated by $1/\epsilon$. Thus, for large enough T the optimal choice of ϵ is $\sigma \sqrt{\log(T)}$, which leads to the attack cost of $O(\sum_{i=1}^{K-1} \Delta_i + \sigma K \sqrt{\log T})$.

B Details on attacking the ϵ -greedy strategy

Lemma 8. For $\delta \leq 1/2$, the $\beta(N)$ defined in (2) is monotonically decreasing in N .

Proof. It suffices to show that $f(x) = \frac{2\sigma^2}{x} \log \frac{\pi^2 K x^2}{3\delta}$ is decreasing for $x \geq 1$. Note that $\delta \leq 1/2 \leq \frac{K}{3} (\frac{\pi}{e})^2$, thus for $x \geq 1$ we have

$$\begin{aligned} f'(x) &= -\frac{2\sigma^2}{x^2} \log \frac{\pi^2 K x^2}{3\delta} + \frac{2\sigma^2}{x} \frac{3\delta}{\pi^2 K x^2} \frac{2\pi^2 K x}{3\delta} \\ &= \frac{2\sigma^2}{x^2} (2 - \log \frac{\pi^2 K x^2}{3\delta}) \leq \frac{2\sigma^2}{x^2} (2 - \log \frac{\pi^2 K}{3\delta}) \\ &\leq \frac{2\sigma^2}{x^2} (2 - \log e^2) = 0. \end{aligned}$$

□

Proof of Corollary 1 When T is larger than the following threshold:

$$\frac{K+1}{K} \left(\sum_{t=1}^T \epsilon_t \right) + \sqrt{12 \log(K/\delta) \left(\frac{K+1}{K} \sum_{t=1}^T \epsilon_t \right)},$$

we have $\tilde{N}_K(T) \geq \tilde{N}(T)$. Because $\beta(N)$ is decreasing in N ,

$$\tilde{N}(T)\beta(\tilde{N}(T)) + 3\tilde{N}(T)\beta(\tilde{N}_K(T)) \leq 4\tilde{N}(T)\beta(\tilde{N}(T)). \quad (9)$$

Due to the the exploration scheme of the strategy,

$$\sum_{t=1}^T \epsilon_t = cK \sum_{t=1}^T 1/t \leq cK(\log(T) + 1).$$

Thus by the definition of $\tilde{N}(T)$,

$$\tilde{N}(T) \leq c(\log T + 1) + \sqrt{3 \log \left(\frac{K}{\delta} \right) c(\log T + 1)}.$$

For sufficiently large T , there exists a constant c_2 depending on c, K, δ to further upper bound the RHS as follows:

$$c(\log T + 1) + \sqrt{3 \log \left(\frac{K}{\delta} \right) c(\log T + 1)} \leq c_2 \log T := \check{N}(T). \quad (10)$$

Since $N\beta(N)$ is increasing in N , combining (9) and (10) we have for sufficiently large T ,

$$\tilde{N}(T)\beta(\tilde{N}(T)) + 3\tilde{N}(T)\beta(\tilde{N}_K(T)) \leq 4\check{N}(T)\beta(\check{N}(T)).$$

Plugging this upper bound into Theorem 1,

$$\begin{aligned} \sum_{t=1}^T \alpha_t &< \left(\sum_{i=1}^K \Delta_i \right) \check{N}(T) + 4(K-1)\check{N}(T)\beta(\check{N}(T)) \\ &= c_2 \left(\sum_{i=1}^K \Delta_i \right) \log T + \sqrt{32c_2}(K-1)\sigma \cdot \sqrt{\log T \left(2 \log \log T + \log \frac{\pi^2 K c_2^2}{3\delta} \right)}. \end{aligned} \quad (11)$$

Proof of Lemma 1 Let $\{X_j\}_{j=1}^\infty$ be a sequence of *i.i.d.* σ^2 -sub-Gaussian random variables with mean μ . Let $\hat{\mu}_N^0 = \frac{1}{N} \sum_{j=1}^N X_j$. By Hoeffding's inequality

$$\mathbb{P}(|\hat{\mu}_N^0 - \mu| \geq \eta) \leq 2 \exp \left(-\frac{N\eta^2}{2\sigma^2} \right).$$

Define $\delta_{iN} := \frac{6\delta}{\pi^2 N^2 K}$. Apply union bound over arms i and pull counts $N \in \mathbb{N}$,

$$\mathbb{P}(\exists i, N : |\hat{\mu}_{i,N}^0 - \mu_i| \geq \beta(N)) \leq \sum_{i=1}^K \sum_{N=1}^{\infty} \delta_{iN} = \delta.$$

Proof of Lemma 2 We show by induction that at the end of any round $t \geq K$ Algorithm 1 maintains the invariance

$$\hat{\mu}_K(t) > \hat{\mu}_i(t), \quad \forall i < K, \quad (12)$$

which forces the learner to pull arm K if $t + 1$ is an exploitation round.

Base case: By definition the learner pulls arm K first, then all the other arms once. During round $t = 2 \dots K$ the attack algorithm ensures $\hat{\mu}_i(t) \leq \hat{\mu}_K(t) - 2\beta(1) < \hat{\mu}_K(t)$ for arms $i < K$, trivially satisfying (12).

Induction: Suppose (12) is true for rounds up to $t - 1$. Consider two cases for round t :

If round t is an exploration round and $I_t \neq K$ is pulled, then only $\hat{\mu}_{I_t}(t)$ changes; the other arms copy their empirical mean from round $t - 1$. The attack algorithm ensures $\hat{\mu}_K(t) \geq \hat{\mu}_{I_t}(t) + 2\beta(N_K(t)) > \hat{\mu}_{I_t}(t)$. Thus (12) is satisfied at t .

Otherwise either t is exploration and K is pulled; or t is exploitation – in which case K is pulled because by inductive assumption (12) is satisfied at the end of $t - 1$. Regardless, this arm K pull is not attacked by Algorithm 1 and its empirical mean is updated by the pre-attack reward. We show this update does not affect the dominance of $\hat{\mu}_K(t)$. Consider any non-target arm $i < K$. Denote the last time $\hat{\mu}_i$ was changed by t' . Note $t' < t$ and $N_K(t') < N_K(t)$. At round t' , Algorithm 1 ensured that $\hat{\mu}_i(t') \leq \hat{\mu}_K(t') - 2\beta(N_K(t'))$. We have:

$$\begin{aligned} \hat{\mu}_K(t) &= \hat{\mu}_K^0(t) && \text{(arm } K \text{ never attacked)} \\ &> \mu_K^0 - \beta(N_K(t)) && \text{((6) lower bound)} \\ &> \mu_K^0 - \beta(N_K(t')) && \text{(Lemma 8)} \\ &> \hat{\mu}_K(t') - 2\beta(N_K(t')) && \text{((6) upper bound)} \\ &\geq \hat{\mu}_i(t') && \text{(Algorithm 1)} \\ &= \hat{\mu}_i(t). \end{aligned}$$

Thus (12) is also satisfied at round t .

Proof of Lemma 3 Without loss of generality assume in round t arm i is pulled and the attacker needed to attack the reward (i.e. $I_t = i$ and $\alpha_t > 0$). By definition (4),

$$\begin{aligned} \alpha_t &= \hat{\mu}_i(t-1)N_i(t-1) + r_t^0 - (\hat{\mu}_K(t) - 2\beta(N_K(t)))N_i(t) \\ &= \sum_{s \in \tau_i(t-1)} (r_s^0 - \alpha_s) + r_t^0 - (\hat{\mu}_K(t) - 2\beta(N_K(t)))N_i(t) \\ &= \sum_{s \in \tau_i(t)} r_s^0 - \sum_{s \in \tau_i(t-1)} \alpha_s - (\hat{\mu}_K(t) - 2\beta(N_K(t)))N_i(t). \end{aligned}$$

Therefore, the cumulative attack on arm i is

$$\begin{aligned} \sum_{s \in \tau_i(t)} \alpha_s &= \sum_{s \in \tau_i(t)} r_s^0 - (\hat{\mu}_K(t) - 2\beta(N_K(t)))N_i(t) \\ &= (\hat{\mu}_i^0(t) - \hat{\mu}_K(t) + 2\beta(N_K(t)))N_i(t). \end{aligned}$$

One can think of the term in front of $N_i(t)$ as the amortized attack cost against arm i . By Lemma 1,

$$\begin{aligned} \hat{\mu}_i^0(t) &< \mu_i + \beta(N_i(t)) \\ \hat{\mu}_K(t) = \hat{\mu}_K^0(t) &> \mu_K - \beta(N_K(t)) \end{aligned}$$

Therefore,

$$\begin{aligned} \sum_{s: I_s=i}^t \alpha_s &< (\mu_i - \mu_K + \beta(N_i(t)) + 3\beta(N_K(t)))N_i(t) \\ &\leq (\Delta_i + \beta(N_i(t)) + 3\beta(N_K(t)))N_i(t). \end{aligned}$$

The last inequality follows from the gap definition $\Delta_i := [\mu_i - \mu_K]_+$.

Proof of Lemma 4 Fix a non-target arm $i < K$. Let X_t be the Bernoulli random variable for round T being arm i pulled. Then,

$$\begin{aligned} N_i(T) &= \sum_{t=1}^T X_t \\ \mathbb{E}[X_t] &= \frac{\epsilon_t}{K} \\ \mathbb{V}[X_t] &= \frac{\epsilon_t}{K} \left(1 - \frac{\epsilon_t}{K}\right) < \frac{\epsilon_t}{K}. \end{aligned}$$

Since X_t 's are independent random variables, we may apply Lemma 9 of Agarwal et al. [2], so that for any $\lambda \in [0, 1]$, with probability at least $1 - \delta/K$,

$$\begin{aligned} \sum_{t=1}^T \left(X_t - \frac{\epsilon_t}{K}\right) &\leq (e-2)\lambda \sum_{t=1}^T \mathbb{V}[X_t] + \frac{1}{\lambda} \log \frac{K}{\delta} \\ &< (e-2)\lambda \sum_{t=1}^T \mathbb{E}[X_t] + \frac{1}{\lambda} \log \frac{K}{\delta}. \end{aligned}$$

Choose $\lambda = \sqrt{\frac{\log(K/\delta)}{(e-2) \sum_{t=1}^T \mathbb{E}[X_t]}}$, and we get that

$$\begin{aligned} \sum_{t=1}^T X_t &< \sum_{t=1}^T \frac{\epsilon_t}{K} + 2\sqrt{(e-2) \sum_{t=1}^T \frac{\epsilon_t}{K} \log \frac{K}{\delta}} \\ &< \sum_{t=1}^T \frac{\epsilon_t}{K} + \sqrt{3 \sum_{t=1}^T \frac{\epsilon_t}{K} \log \frac{K}{\delta}} := \tilde{N}(T). \end{aligned}$$

The same reasoning can be applied to all non-target arm $i < K$.⁴

The case with the target arm is similar, with the only change that $\mathbb{E}[X_t] > 1 - \epsilon_t$ and $\mathbb{V}[X_t] < \epsilon_t$, leading to the lower bound:

$$N_K(T) > T - \sum_{t=1}^T \epsilon_t - \sqrt{3 \sum_{t=1}^T \epsilon_t \log \frac{K}{\delta}} =: \tilde{N}_K(T).$$

Finally, a union bound is applied to all K arms to complete the proof.

C Details on attacking the UCB strategy

Proof of Lemma 5 Fix some $t \geq 2K$. If $N_i(t) \leq 2$ for all $i < K$, then $N_K(t) \geq 2$, which implies $N_i(t) \leq \min\{N_K(t), 2\}$. Thus, (8) holds trivially and we are done.

Now fix any $i < K$ such that $N_i(t) > 2$. As the desired upper bound is nondecreasing in t , we only need to prove the result for t where $I_t = i$. Let t' be the previous time where arm i was pulled. Note that t' satisfies $K < t' < t$ as $N_i(t) > 2$, so the attacker has started attacking at round t' . This implies that $N_i(t' - 1) + 1 = N_i(t') = N_i(t - 1) = N_i(t) - 1$.

On one hand, it is clear that after attack $\alpha_{t'}$ was added at round t' , the following holds:

$$\hat{\mu}_i(t') \leq \hat{\mu}_K(t') - 2\beta(N_K(t')) - \Delta_0. \quad (13)$$

⁴Note the upper bound above is valid for T such that $\sum_{t=1}^T \epsilon_t \geq \frac{K}{e-2} \log(K/\delta)$ only as otherwise λ is greater than 1. One can get rid of such a condition by a slightly looser bound. Specifically, using $\lambda = 1$ gives us a bound that holds true for all T . We then take the max of the two bounds, which can be simplified as $\sum_{t=1}^T X_t < (e-1) \sum_{t=1}^T \frac{\epsilon_t}{K} + \sqrt{3 \sum_{t=1}^T \frac{\epsilon_t}{K} \log \frac{K}{\delta}} + \log \frac{K}{\delta}$. The condition on T in Theorem 1 can be removed using this bound. However, by keeping the mild assumption on T we keep the exposition simple.

On the other hand, at round t , it must be the case that

$$\hat{\mu}_i(t-1) + 3\sigma\sqrt{\frac{\log t}{N_i(t-1)}} \geq \hat{\mu}_K(t-1) + 3\sigma\sqrt{\frac{\log t}{N_K(t-1)}},$$

which is equivalent to

$$\hat{\mu}_i(t') + 3\sigma\sqrt{\frac{\log t}{N_i(t')}} \geq \hat{\mu}_K(t-1) + 3\sigma\sqrt{\frac{\log t}{N_K(t-1)}}.$$

Therefore,

$$\begin{aligned} 3\sigma\sqrt{\frac{\log t}{N_i(t')}} - 3\sigma\sqrt{\frac{\log t}{N_K(t-1)}} &\geq \hat{\mu}_K(t-1) - \hat{\mu}_i(t') \\ &\geq \hat{\mu}_K(t-1) - \hat{\mu}_K(t') + 2\beta(N_K(t')) + \Delta_0 \\ &\geq \Delta_0, \end{aligned}$$

where we have used Eqn. 13 in the second inequality, the condition in event E as well as Lemma 8 in the third. Since $\Delta_0 > 0$, we can see that $N_i(t') < N_K(t-1)$, and thus

$$N_i(t) = N_i(t') + 1 \leq N_K(t-1) = N_K(t). \quad (14)$$

Furthermore, since $3\sigma\sqrt{\frac{\log t}{N_K(t-1)}} > 0$, we have $3\sigma\sqrt{\frac{\log t}{N_i(t')}} > \Delta_0$, which implies

$$N_i(t) = 1 + N_i(t') \leq 1 + \frac{9\sigma^2}{\Delta_0^2} \log t. \quad (15)$$

Combining (14) and (15) gives the desired bound (8).

Proof of Lemma 6 Fix any $i < K$. As the desired upper bound is increasing in t , we only need to prove the result for t where $I_t = i$ and $\alpha_t > 0$. It follows from (7) that,

$$\frac{1}{N_i(t)} \sum_{s \in \tau_i(t)} \alpha_s = \hat{\mu}_i^0(t) - \hat{\mu}_K(t-1) + 2\beta(N_K(t-1)) + \Delta_0.$$

Since event E holds, we have

$$\frac{1}{N_i(t)} \sum_{s \in \tau_i(t)} \alpha_s \leq \Delta_i + \Delta_0 + \beta(N_i(t)) + 3\beta(N_K(t-1)).$$

The proof is completed by observing $N_K(t-1) = N_K(t)$, $N_i(t) \leq N_K(t)$ (Lemma 5) and Lemma 8.

D Simulations on Heuristic Constant Attack

We run simulations on ϵ -greedy and UCB to illustrate the heuristic constant attack algorithm. The bandit has two arms, where the reward distributions are $\mathcal{N}(1, 0.1^2)$ and $\mathcal{N}(0, 0.1^2)$ respectively, thus $\max_i \Delta_i = \mu_1 - \mu_2 = 1$. Alice's target arm is arm 2. In our experiment, Alice tried two different constants for attack: $A = 1.2$ and $A = 0.8$, one being greater and the other being smaller than $\max_i \Delta_i$. We run the attack for $T = 10^4$ rounds. Fig. 3 and Fig. 4 show Alice's cumulative attack cost and Bob's number of target arm pulls $N_K(t)$ for ϵ -greedy and UCB. Note that if $A > \max_i \Delta_i$, then $N_K(t) \approx t$, which verifies that Alice succeeds with the heuristic constant attack. At the same time, pushing up the target arm would incur linear cost; while dragging down the non-target arm achieves logarithmic cost. In summary, Alice should use an A value larger than Δ , and should drag down the expected reward of the non-target arm by amount A .

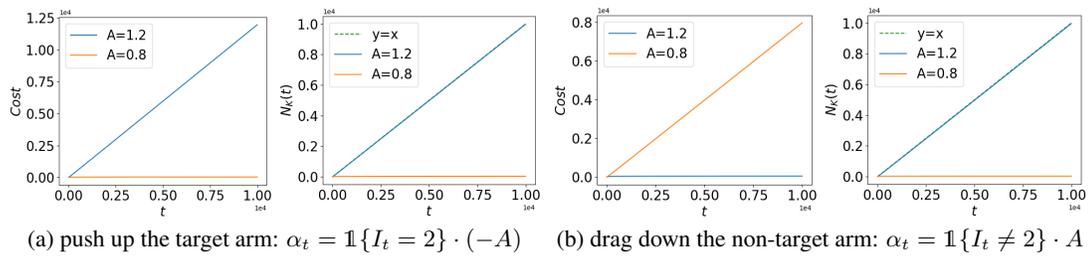


Figure 3: Constant attack on ϵ -greedy

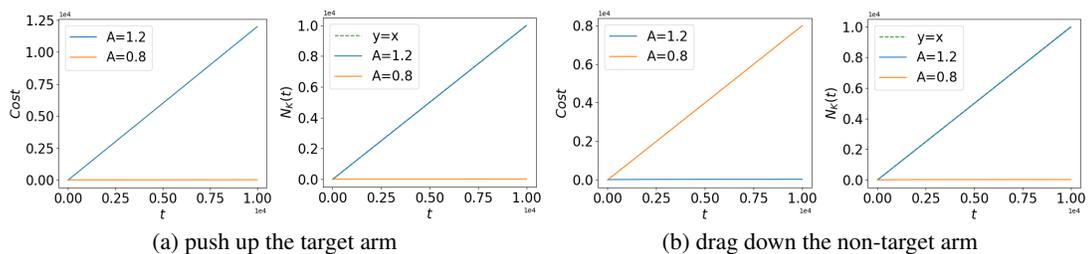


Figure 4: Constant attack on UCB1