Adversarial Bandits with Corruptions

Lin Yang and Mohammad H. Hajiesmaili
University of Massachusetts Amherst
{linyang,hajiesmaili}@cs.umass.edu

M. Sadegh Talebi
University of Copenhagen
m.shahi@di.ku.dk

John C. S. Lui and Wing S. Wong
Chinese University of Hong Kong
cslui@cse.cuhk.edu.hk,wswong@ie.cuhk.edu.hk

Abstract

This paper studies adversarial bandits with corruptions. In the basic adversarial bandit setting, the reward of arms is predetermined by an adversary who is oblivious to the learner's policy. In this paper, we consider an extended setting in which an attacker sits in-between the environment and the learner, and is endowed with a limited budget to corrupt the reward of the selected arm. We have two main results. First, we derive a lower bound on the regret of any bandit algorithm that is aware of the budget of the attacker. Also, for budget-agnostic algorithms, we characterize an impossibility result demonstrating that even when the attacker has a sublinear budget, i.e., a budget growing sublinearly with time horizon T, they fail to achieve a sublinear regret. Second, we propose `ExpRb`, a bandit algorithm that incorporates a biased estimator and a robustness parameter to deal with corruption. We characterize the regret of `ExpRb` and show that for the case of a known corruption budget, the regret of `ExpRb` is tight.

1 Introduction

Multi-armed bandits (MABs) [23] present a powerful online learning framework that is applicable to a broad range of application domains including medical trials, web search advertisement, datacenter design, and recommender systems; see, e.g., [5, 24] and references therein. In the basic MAB problem, in each round a learner pulls an arm (corresponding to selecting an action) from a finite set of arms, and observes the reward associated to the selected arm, but not for the other unselected arms. The goal of the learner is to maximize the rewards accumulated in the course of her interaction. MAB problems are typically categorized into stochastic and non-stochastic (or adversarial) problems depending on how the reward sequences are generated. In stochastic bandits [23, 14], rewards are drawn from fixed but unknown distributions, whereas in non-stochastic bandits [3], no statistical assumption on rewards are made and rewards are arbitrary as if they were generated by an adversary.

Motivated by malicious activities in bandit-related applications such as click fraud via malware [26, 22], fake reviews and ratings in recommender systems [11, 17, 27], and email spam [13, 6], there have been recent effort on studying bandit problems under some notion of *corruption* [12, 28, 16, 18, 9, 15, 8, 19]. In the case of click fraud, for example, botnets maliciously simulate users clicking on an ad to mislead learning algorithms. More specifically, there are some rewards (click rates) associated to each arm (ad), and an attacker (the botnet) corrupts the rewards based on the learner's action. The majority of past efforts, however, are limited to studying stochastic bandits with corruption, either on understanding the vulnerability of existing algorithms and designing attacks [12, 28, 16, 15, 8, 19], or developing algorithms that are robust against corruption [18, 9, 30]. In those works, stochastic patterns are corrupted by an attacker and bandit algorithms strive to be robust against the corruption. A detailed literature review is provided in §A of the supplementary material.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Table 1: Summary of prior literature and this work

	Stochastic Bandits				Non-stochastic Bandits		
Reference	Oblivious	Targeted	Vulnerability	Robustness	Targeted	Vulnerability	Robustness
Lykouris *et al.* [18]	✓		✓	✓			
Gupta *et al.* [9]	✓		✓	✓			
Jun *et al.* [12]		✓	✓				
Liu *et al.* [16]	✓	✓	✓				
This work		✓	✓		✓	✓	✓

In contrast, this paper is the first, to the best of our knowledge, that studies *non-stochastic bandits with corruptions*. In some application domains such as shortest path routing [21] and inventory control problem [7], the reward functions are very complex to model using stochastic bandits, and hence, from a practical perspective, non-stochastic bandits are relevant for such intrinsically involved applications.

A concrete example of non-stochastic bandits with corruptions is the Online Shortest Path Routing (OSPR) problem under the denial of service (DoS) attacks. OSPR is a classic example of MAB problems that has been studied in both stochastic and adversarial settings [4, 21, 25]. And there is also extensive research on routing under DoS attacks, including the recent work [29] focusing on bandit modeling of this scenario. OSPR could be reasonably modeled as non-stochastic bandits when the delays on the links change dynamically in an predictable manner [10], or in situations where the combined distribution of a path including multiple links is difficult to characterize [21]. In this non-stochastic scenario, the DoS attack could be modeled by our bandit with targeted corruptions. Specifically, the DoS attacker can be aware of the selected paths by detecting the transmitted packets over the path and manipulate the latency of the selected path by flooding the path with dummy packets. Also, the budget of the attacker is simply the available resources for the DoS attacker to keep her undetectable. Arguably, none of "non-stochastic bandits" and "stochastic bandits with corruption" models alone would suffice to fully characterize the underlying model here. In addition, this problem is interesting with unique challenges different from stochastic bandits with corruptions and calls for non-trivial algorithm design and regret analysis. Consequently, studying the vulnerability and robustness of non-stochastic bandit algorithms with corruption becomes important as well. We formally define the corruption model in the following.

1.1 The Corruption Model

Consider a K-armed non-stocahstic bandit, similar to the model in [3], where the rewards are generated by an adversary *obliviously*, namely they are generated before the game starts. At each round $t \in [T]$, the learner selects an arm $I_t \in [K]$ with the *primary reward* $x_{I_t}(t) \in [0, 1]$. In the corruption model, *there is an attacker that sits in-between the environment and the learner, observes the arm chosen by the learner, and corrupts its rewards aiming to mislead the learner to select sub-optimal arms.* More specifically, the attacker manipulates the reward into $\tilde{x}_{I_t}(t) = x_{I_t}(t) - a(t)$, where $a(t) \in [x_{I_t}(t) - 1, x_{I_t}(t)]$ denotes the attack in round t. The learner receives $\tilde{x}_{I_t}(t)$ without knowing the original reward $x_{I_t}(t)$. The attacker is aware of the selected arm, and can set the value of $a(t)$ to attack the learner to end up with selecting a sub-optimal arm. Further, similar to existing work on stochastic bandits with corruptions [9, 18], the total budget of the attacker is upper bounded by Φ. The formal statement of the model is given in §2. We emphasize that while considering an *oblivious adversary*, in this model the attacker manipulates the reward adaptively to the learner's chosen arm; hence, the attacker is different from the adaptive adversary in which the rewards is determined right before the learner's action. We refer to this attack as *targeted*. In contrast, the prior literature on stochastic bandits with corruption [18, 9] assume an oblivious attacker who manipulates the rewards before observing the learner's chosen arm. We call these attacks *oblivious*. Last, we refer to the algorithms that are unaware of the existence of the attacker (or its budget) as *attach-agnostic* algorithms, and *attack-aware* algorithms know the attacker and its budget.

1.2 Summary of Contributions

In addition to introducing the above non-stochastic bandits with targeted corruptions, this paper investigates the vulnerability of attack-agnostic algorithms and establishes a regret lower bound for

attack-aware algorithms. Then, as the main contribution, this paper presents a robust bandit algorithm in the corrupted setting. Table 1 highlights the high-level contributions of this work as compared to the related literature.

1.2.1 Vulnerability and Regret Lower Bound

We first derive an impossibility result for obtaining a sublinear regret for *attack-agnostic* algorithms for non-stochastic bandits with a sublinear attacker. Our results, presented in Theorem 1 in §3, show that even when an attacker has a sublinear budget, any attack-agnostic bandit algorithm fails to achieve a sublinear regret. This impossibility result applies to stochastic bandit algorithms with targeted corruptions as well. Our impossibility result does not contradict the attack-agnostic algorithms in [18, 9] that develop no-regret algorithms for oblivious attackers.

1.2.2 Robust Algorithm Design and Regret Analysis

As the main contribution, in §4, we then propose `ExpRb`, that if aware of Φ, achieves a sublinear regret given sublinear Φ, hence robust. The key ideas of `ExpRb` is to first identify the most vulnerable arms against attacker as a function of selection probabilities; a piece of information that is available to the learner. Then, `ExpRb` constructs a robust estimator that biases (possibly) corrupted reward of the vulnerable arms to mitigate the risk of underestimating the actual reward. Our robust estimator is carefully designed to bias the observed rewards just enough to prevent overestimating the actual reward as well. The impossibility result in Theorem 1 shows that a no-regret algorithm should be attack-aware, which may not be possible in practice. Hence, we adapt a middle-ground approach such that the robustness power of `ExpRb` against corruption is controlled by a robustness parameter γ, which impacts the design of the robust estimator. Last, in §5, we analyze the regret of `ExpRb` and in Theorem 3 and show that if $\gamma = \Phi$, the regret of `ExpRb` is $O(\sqrt{T} + \Phi \log T)$.

2 Preliminaries and Problem Statement

2.1 The Classical Adversarial MAB Problem

The adversarial (or non-stochastic, used interchangeably) MAB problem, initially introduced in [3], is a game in which a learner repeatedly chooses an arm from a set $[K] := \{1, \dots, K\}$ of arms in each round. Let $x_i(t) \in [0, 1]$ denote the reward associated to arm $i \in [K]$ in round t. For each i, the reward sequence $(x_i(t))_{t\in[T]}$ is determined by an adversary before the game starts.[1] At each round $t \in [T]$, the learner chooses an arm $I_t \in [K]$ and receives $x_{I_t}(t)$ as feedback. The objective of the learner is to devise an arm selection algorithm $\mathcal{A}$ maximizing the cumulative rewards over T steps. The performance of the algorithm $\mathcal{A}$ is measured through the notion of pseudo-regret (regret, for short), which is defined as the difference between the cumulative rewards attained by always taking an optimal static decision (in hindsight) and that of $\mathcal{A}$, i.e.,

$$\textsc{Regret}(T, \mathcal{A}) = \max_{i\in[K]} \sum_{t=1}^{T} x_i(t) - \mathbb{E}\left[\sum_{t=1}^{T} x_{I_t}(t)\right], \tag{1}$$

where the expectation is taken with respect to possible internal randomizations of $\mathcal{A}$. The `Exp3` algorithm [3] is the first proposed algorithm achieving a regret of $O(\sqrt{KT\log(K)})$ for the classical adversarial bandit problem described above, and whose advent has led to several other learning strategies with improved regret bounds or applicable to more general settings; see, e.g., [1, 2] and references in [24]. In the following, we introduce a new extended model in which an attacker sits in-between the environment and the learner and corrupts the reward of the selected arm.

2.2 Adversarial Bandits with Corruptions

Consider an adversarial bandit problem, where an adversary and an attacker with more powerful ability to manipulate the reward coexist. Similarly to the classical adversarial bandit described above,

[1] Some literature consider *loss formulation* of adversarial bandits, where the learner receives a loss $\ell_i(t) \in [0, 1]$ upon choosing arm i in round t. Here we consider the reward formulation. We note however that most results for reward formulation can be translated to the corresponding loss formulation via the relation $\ell_i(t) = 1 - x_i(t)$; see [5].

the adversary determines the reward in an arbitrary way prior to the first round. In runtime, after the learner commits to an arm, the attacker is able to corrupt the reward of the selected arm I_t, and the learner receives the corrupted reward. Specifically speaking, the attacker manipulates the reward $x_{I_t}(t)$ of the selected arm I_t into

$$\tilde{x}_{I_t}(t) = x_{I_t}(t) - a(t), \qquad a(t) \in [x_{I_t}(t) - 1, x_{I_t}(t)], \tag{2}$$

where $a(t)$ is the *injected corruption* (or corruption, for short) at round t. Note that the feasible range of corruption at round t implies $\tilde{x}_{I_t}(t) \in [0, 1]$. The learner receives $\tilde{x}_{I_t}(t)$ without knowing the original reward $x_{I_t}(t)$ or the corruption $a(t)$.

The value of $a(t)$ in Eq. (2) determines the design space of the attacker in each round to mislead the learner to end up with selecting a suboptimal arm. However, we assume that the attacker is endowed with a predetermined corruption budget. Let $\Phi(T)$ represent the budget of the attacker, so that cumulative exerted corruption (magnitude-wise) over all rounds must statisfy $\sum_{t=1}^{T} |a(t)| \leq \Phi(T)$. We further refer to such an attacker as a $\Phi(T)$-*attacker*. Clearly, the performance of algorithms degrades more for larger values of $\Phi(T)$. Hereafter, we denote $\Phi := \Phi(T)$ for brevity.

In the following definition, we formally characterize the notion of robustness of a bandit algorithm against corruptions.

Definition 1 *An algorithm $\mathcal{A}$ is said to be Φ-robust if $\textsc{Regret}(T, \mathcal{A}) = \widetilde{O}(\sqrt{T} + \Phi)$ against any Φ-attacker, where the $\widetilde{O}(\cdot)$ notation hides multiplicative terms that are poly-logarithmic in T.*

We finally turn to introducing the notion of regret for the adversarial bandits with corruptions. The regret of the algorithm $\mathcal{A}$ is defined as

$$\textsc{Regret}(T, \mathcal{A}) = \max_{i \in [K]} \sum_{t=1}^{T} x_i(t) - \mathbb{E}\left[\sum_{t=1}^{T} \tilde{x}_{I_t}(t)\right], \tag{3}$$

where the second term in the right-hand side corresponds to the expected return in terms of corrupted values. We remark that it is plausible to consider a slightly different version of the attack model, which only changes the *observation* of the learner without changing the *actual* accrued reward. In this case, the definition of regret coincides with that in Eq. (1). Our regret analysis for the notion of regret in Eq. (3) could be straightforwardly applied to that of Eq. (1). Details in Remark 5.1 in §5. Unless stated otherwise, the term "regret" in this paper refers to the notion formalized in Eq. (3).

Remark 2.1 *We mention that there is growing literature on oblivious attack models for stochastic bandit problems; see, e.g., [18, 9]. These papers target at a middle ground of a mixed stochastic and adversarial model that aim to achieve the best of both worlds. Different from these works, our work focuses on targeted attack models for adversarial bandits, since an oblivious attacker can be intrinsically captured in the basic setting of adversarial bandits.*

Remark 2.2 *There is a rich literature on non-stochastic bandits with adaptive adversaries [24], where the adversary is able to see the past actions of the learner and determines the reward right before the current action. The attacker in our model is more powerful than an adaptive adversary, since it observes the action of the learner and perturbs the reward before revealing it to the learner.*

3 Vulnerability and Regret Lower Bound

In this section, we present a regret lower bound for *attack-agnostic* algorithms, i.e., algorithms that are unaware of the existence of an attacker.

We begin with the following theorem establishing a linear regret for *attack-agnostic* algorithms against a Φ-attacker with $\Phi = o(T)$:

Theorem 1 *Consider an attack-agnostic bandit algorithm $\mathcal{A}$ satistfying the following property: For any two-armed problem instance, the expected regret of $\mathcal{A}$ is $O(\sqrt{T})$. Then, for any $\varepsilon \in (0, \frac{1}{8})$, there exists a Φ-attacker with $\Phi = O(T^{1/2+2\epsilon})$ such that the regret of $\mathcal{A}$ (without knowing the attack) is $\Omega(T^{1-\varepsilon})$ with high probability.*

The above theorem demonstrates an impossibility result for attack-agnostic bandit algorithms to achieve a sublinear regret. We stress that this result is applicable to stochastic MABs with targeted corruptions as well. We however stress that Theorem 1 has no conflict with the results in [18, 9] where robust corruption-agnostic algorithms designed for stochastic MABs with *oblivious* corruption. In fact, the proof of this theorem, provided in §B in the supplementary, constructs an instance of a stochastic bandit problem and considers the setting that the reward on each arm is subject to a fixed and unknown distribution. In order to attain a sublinear regret, the learning algorithm can only sample a "sub-optimal" arm for sublinear number of times. Otherwise, the learning algorithm fails to attain a sublinear regret even without attacks. Thus, the attacker can mislead the algorithm by manipulating the reward on the optimal arm for sublinear number of times. Consequently, the optimal arm is sampled for only sublinear number of times, and the regret of any attack-agnostic bandit algorithm can thus be made arbitrarily close to linear (by choosing small enough ε).

As a concrete example, in the following we show that the classic `Exp3` algorithm[2] cannot achieve a sublinear regret against an $O(\sqrt{T})$-corrupted attacker.

Corollary 2 (Vulnerability of `Exp3`) $\varepsilon \in (0, \frac{1}{8})$. *There exists a* Φ*-attacker with* $\Phi = O(T^{1/2+2\epsilon})$ *such that the regret under* `Exp3` *is* $\Omega(T^{1-\varepsilon})$.

Theorem 1 demonstrates that to develop a robust algorithm for non-stochastic bandits with corruptions, it is inevitable to provide the algorithm with the information of the existence of the attacker. We call these algorithms attack-aware algorithms. However, it remains open whether the knowledge of the attacker's budget is necessary to attain a sublinear regret.

4 The `ExpRb` Algorithm

In this section, we propose `ExpRb`, a bandit algorithm that is robust to corruption from a targeted attacker. The logical flow of `ExpRb` follows the rationality of the `Exp3` algorithm with an additional novel biased estimator to make the algorithm robust against corruption. In round $t \in [T]$, `ExpRb` draws arm I_t according to the following distribution

$$p_i(t) = (1-\eta)\frac{w_i(t-1)}{\sum_{j=1}^{K} w_j(t-1)} + \frac{\eta}{K}, \quad i \in [K], \tag{4}$$

which is a weighted combination (parameterized by $\eta \in (0, 1]$) of a uniform distribution and a weighted distribution determined by the weights $w_i(t-1)$ maintained for each arm. The weight parameter $w_i(t)$ is defined for each arm with initial values of 1. The intuition behind selecting this mixed distribution is to make sure that all arms are chosen [3].

Once the algorithm selects arm I_t the estimated reward is calculated as follows.

$$\hat{x}_i(t) = \mathbb{1}_{\{I_t=i\}}\frac{\tilde{x}_i(t) + \delta(t)}{p_i(t)}, \quad i \in [K], \tag{5}$$

where $\mathbb{1}_A$ denotes the indicator function of an event A, and where $\delta(t)$ is a compensate variable explained in details in §4.1. Finally, the algorithm updates the weight of the various arms as

$$w_i(t) = w_i(t-1)\exp\left(\eta\hat{x}_i(t)/K\right), \quad i \in [K]. \tag{6}$$

In the next section, we explain the details of the robust estimator as the key novelty of the `ExpRb` algorithm.

4.1 Robust Estimator and Intuitions

Once the arm I_t is selected the main step of `ExpRb` toward robustification of the observed reward $\tilde{x}_{I_t}(t)$ begins. The high-level idea of robustification is two-fold: (i) we introduce a compensate variable $\delta(t)$ to augment the estimated reward of the selected arm and mitigate the risks of underestimation and overestimation of the actual reward; and (ii) we introduce a robustness parameter γ that could be tuned based on the budget of the attacker, to determine the design space of learner in biasing the estimated reward.

[2] We refer the reader to [3] for the detailed explanation of the `Exp3` algorithm. The `Exp3` algorithm, however, could be recovered from Algorithm 1 in this paper by simply setting $\tilde{x}_i(t) = x_i(t)$ and $\delta(t) = 0$ for all i, t.

Algorithm 1 The `ExpRb` Algorithm

1: **Initialization:** $\eta \in (0, 1]$, robustness parameter γ, $w_i(0) = 1$ and $q_i(0) = 1$ for all $i \in [K]$
2: **for** $t = 1$ to T **do**
3: Set

$$p_i(t) = (1 - \eta)\frac{w_i(t-1)}{\sum_{j=1}^{K} w_j(t-1)} + \frac{\eta}{K}, \quad i \in [K]$$

4: Draw arm I_t randomly according to the probabilities $p_1(t), \ldots, p_K(t)$
5: Observe reward $\tilde{x}_{I_t}(t)$
6: Set $\delta(t) = 0$
7: **if** $p_{I_t}(t) < q_{I_t}(t-1)$ **then**
8: Set $\delta(t) = \min\left\{\gamma\left(1 - p_{I_t}(t)/q_{I_t}(t-1)\right), 1\right\}$
9: Update $q_{I_t}(t) = \begin{cases} \max\left\{p_i(t), (1 - 1/\gamma)q_i(t-1)\right\} & i = I_t \\ q_i(t-1) & i \neq I_t \end{cases}$
10: **end if**
11: Set the reward estimates

$$\hat{x}_i(t) = \mathbb{1}_{\{I_t = i\}}\frac{\tilde{x}_i(t) + \delta(t)}{p_i(t)}, \quad i \in [K]$$

12: Update the weights

$$w_i(t) = w_i(t-1)\exp\left(\eta\hat{x}_i(t)/K\right), \quad i \in [K]$$

13: **end for**

Now, we proceed to explain the details of the robust estimator. As Eq. (5) indicates, if $p_{I_t}(t)$, the selection probability of the selected arm I_t, is small, the attacker is able to greatly impact the estimated reward of I_t with small corruption. In other words, when the selection probability for the selected arm is small, the required budget for the attacker to trick the learning algorithm to "underestimate" the arm is also small. This leads us to set the value of compensate variable as a function of selection probability. However, the learner should be able to track the historical evolution of compensate variable for each arm to prevent "overestimation" of the corruption. Hence, we initiate an auxiliary variable $q_i(0) = 1, i \in [K]$, to record the smallest selection probability of each arm (if chosen) so far. The value of compensate variable is set as follows.

$$\delta(t) = \min\left\{\gamma\left(1 - \frac{p_{I_t}(t)}{q_{I_t}(t-1)}\right), 1\right\}. \tag{7}$$

The algorithmic nuggets of setting the compensate variable are as follows: (i) as in Line 7 of `ExpRb`, $\delta(t)$ is set only when $p_{I_t}(t) < q_{I_t}(t-1)$, since otherwise, the algorithm has already biased the estimated reward of I_t in previous rounds; (ii) $\delta(t)$ is capped to at most 1, since the value of $a(t)$, i.e., the attacker's corruption, is at most 1; (iii) $\delta(t)$ is a function of γ that determines how much bias is required; γ has a direct relationship to the budget of attacker, i.e., the greater the budget of the attacker, the greater the robustness parameter γ; and last (iv) the larger the difference between $p_{I_t}(t)$ and $q_{I_t}(t-1)$, the greater the $\delta(t)$. And finally, we update $q_i, i \in [K]$ to either $p_{I_t}(t)$ (once the first term in Eq. (7) is active) or the value of $p_{I_t}(t)$ at which $\gamma\left(1 - p_{I_t}(t)/q_{I_t}(t-1)\right) = 1$, representing the second term in Eq. (7) in which $\delta(t) = 1$. More compactly, we have

$$q_{I_t}(t) = \max\left\{p_{I_t}(t), \left(1 - 1/\gamma\right)q_{I_t}(t-1)\right\}. \tag{8}$$

The running time of `ExpRb` is similar to `Exp3` which is $O(K)$. The pseudocode of `ExpRb` is summarized as Algorithm 1.[3] Last, it is worth noting that the idea of compensate variable (a.k.a. biased estimator) has been used for a variety of reasons in the non-stochastic bandits, e.g., in `Exp3.P` [3] and `Exp3.IX` [20] the idea of *biased reward-estimates* is leveraged to achieve improved high-probability regret bounds for non-stochastic bandits. Although the high-level idea of "robust estimator" is the same, our design in this work is to make the algorithm robust against corruption.

[3] In the paper, the algorithm is presented with fixed parameters with respect to the length of time horizon. One can extend the proposed algorithm to the anytime version by using the doubling trick policy [3].

Remark 4.1 *We remark that [31] presents the* `Tsallis-INF` *algorithm for the so-called 'best of both worlds' setting.* `Tsallis-INF` *is shown to be robust to adversarial corruptions not only in stochastic bandits but also in a class of adversarial bandits with* stochastically-constrained adversaries*; we refer to Corollary 8 in [31] for the corresponding regret bound of* `Tsallis-INF` *for such adversarial bandits with corruptions. As such,* `Tsallis-INF` *is guaranteed to achieve a sublinear regret in a restricted class of adversarial problems with corruptions. In contrast to [31], in this paper we consider adversarial bandits with corruptions with no such restrictions. However, we would like to note that when applied to the adversarial bandits with corruptions with stochastically-constrained adversaries,* `Tsallis-INF` *is expected to attain a tighter regret bound and without requring the knowledge of* Φ.

5 Regret Analysis

Finally, we analyze the regret of `ExpRb`, and specifically demonstrate it matches the lower bound (up to a logarithmic factor) for the case where the corruption budget is upper bounded.

5.1 Summary and Highlights of the Results

The main result is summarized in the following theorem.

Theorem 3 *The regret under* `ExpRb`*, when it is run with parameters* $\gamma = \Phi$ *and* $\eta = O(\sqrt{(K \log K)/T})$*, satisfies*

$$\textsc{Regret}(T, \texttt{ExpRb}) \leq O\left(\sqrt{K \log KT} + K\Phi \log T\right). \tag{9}$$

The above theorem asserts that the regret upper bound of `ExpRb` scales as $\widetilde{O}(\sqrt{T} + \Phi)$. In view of Definition 1, this implies that `ExpRb` is Φ-robust.

Remark 5.1 *The result in Theorem 3 uses the modified definition of regret in Eq.* (3)*, where the attacker corrupts the actual reward observed by the learner. However, this result can be straightforwardly translated to the original definition of regret in Eq.* (1)*, where the attacker only manipulates the observations of the learner (i.e., feedback), not her actually accrued rewards. A closer look reveals that the difference between the notions of regret in Eq.* (1) *and* (3) *is always upper bounded by* Φ*, which does not dominate the regret upper bound of Theorem 3.*

In the following, we proceed to highlight the key steps to prove the regret result in Theorem 3.

5.2 Regret Analysis of `ExpRb`

The full proof of the theorem appears in §C of the supplementary material. We split the regret analysis of `ExpRb` into two parts. First, we analyze the properties of the robust estimator of `ExpRb` as a function of the robustness parameter γ. These properties then is further applied to analyze the regret of `ExpRb` with respect to γ and Φ.

Recall that the robustness parameter γ impacts the amount of compensate variable $\delta(t)$ in `ExpRb`. We first characterize an upper bound on the cumulative amount of compensate variable with respect to γ in Lemma 4. This result could be interpreted as an upper bound on "overestimation" of rewards. Then, in Lemma 5, we derive a lower bound on the difference between the expected value of the cumulative estimated rewards of arms in `ExpRb` and the actual rewards of the arms. This result could be represented as a lower bound on the "underestimation" of rewards.

The following lemma provides an upper bound on the cumulative compensate variable $\delta(t)$:

Lemma 4 *Under* `ExpRb`*, we have:* $\sum_{t=1}^{T} \delta(t) \leq \gamma K \log(K/\eta)$.

This result provides an upper bound for the cumulative value of compensate variable, which is $O(\gamma \log(1/\eta))$. The proof of this result follows by re-expressing the value of $\delta(t)$ as a function of

γ and the auxiliary parameter q_i, and then applying straightforward calculus to derive the bound. Details in §C in the supplementary.

The following result characterizes the performance of the robust estimator as a function of γ and ϕ.

Lemma 5 *When* `ExpRb` *is run with* $\gamma \geq \Phi$ *against a* Φ*-attacker, we have:*

$$\sum_{t=1}^{T} \hat{x}_i(t) \geq \sum_{t=1}^{T} x_i/p_i(t), \quad \forall i \in [K].$$

This implies that when $\gamma \geq \Phi$, i.e., the robustness parameter is large enough to be able to compensate the corruption, the estimator can effectively avoid underestimation, thus guaranteeing that $\sum_{t=1}^{T} \mathbb{E}[\hat{x}_i(t)] \geq \sum_{t=1}^{T} x_i(t)$.

We are ready to sketch the proof of Theorem 3. The proof of Theorem 3 follows similar steps as in the proof of `Exp3` in [3]. We stress, however, that the proof here relies on more involved steps as one has to take into account the impact of compensate variable $\delta(t)$ on the final regret. By applying similar analysis for the proof of `Exp3`, we have

$$\mathbb{E}\left[\sum_{t=1}^{T} \hat{x}_i(t)\right] - \mathbb{E}\left[\sum_{t=1}^{T} \tilde{x}_{I_t}(t)\right] \leq (e^2-1)\eta T + \frac{K \log K}{\eta} + \mathbb{E}\left[\sum_{t=1}^{T} \delta(t)\right], \; i \in [K].$$

Compared to the basic setting, our algorithm introduces an additional term $\sum_{t=1}^{T} \mathbb{E}[\delta(t)]$, which corresponds to the long-term sum of the compensate variable. Lemma 4 implies that the sum of the compensate variable is upper bounded by $O(\gamma K \log(K/\eta))$. In addition, in Lemma 5, we have characterized an upper bound on the difference between the cumulative reward $\sum_{t=1}^{T} x_i(t)$ and the estimation $\sum_{t=1}^{T} \mathbb{E}[\hat{x}_i(t)]$. Finally, applying the upper bounds in Lemma 5 concludes the proof of Theorem 3. A detailed proof is given in §C in the supplementary.

6 Concluding Remarks

Motivated by the recent interests in making the online learning algorithms robust against manipulation attacks, this paper studied non-stochastic multi-armed bandit problems with targeted corruptions. It first showed that under targeted corruptions, existing attack-agnostic algorithms for non-stochastic bandits, e.g., `Exp3`, are vulnerable against targeted corruptions with limited budget, and fail to achieve a sublinear regret. Second, it proposed `ExpRb`, as a robust algorithm against targeted corruptions and characterized its regret as a function of a parameter that determines the robustness budget of the algorithm against targeted corruptions. The regret analysis shows that if the corruption budget is sublinear and `ExpRb` is aware of this budget, it achieves a sublinear regret. While there are several recent studies that focus on stochastic MAB problems with corruptions, to the best of our knowledge, this paper is the first that tackles non-stochastic MABs with targeted corruptions.

7 Broader Impacts

Our work fits within the broad direction of research concerning safety issues in AI/ML at large. With the recent radical advances in machine learning, ML-assisted decision making is fast becoming an intrinsic part of the design of systems and services that billions of people around the world use every day. And not surprisingly, investigating the vulnerability of existing learning models and robustness against manipulation attacks are becoming critically important in the light of *trustworthy learning paradigm*. Hence, there has been a surge of interest in making learning models that are robust against adversarial attacks for both applied ML such as supervised learning and deep learning, and theoretical ML such as reinforcement learning and multi-armed bandits. This is critically important for society, since the ML algorithms are being adopted more and more in safety-critical domains across sciences, businesses, and governments that impact people's daily lives. Last, we see no ethical concerns related to this paper.

Acknowledgments and Disclosure of Funding

Lin Yang and Wing Shing Wong acknowledge the support from Schneider Electric, Lenovo Group (China) Limited and the Hong Kong Innovation and Technology Fund (ITS/066/17FP) under the HKUST-MIT Research Alliance Consortium. Mohammad Hajiesmaili's research is supported by NSF CNS-1908298. The work of John C.S. Lui is supported in part by the GRF 14201819. Sadegh Talebi's research is supported by Department of Computer Science, University of Copenhagen.

References

[1] J.-Y. Audibert and S. Bubeck. Minimax policies for adversarial and stochastic bandits. In *Conference on Learning Theory*, pages 217–226, 2009.

[2] J.-Y. Audibert and S. Bubeck. Regret bounds and minimax policies under partial monitoring. *Journal of Machine Learning Research*, 11(Oct):2785–2836, 2010.

[3] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

[4] B. Awerbuch and R. D. Kleinberg. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 45–53, 2004.

[5] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[6] G. V. Cormack. Email spam filtering: A systematic review. *Foundations and Trends® in Information Retrieval*, 1(4):335–455, 2008.

[7] E. Even-Dar, S. M. Kakade, and Y. Mansour. Online Markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009.

[8] Z. Feng, D. Parkes, and H. Xu. The intrinsic robustness of stochastic bandits to strategic manipulation. In *International Conference on Machine Learning*, pages 3092–3101. PMLR, 2020.

[9] A. Gupta, T. Koren, and K. Talwar. Better algorithms for stochastic bandits with adversarial corruptions. In *Conference on Learning Theory*, pages 1562–1578, 2019.

[10] A. György, T. Linder, and G. Ottucsak. The shortest path problem under partial monitoring. In G. Lugosi and H. U. Simon, editors, *Learning Theory*, volume 4005 of *Lecture Notes in Computer Science*, pages 468–482. Springer Berlin Heidelberg, 2006.

[11] A. Heydari, M. ali Tavakoli, N. Salim, and Z. Heydari. Detection of review spam: A survey. *Expert Systems with Applications*, 42(7):3634–3642, 2015.

[12] K.-S. Jun, L. Li, Y. Ma, and J. Zhu. Adversarial attacks on stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 3640–3649, 2018.

[13] W. Z. Khan, M. K. Khan, F. T. B. Muhaya, M. Y. Aalsalem, and H.-C. Chao. A comprehensive study of email spam botnet detection. *IEEE Communications Surveys & Tutorials*, 17(4).

[14] T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

[15] Y. Li, E. Y. Lou, and L. Shan. Stochastic linear optimization with adversarial corruption. *arXiv preprint arXiv:1909.02109*, 2019.

[16] F. Liu and N. Shroff. Data poisoning attacks on stochastic bandits. In *International Conference on Machine Learning*, pages 4042–4050, 2019.

[17] M. Luca and G. Zervas. Fake it till you make it: Reputation, competition, and yelp review fraud. *Management Science*, 62(12):3412–3427, 2016.

[18] T. Lykouris, V. Mirrokni, and R. Paes Leme. Stochastic bandits robust to adversarial corruptions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 114–122, 2018.

[19] Y. Ma, K.-S. Jun, L. Li, and X. Zhu. Data poisoning attacks in contextual bandits. In *International Conference on Decision and Game Theory for Security*, pages 186–204. Springer, 2018.

[20] G. Neu. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. *Advances in Neural Information Processing Systems*, 28:3168–3176, 2015.

[21] G. Neu, A. Gyorgy, and C. Szepesvári. The adversarial stochastic shortest path problem with unknown transition probabilities. In *Artificial Intelligence and Statistics*, pages 805–813, 2012.

[22] P. Ozisik and P. S. Thomas. Security analysis of safe and seldonian reinforcement learning algorithms. In *Advances in Neural Information Processing Systems*, 2020.

[23] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

[24] A. Slivkins. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019.

[25] M. S. Talebi, Z. Zou, R. Combes, A. Proutiere, and M. Johansson. Stochastic online shortest path routing: The value of feedback. *IEEE Transactions on Automatic Control*, 63(4):915–930, 2017.

[26] K. C. Wilbur and Y. Zhu. Click fraud. *Marketing Science*, 28(2):293–308, 2009.

[27] X. Wu, Y. Dong, J. Tao, C. Huang, and N. V. Chawla. Reliable fake review detection via modeling temporal and behavioral patterns. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 494–499. IEEE, 2017.

[28] X. Zhang and X. Zhu. Online data poisoning attack. *arXiv preprint arXiv:1903.01666*, 2019.

[29] P. Zhou, J. Xu, W. Wang, Y. Hu, D. O. Wu, and S. Ji. Toward optimal adaptive online shortest path routing with acceleration under jamming attack. *IEEE/ACM Transactions on Networking*, 27(5):1815–1829, 2019.

[30] J. Zimmert and Y. Seldin. An optimal algorithm for stochastic and adversarial bandits. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 467–475. PMLR, 2019.

[31] J. Zimmert and Y. Seldin. Tsallis-INF: An optimal algorithm for stochastic and adversarial bandits. *arXiv preprint arXiv:1807.07623v4*, 2020.

A Related Work

The basic MAB problems have been extensively extended to several other settings. Our literature review, however, is centered on MABs with corruptions. The existing literature on MAB with adversarial corruptions could be categorized based on the corruption model into two categories of *oblivious* and *targeted* corruption models. Further the existing literature could be categorized into those work that study the *vulnerability* of existing algorithms versus those that develop *robust* algorithms against corruptions. Based on these four criteria, Table 1 summarizes the settings of the existing work and this work. In short, the majority of the existing works focus on either oblivious or targeted corruptions for stochastic MAB problems, and this work, to the best of our knowledge, is the first that studies corruption models for non-stochastic bandits.

A.1 MAB Problems with Oblivious Corruptions

In the oblivious corruption model, an attacker, *oblivious* to the behavior of the bandit algorithm, corrupts the stochastic patterns of some arms in each round. Specifically, this corruption model targets stochastic bandit problems in which the reward of each arm follows a stochastic distribution. The goal of the attacker is to adversarially manipulate the rewards of some arms to trick the algorithm to choose sub-optimal arms. This model targets a middle ground of a mixed stochastic and adversarial model that aims to achieve the best of both worlds. The oblivious corruption model is intrinsically captured in the basic setting of non-stochastic MAB, since the adversary determines the reward in adversarial manner, however, oblivious to the learner's algorithm [3, 5]. In the following, then, we focus on reviewing the related works on stochastic MABs with oblivious corruptions.

Ma *et al.* [19] introduced an attack framework based on a convex optimization formulation that shows by slightly manipulation of the rewards, existing MAB algorithms are highly vulnerable against oblivious corruption models. In [16], the framework has been extended to develop attack strategies to a broad range of stochastic bandit algorithms. Both works, however, focus on designing attack strategies to show the vulnerability of existing algorithms.

In another category [9, 18], the goal is to develop robust algorithms against oblivious corruptions. The high-level idea is to expand the confidence bounds of the existing algorithms to be robust against manipulation attacks on rewards. This setting was first proposed by Lykouris et al. [18] and a sublinear regret algorithm with respect to the corruption budget was proposed. Specifically, the proposed algorithm in [18] achieves the regret of $\tilde{O}(KG\sum_{i\neq i^*} 1/\Delta_i)$, where K is the number of arms, G is the corruption budget, i^* is the optimal arm, and Δ_i is the gap between μ^*, the expected reward of the optimal arm and μ_i, the expected reward of arm i, i.e., $\Delta_i = \mu^* - \mu_i$, and notation $\tilde{O}$ suppresses all dependence on logarithmic terms. This bound is $O(KG)$ times worse than the standard bound achievable by existing algorithms like `UCB` in uncorrupted setting. This result has been improved to an algorithm with the regret of $O(KG) + \tilde{O}(\sum_{i\neq i^*} 1/\Delta_i)$ in [9]. That is, the new algorithm in [9] attains a regret bound which removes the multiplicative dependence on G in [18] and replace it with an additive term. When the corruption is more powerful, i.e., larger G, the reward pattern is more like that of the adversarial model, thereby the performance of the online algorithm is expected to be degraded to fully non-stochastic setting. Last, Zimmert and Seldin [30] study the problem of optimal algorithms for stochastic and adversarial bandits that includes [18, 9] as special case.

A.2 MAB Problems with Targeted Corruptions

In the targeted corruption model, which is mainly the focus of this paper, *the adversary sits in-between the environment and the learner, observes the selected arm by the learner, corrupts its reward, and the learner just observes the corrupted reward.* That means the corruption policy targets the action of the player, and hence, the corruption is more powerful than the oblivious corruption model. Different from the previous setting, this corruption model could be considered in both stochastic and non-stochastic models.

The prior work in this direction [12, 16] studied the vulnerability of existing stochastic MAB algorithms against targeted corruptions. The authors in [12] design specific targeted attacks with logarithmic budget that hijack two popular stochastic bandit algorithms, i.e., ϵ-greedy and `UCB` algorithms, by failing to achieve sublinear regret. A more comprehensive vulnerability study is

conducted in [16] where a targeted corruption strategy is proposed that can hijack any stochastic bandit algorithm without knowing the bandit algorithm.

Our work, to the best of our knowledge, is the first that focuses on non-stochastic bandits with targeted corruptions. Similar to [12, 16], it investigates the vulnerability of existing bandit algorithms, however, different from [12, 16] for non-stochastic setting, e.g., `Exp3`. Similar to [9, 18], it develops a robust algorithm, called `ExpRb` for corrupted bandits, however, different form [9, 18] for non-stochastic setting. Last, our analysis on vulnerability is applicable to both stochastic and non-stochastic bandit algorithms.

B Proof of Theorem 1

We consider a two-armed bandit problem with Bernoulli arms with means $(\mu_1, \mu_2) = (\frac{1}{2}, \frac{1}{2} + \Delta)$, for some Δ that we specify later. The rewards of each arm are i.i.d. and the rewards are independent across arms.

Consider a learning algorithm $\mathcal{A}$. Further, consider an attacker, which adds a randomly generated noise with mean -2Δ to the rewards of the second arm (i.e., the optimal arm) whenever it is selected by the algorithm. Thus, from the learner's view, arm 2 has mean $1/2 - \Delta$. We assume that the adversary has enough budget to do so over T rounds. The algorithm is unaware of the attacker's existence.

Let N_1 and N_2 denote the number of pulls of arm 1 and arm 2 after T rounds, respectively. Thus, $N_1 + N_2 = T$. The regret on the corrupted problem is $R(T) = \Delta\mathbb{E}[N_2]$. Since we assume that $\mathcal{A}$ attains a regret of at most $O(\sqrt{T})$ for any T, we have $\Delta\mathbb{E}[N_2] \leq O(\sqrt{T})$, or $\mathbb{E}[N_2] \leq O(\sqrt{T})/\Delta$. Hence, using $N_1 + N_2 = T$, we get

$$\mathbb{E}[N_1] \geq T - \frac{O(\sqrt{T})}{\Delta},$$

and thus, the regret of $\mathcal{A}$ on the corrupted problem is at least

$$R(T) = \Delta\mathbb{E}[N_1] \geq \Delta T - \frac{O(\sqrt{T})}{\Delta}.$$

Next we find a high probability upper bound on the budget Φ of the attacker. Observe that $\Phi < N_2$, so we need to find a high probability upper bound on N_2. For $X > 0$, we have:

$$X\mathbb{P}(N_2 > X) \leq \mathbb{E}[N_2] \leq \frac{O(\sqrt{T})}{\Delta}$$

Hence, $\mathbb{P}(N_2 < X) \geq 1 - \frac{O(\sqrt{T})}{X\Delta}$ and thus, $\mathbb{P}(\Phi < X) \geq 1 - O(\sqrt{T})/(X\Delta)$. Now choosing $\Delta = T^{-\varepsilon}$, $\varepsilon \in (0, \frac{1}{8})$ yields

$$R(T) = \Delta\mathbb{E}[N_1] \geq \Delta T - \frac{O(\sqrt{T})}{\Delta} = T^{1-\varepsilon} - O(T^{1/2+\varepsilon}).$$

Hence, $R(T) = \Omega(T^{1-\varepsilon})$. Furthermore, choose $X = T^{1/2+2\varepsilon}$. Then, with high probability, $\Phi \leq O(T^{1/2+2\varepsilon})$. □

C Regret Analysis of `ExpRb`: Proof of Theorem 3

Let $T > 1$. For any arm i, we let $\mathcal{T}_i \subseteq [T]$ denote the set of time slots where arm i is selected and the selection probability for arm i is lower than all previous ones:

$$\mathcal{T}_i = \left\{ t \in [T] : I_t = i \text{ and } p_i(t) \leq \min_{t' < t : I_{t'} = i} p_i(t') \right\}.$$

We denote the size of $\mathcal{T}_i$ by N_i. Alternatively, we may write $\mathcal{T}_i = \{t_i(n), n \in [N_i]\}$. Note that $t_i(n)$, $n = 1, 2, \ldots, N_i$ correspond to the time slots that arm i is selected by `ExpRb` and the maintained probability is updated.

We first provide the following lemmas, which we prove later:

Lemma 6 *For all $j \in [K]$,*

$$(1-\eta)\sum_{t=1}^{T}\hat{x}_j(t) \le \sum_{t=1}^{T}\tilde{x}_{I_t}(t) + \sum_{t=1}^{T}\delta(t) + \frac{\eta}{K}\sum_{t=1}^{T}\sum_{i=1}^{K}\hat{x}_i(t) + \frac{K\log K}{\eta}$$

Lemma 4 (restated) *We have: $\sum_{t=1}^{T}\delta(t) \le \gamma K\log(K/\eta)$.*

Using the above lemmas and taking expectations, we get

$$\begin{aligned}(1-\eta)\sum_{t=1}^{T}\mathbb{E}[\hat{x}_j(t)] &\le \sum_{t=1}^{T}\mathbb{E}[\tilde{x}_{I_t}(t)] + \gamma K\log\frac{K}{\eta} + \frac{\eta}{K}\sum_{t=1}^{T}\sum_{i=1}^{K}\mathbb{E}[\hat{x}_i(t)] + \frac{K\log K}{\eta}\\ &= \sum_{t=1}^{T}\mathbb{E}[\tilde{x}_{I_t}(t)] + \gamma K\log\frac{K}{\eta} + 2\eta T + \frac{K\log K}{\eta},\end{aligned}$$

where we used the fact that

$$\mathbb{E}[\hat{x}_j(t)] = \mathbb{E}[\mathbb{E}[p_{I_t}(t)(\tilde{x}_{I_t}(t)+\delta(t))/p_{I_t}(t)|\mathcal{F}_{t-1}]] \le 2\,,$$

where $\mathcal{F}_{t-1}$ denotes the history of the game up to time slot t. We will be using the following lemma to simplify the left-hand side of the inequality:

Lemma 5 (restated) *For all $i \in [K]$, and $\gamma \ge \Phi$, we have: $\sum_{t=1}^{T}\hat{x}_i(t) \ge \sum_{t=1}^{T}x_i(t)/p_i(t)$.*

We therefore obtain:

$$\max_j \sum_{t=1}^{T}x_j(t) - \sum_{t=1}^{T}\mathbb{E}[\tilde{x}_{I_t}(t)] \le \gamma K\log\frac{K}{\eta} + 3\eta T + \frac{K\log K}{\eta}\,.$$

Finally, the proof is completed by setting $\gamma = \Phi$ and $\eta = O(\sqrt{K\log K/T})$. □

C.1 Proof of Lemma 6

For $t \ge 1$, denote $W_t := \sum_{i=1}^{K} w_i(t)$. We derive upper and lower bounds on $\log\frac{W_{T+1}}{W_1}$.

Lower Bound. Note that $W_1 = K$. Then,

$$\log\frac{W_{T+1}}{W_1} \ge \log\frac{w_j(T+1)}{K} = \frac{\eta}{K}\sum_{t=1}^{T}\hat{x}_j(t) - \log K\,, \tag{10}$$

where $j \in [K]$ is arbitrary.

Upper Bound. First observe that $\log\frac{W_{T+1}}{W_1} = \sum_{t=1}^{T}\log\frac{W_{t+1}}{W_t}$. Next we derive an upper bound on W_{t+1}/W_t. We have:

$$\begin{aligned}\frac{W_{t+1}}{W_t} &= \sum_{i=1}^{K}\frac{w_i(t)}{W_t}e^{\eta\hat{x}_i(t)/K}\\ &\le \sum_{i=1}^{K}\Big(\frac{p_i(t)-\eta/K}{1-\eta}\Big)\Big(1+\frac{\eta}{K}\hat{x}_i(t)+\frac{\eta^2}{2K^2}\hat{x}_i(t)^2\Big)\\ &= \underbrace{\sum_{i=1}^{K}\frac{p_i(t)-\eta/K}{1-\eta}}_{=1} + \frac{\eta}{K(1-\eta)}\sum_{i=1}^{K}p_i(t)\hat{x}_i(t) + \frac{\eta^2}{2K^2(1-\eta)}\sum_{i=1}^{K}p_i(t)\hat{x}_i(t)^2\end{aligned}$$

where in the second line, we have used the inequality $e^z \le 1+z+\frac{1}{2}z^2$ valid for all $z > 0$.

Note that $\sum_{i=1}^{K}p_i(t)\hat{x}_i(t) = \tilde{x}_{I_t}(t) + \delta(t)$ and

$$\sum_{i=1}^{K}p_i(t)\hat{x}_i(t)^2 = p_{I_t}(t)\hat{x}_{I_t}(t)^2 = (\tilde{x}_{I_t}(t)+\delta(t))\hat{x}_{I_t}(t) \le 2\hat{x}_{I_t}(t) = 2\sum_{i=1}^{K}\hat{x}_i(t)$$

Hence,

$$\frac{W_{t+1}}{W_t} \leq 1 + \frac{\eta}{K(1-\eta)}(\tilde{x}_{I_t}(t) + \delta(t)) + \frac{\eta^2}{K^2(1-\eta)} \sum_{i=1}^{K} \hat{x}_i(t)$$

Taking logarithim from both sides and using the inequality $\log(1+z) \leq z$ valid for all $z > -1$, we obtain:

$$\log \frac{W_{t+1}}{W_t} \leq \frac{\eta}{K(1-\eta)}(\tilde{x}_{I_t}(t) + \delta(t)) + \frac{\eta^2}{K^2(1-\eta)} \sum_{i=1}^{K} \hat{x}_i(t)\,,$$

which further gives:

$$\log \frac{W_{T+1}}{W_1} = \sum_{t=1}^{T} \log \frac{W_{t+1}}{W_t} \leq \frac{\eta}{K(1-\eta)} \sum_{t=1}^{T} \tilde{x}_{I_t}(t) + \frac{\eta}{K(1-\eta)} \sum_{t=1}^{T} \delta(t) + \frac{\eta^2}{K^2(1-\eta)} \sum_{t=1}^{T} \sum_{i=1}^{K} \hat{x}_i(t)\,.$$

Putting the upper and lower bounds together, we obtain: For all $j \in [K]$,

$$\frac{\eta}{K} \sum_{t=1}^{T} \hat{x}_j(t) - \log K \leq \frac{\eta}{K(1-\eta)} \sum_{t=1}^{T} \tilde{x}_{I_t}(t) + \frac{\eta}{K(1-\eta)} \sum_{t=1}^{T} \delta(t) + \frac{\eta^2}{K^2(1-\eta)} \sum_{t=1}^{T} \sum_{i=1}^{K} \hat{x}_i(t)$$

which concludes the proof. □

C.2 Proof of Lemma 4

By the design of `ExpRb`, the value of $\delta(t)$ is set to a non-zero value only when the current selection probability of the selected arm, i.e., $p_{I_t}(t)$ is smaller than $q_{I_t}(t-1)$. Fix an arm $i \in [K]$, and consider time slots $t_i(n)$, $n = 1, 2, \ldots, N_i$, where i is selected. We can show that

$$\delta(t_i(n)) = \gamma \left(1 - \frac{q_i(t_i(n))}{q_i(t_i(n-1))}\right), \quad n = 1, 2, \ldots, N_i. \tag{11}$$

To prove this claim, we consider all possible cases a time slot $t_i(n)$ as follows:

Case (i): $p_i(t) \geq q_i(t_i(n-1))$**.** In this case, $q_i(t_i(n))$ will be set to $q_i(t_i(n-1))$. Then, the value of $\delta(t_i(n))$ from Eq. (11) will be 0, complying with Line 6 of `ExpRb`.

Case (ii): $q_i(t_i(n-1)) \geq p_i(t) \geq (1-1/\gamma)q_{I_t}(t_i(n-1))$**.** Here, $q_i(t_i(n))$ will be set to $p_i(t_i(n))$. Based on Eq. (11), the value of $\delta(t_i(n))$ will be set to $\gamma\,(1 - p_i(t_i(n))/q_i(t_i(n-1)))$. This case complies with Eq. (7), since $p_i(t)$ satisfies $\gamma\,(1 - p_{I_t}(t)/q_{I_t}(t_i(n-1))) \leq 1$.

Case (iii): $p_i(t) < (1-1/\gamma)q_{I_t}(t_i(n-1))$**.** In this case, $q_i(t_i(n))$ will be set to $(1-1/\gamma)q_{I_t}(t_i(n-1))$. In this case, $\delta(t_i(n))$ based on Eq. (11) will be equal to 1. Moreover, when $p_i(t) < (1-1/\gamma)q_{I_t}(t_i(n-1))$, we have $\gamma\,(1 - p_{I_t}(t)/q_{I_t}) > 1$, which implies that $\delta(t_i(n))$ complies with Eq. (7).

Putting these together proves the claim in Eq. (11).

$$\begin{aligned}
\sum_{n\in[N_i]} \delta(t_i(n)) &= \sum_{n\in[N_i]} \gamma \left(1 - \frac{q_i(t_i(n))}{q_i(t_i(n-1))}\right) \\
&= \sum_{n\in[N_i]} \gamma \frac{1}{q_i(t_i(n-1))} \big(q_i(t_i(n-1)) - q_i(t_i(n))\big) \\
&\leq -\gamma \int_{q_i(t_i(1))}^{q_i(t_i(N_i))} \frac{1}{z} dz = -\gamma \log z \Big|_1^{q_i(t_i(N_i))} = -\gamma \log q_i(t_i(N_i))\,.
\end{aligned}$$

Moreover, by the design of `ExpRb`, $p_i(t) \geq \eta/K$ for all i and t, which further implies $q_i(t_i(N_i)) \geq \eta/K$. Hence,

$$\sum_{t\in[T]} \delta(t) = \sum_{i=1}^{K} \sum_{n\in[N_i]} \delta(t_i(n)) \leq \gamma K \log(K/\eta),$$

thus completing the proof. □

C.3 Proof of Lemma 5

Let $i \in [K]$. Due to using compensate variables, the estimation on arm i at time slot t will be increased by $\delta(t)/p_i(t)$. Specifically, by the design of the algorithm, we have

$$\sum_{t\in\mathcal{T}_i} \frac{\delta(t)}{p_i(t)} = \sum_{n\in[N_i]} \frac{1}{p_i(t_i(n))} \left[\min\left\{1, \gamma\left(1 - \frac{p_i(t_i(n))}{q_i(t_i(n-1))}\right)\right\}\right].$$

To further simplify the above equation, we consider the following two possibilities for a time slot $t_i(n), i \in [N_i]$:

(i) If $\gamma\left(1 - \frac{p_i(t_i(n))}{q_i(t_i(n-1))}\right) \leq 1$, then $q_i(t_i(n)) = p_i(t_i(n))$ (see Eq. ((8))) and

$$\begin{aligned}\frac{1}{p_i(t_i(n))}\left[\min\left\{1, \gamma\left(1 - \frac{p_i(t_i(n))}{q_i(t_i(n-1))}\right)\right\}\right] &= \frac{\gamma}{q_i(t_i(n))}\left(1 - \frac{q_i(t_i(n))}{q_i(t_i(n-1))}\right) \\ &= \frac{1}{q_i(t_i(n))}\gamma\left(1 - \frac{q_i(t_i(n))}{q_i(t_i(n-1))}\right) + \left(\frac{1}{p_i(t_i(n))} - \frac{1}{q_i(t_i(n))}\right).\end{aligned}$$

(ii) If $\gamma\left(1 - \frac{p_i(t_i(n))}{q_i(t_i(n-1))}\right) > 1$, according to Eq. ((8)), we have $q_i(t_i(n)) = (\gamma-1)/\gamma q_i(t_i(n-1))$, so that $\gamma\left(1 - \frac{q_i(t_i(n))}{q_i(t_i(n-1))}\right) = 1$. Hence,

$$\begin{aligned}\frac{1}{p_i(t_i(n))}\left[\min\left\{1, \gamma\left(1 - \frac{p_i(t_i(n))}{q_i(t_i(n-1))}\right)\right\}\right] &= \frac{1}{p_i(t_i(n))} \\ =&\frac{1}{q_i(t_i(n))} + \frac{1}{p_i(t_i(n))} - \frac{1}{q_i(t_i(n))} \\ =&\frac{\gamma}{q_i(t_i(n))}\left(1 - \frac{q_i(t_i(n))}{q_i(t_i(n-1))}\right) + \left(\frac{1}{p_i(t_i(n))} - \frac{1}{q_i(t_i(n))}\right).\end{aligned}$$

Putting together both cases yields

$$\sum_{t\in\mathcal{T}_i} \frac{\delta(t)}{p_i(t)} = \sum_{n\in[N_i]} \frac{\gamma}{q_i(t_i(n))}\left(1 - \frac{q_i(t_i(n))}{q_i(t_i(n-1))}\right) + \sum_{n\in[N_i]}\left(\frac{1}{p_i(t_i(n))} - \frac{1}{q_i(t_i(n))}\right) \tag{12}$$

Then, we have

$$\begin{aligned}\sum_{t\in[T]} \hat{x}_i(t) &= \sum_{t\in\mathcal{T}_i} \frac{\tilde{x}_i(t) + \delta(t)}{p_i(t)} \\ &= \sum_{t\in\mathcal{T}_i} \frac{\tilde{x}_i(t)}{p_i(t)} + \sum_{n\in[N_i]} \frac{\gamma}{q_i(t_i(n))}\left(1 - \frac{q_i(t_i(n))}{q_i(t_i(n-1))}\right) + \sum_{n\in[N_i]}\left(\frac{1}{p_i(t_i(n))} - \frac{1}{q_i(t_i(n))}\right) \\ &= \sum_{t\in\mathcal{T}_i} \frac{\tilde{x}_i(t)}{p_i(t)} + \sum_{n\in[N_i]} \gamma\left(\frac{1}{q_i(t_i(n))} - \frac{1}{q_i(t_i(n-1))}\right) + \sum_{n\in[N_i]}\left(\frac{1}{p_i(t_i(n))} - \frac{1}{q_i(t_i(n))}\right) \\ &= \sum_{t\in\mathcal{T}_i} \frac{\tilde{x}_i(t)}{p_i(t)} + \frac{\gamma}{q_i(t_i(N_i))} + \sum_{n\in[N_i]}\left(\frac{1}{p_i(t_i(n))} - \frac{1}{q_i(t_i(n))}\right).\end{aligned} \tag{13}$$

Now, assuming $\gamma \geq \Phi$, we have

$$\begin{aligned}\sum_{t\in[T]} \hat{x}_i(t) &= \sum_{t\in\mathcal{T}_i} \frac{\tilde{x}_i(t)}{p_i(t)} + \frac{\gamma}{q_i(t_i(N_i))} + \sum_{n\in[N_i]}\left(\frac{1}{p_i(t_i(n))} - \frac{1}{q_i(t_i(n))}\right) \\ &= \sum_{t\in\mathcal{T}_i} \frac{\tilde{x}_i(t)}{p_i(t)} + \frac{\gamma - \Phi}{q_i(t_i(N_i))} + \frac{\Phi}{q_i(t_i(N_i))} + \sum_{n\in[N_i]}\left(\frac{1}{p_i(t_i(n))} - \frac{1}{q_i(t_i(n))}\right) \\ &\geq \sum_{t\in\mathcal{T}_i} \frac{\tilde{x}_i(t)}{p_i(t)} + \frac{\gamma - \Phi}{q_i(t_i(N_i))} + \frac{1}{q_i(t_i(N_i))}\sum_{t\in\mathcal{T}_i}|a(t)| + \sum_{n\in[N_i]}\left(\frac{1}{p_i(t_i(n))} - \frac{1}{q_i(t_i(n))}\right).\end{aligned}$$

Using $q_i(t_i(N_i)) \leq q_i(t)$ for any t, and rewriting some terms in the above equation, we have

$$\sum_{t\in[T]} \hat{x}_i(t) \geq \sum_{t\in\mathcal{T}_i} \frac{\tilde{x}_i(t)}{p_i(t)} + \frac{\gamma-\Phi}{q_i(t_i(N_i))} + \sum_{t\in\mathcal{T}_i} \frac{|a(t)|}{q_i(t)} + \sum_{t\in\mathcal{T}_i} \left(\frac{1}{p_i(t)} - \frac{1}{q_i(t)}\right). \tag{14}$$

In view of $0 \leq |a(t)| \leq 1$, the last two terms in the right-hand side satisfy

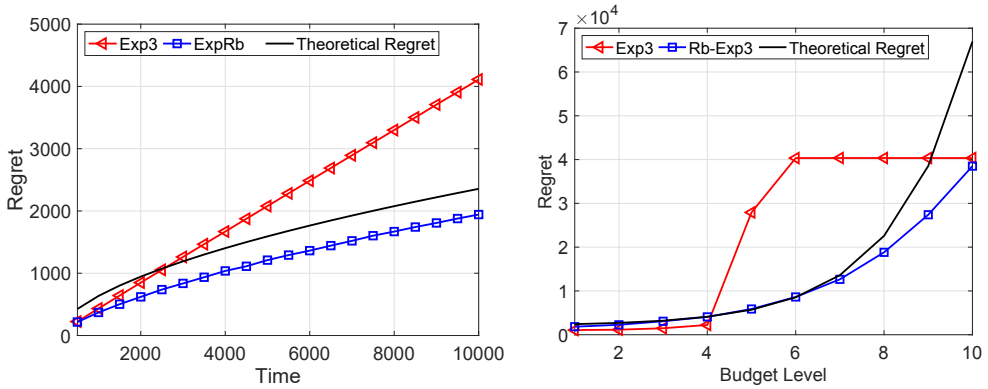
$$\begin{aligned}\sum_{t\in\mathcal{T}_i} \frac{|a(t)|}{q_i(t)} + \sum_{t\in\mathcal{T}_i} \left(\frac{1}{p_i(t)} - \frac{1}{q_i(t)}\right) &\geq \sum_{t\in\mathcal{T}_i} \frac{|a(t)|}{q_i(t)} + \sum_{t\in\mathcal{T}_i} |a(t)| \left(\frac{1}{p_i(t)} - \frac{1}{q_i(t)}\right) \\ &= \sum_{t\in\mathcal{T}_i} \frac{|a(t)|}{p_i(t)} \geq \sum_{t\in\mathcal{T}_i} \frac{a(t)}{p_i(t)}\end{aligned}$$

Putting this together with the fact that $q_i(t_i(N_i)) \leq 1/K$, we thus the desired result:

$$\sum_{t\in[T]} \hat{x}_i(t) \geq \sum_{t\in\mathcal{T}_i} \frac{\tilde{x}_i(t)}{p_i(t)} + \frac{\gamma-\Phi}{q_i(t_i(N_i))} + \sum_{t\in\mathcal{T}_i} \frac{a(t)}{p_i(t)} \geq \sum_{t\in\mathcal{T}_i} \frac{x_i(t)}{p_i(t)} + (\gamma-\Phi)K\,.$$

□

D Numerical Results



(a) Empirical Regret of `ExpRb` and `Exp3` Under Attacks

(b) Performance Comparison Under Different Corruption Levels

Figure 1: Experimental results

In the simulation, we evaluate the performance of `ExpRb` and compare it with the `Exp3` algorithm in different scenarios. In order to evaluate our algorithm under adversarial corruptions, we assume the attacker follows the so-called attack-optimal-arms policy introduced in Section 2. The attack-optimal-arms policy can efficiently attack the empirical reward estimation of the optimal arm and trick the learning algorithm to select a suboptimal arm.

The experimental scenario is as follows. Consider an environment with one high-reward arm and $K-1$ low-reward arms. The attacker aims to decrease the observed reward when the online learner chooses the optimal arm. When the budget is available, the attacker will always set the reward on the optimal arm to be zero when it was chosen. We report the average regret obtained by collecting the actual regret of 100 execution of this scenario. The first scenario involves an attacker with attack budget of $O(\sqrt{T})$. The simulation results shown in Figure 1(a) imply that the performance of the `Exp3` algorithm is largely degraded with the attack. The `ExpRb` algorithm, however, achieves a sublinear regret. In the second scenario, we compare the performance of two algorithms under different amount of budget of attacker. Toward this, we vary the available budget of the attacker in 10 levels. The corresponding budget for the l-th level is $T^{0.2+l/20}$. Figure 1(b) shows the performance comparison between the `Exp3` algorithm and the `ExpRb` algorithm. One can find that the performance of `Exp3` is largely degraded when the attacker budget reaches $T^{1/2}$, while the `ExpRb` algorithm can tolerate heavier attacks.