Adversarial Bandits with Corruptions: Regret Lower Bound and No-regret Algorithm

Anonymous Author(s)
Affiliation
Address
email

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` as a function of the corruption budget and show that for the case of a known corruption budget, the regret of `ExpRb` is tight.

1 Introduction

Multi-armed bandits (MABs) [21] 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., [4, 22] and references therein. In the basic MAB problem, a learner repeatedly chooses an action (or pulls an arm) in each round, and observes the reward associated to the selected arm, but not for other unselected arms. The goal of the learner is to maximize the long-term reward collected. MAB problems are typically categorized into stochastic and non-stochastic (or adversarial) categories depending on how the reward sequences are generated. In stochastic bandits [13, 21], 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 application such as click fraud via malware [23], fake reviews and ratings in recommender systems [10, 16, 24], and email spam [6, 12], there have been recent effort on studying bandit problems under corruption [8, 9, 11, 14, 15, 17, 18, 25]. In the 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 effort, however, are limited to studying stochastic bandits with corruption, either on understanding the vulnerability of existing algorithms and designing attacks [8, 11, 14, 15, 18, 25], or developing robust algorithms against corruption [9, 17, 26]. In those works, stochastic patterns are corrupted by an attacker and bandit algorithms aim to be robust against the corruption. A detailed literature review is in §A of the supplementary material.

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

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.* [17]	✓		✓	✓			
Gupta *et al.* [9]	✓		✓	✓			
Jun *et al.* [11]		✓	✓				
Liu *et al.* [15]	✓	✓	✓				
This work		✓	✓		✓	✓	✓

In contrast, this paper is the first, to the best of our knowledge, that studies *non-stochastic bandits with corruptions*. This problem is interesting with unique challenges different from stochastic bandits with corruptions and calls for non-trivial algorithm design and regret analysis. In addition, in some application domains such as shortest path routing [20] and inventory control problem [7], the reward functions are very complex to model using stochastic bandits, hence, from a practical perspective, non-stochastic bandits are relevant for such intrinsically involved applications. Consequently, studying the vulnerability and robustness of non-stochastic bandit algorithms with corruption becomes important as well. We formally define the model in the following.

The Corruption Model. Consider a K-armed non-stocahstic bandit, where as in [3], 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 of the learner, and corrupts its rewards aiming to mislead the learner to take 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, 17], 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*, the attacker in this model manipulates the reward adaptively to the learner's chosen arm; hence, the attacker is different from adaptive adversary in which the rewards is determined right before learner's action. We refer to this attack as *targeted*. In contrast, the prior literature on stochastic bandits with corruption [9, 17] 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 and budget of the attacker as *attach-agnostic* algorithms, and *attack-aware* algorithms know the attacker and its budget.

1.1 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.

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. This has not conflict to the attack-agnostic algorithms in [9, 17] that develop sublinear algorithms for oblivious attackers. Second, we derive a regret lower bound (Theorem 3) for attack-aware algorithms for non-stochastic bandits with corruption as a function of corruption budget Φ. Informally, our results shows that the regret of any attack-aware bandit algorithm is lower bounded by $\Omega(\sqrt{T} + \Phi)$.

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 4 and show that if $\gamma = \Phi$, the regret of ExpRb is $O(\sqrt{T} + \Phi \log(\sqrt{T}))$, hence the performance of attack-aware ExpRb matches the regret lower bound in Theorem 3, up to logarithmic factors. Said succinctly, while with sublinear Φ, attack-aware ExpRb enjoys a sublinear regret, it is general enough to work even if Φ is not known thanks to the robustness parameter γ.

2 Preliminaries and Problem Statement

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, 5, 26] and references in [22]. 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.

The 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, 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 $\sum_{t=1}^{T} |a(t)| \le \Phi(T)$. 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, for brevity, we drop T from $\Phi(T)$ and denote it by Φ.

[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 [4].

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 to stochastic bandit problems; see, e.g., [9, 17]). 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 non-stochastic 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 adversary [22] where the adversary is able to see the past actions of the learner and determine the reward right before the current action. The attacker in our model is more powerful than the adaptive adversary, since it observers the action of the learner and perturbs it before revealing to the learner.*

3 Vulnerability and Regret Lower Bound

In this section, we present regret lower bounds for the adversarial MAB with corruption problem described above. In particular, we first establish that an *attack-agnostic* algorithm, i.e., the algorithm that is not aware of the existence and budget of the attacker, may incur a regret growing linearly in T. We then provide a lower bound on the regret of any *attack-aware* algorithms, i.e., those that are aware of the existence and budget of 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 *For any attack-agnostic algorithm $\mathcal{A}$, there exist some $\alpha \in (0, 1)$ and a Φ-attacker with $\Phi = O(T^{1-\alpha})$ such that the regret of $\mathcal{A}$ (without knowing the attack) is $\Omega(T)$.*

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 [9, 17] where robust corruption-agnostic algorithms designed for stochastic MABs with *oblivious* corruption. In fact, the proof of this theorem, provided in §C 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 is thus $\Omega(T)$.

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) *There exists a Φ-attacker with $\Phi = O(\sqrt{T})$ such that* $\textsc{Regret}(T, \texttt{Exp3}) = \Omega(T)$.

A detailed construction of the attack policy of the targeted attacker for Exp3 is provided in §D of the supplementary. Our vulnerability analysis is based on a simple attack policy as follows. Consider *attack-optimal-arms* that simply corrupt the reward of the optimal arm(s) once an optimal arm is chosen. Specifically, once the learner chooses an optimal arm, the attacker decreases the corresponding reward while respecting the budget Φ. In this way, the attacker degrades empirical observations of the learner for the optimal arm(s) so as to mislead her towards selecting suboptimal ones. Using the attack-optimal-arms policy, we construct a case where Exp3 selects suboptimal arms for $O(\sqrt{T})$ rounds, which is the regret of the Exp3 algorithm. Thus, to mislead Exp3 to select a suboptimal arm, the attacker only needs to manipulate the rewards on any optimal arm for $O(\sqrt{T})$ times.

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 and budget of attacker. We call these algorithms attack-aware algorithms. Next, the following result provides a lower bound on the regret of any attack-aware algorithm for non-stochastic bandits with a Φ-corrupted attacker.

Theorem 3 *Let $T \in \mathbb{N}$ and $\Phi > 0$. For any attack-aware algorithm $\mathcal{A}$ (i.e., being aware of Φ), there exist an instance of adversarial MAB with corruption and a Φ-attacker such that the regret under $\mathcal{A}$ after T steps is $\Omega(\sqrt{T} + \Phi)$.*

The lower bound above implies that to achieve a sublinear regret, it is inevitable to have $\Phi = o(T)$, i.e., limit the corruption budget of the attacker. This is indeed aligned with the robust algorithm design for stochastic bandits with corruptions [9, 17]. We report a brief sketch for the proof of Theorem 3, leaving details to §E in the supplementary. First recall that in the absence of attacks, the regret of any non-stochastic bandit algorithms is lower bounded by $O(\sqrt{T})$ [3]. Thus, for any algorithm, we can construct a sequence of rewards after manipulating, denoted by $\tilde{x}_i(t)$, such that $\max_i \sum_{t\in[T]} \tilde{x}_i(t) - \sum_{t\in[T]} \tilde{x}_{I_t}(t) \geq \Omega(\sqrt{T})$. In addition, it is possible to consider attacking the optimal arm(s) only. We thus have $\max_i \sum_{t\in[T]} x_i(t) - \max_i \sum_{t\in[T]} \tilde{x}_i(t) = \Phi$. Substituting this into the definition of regret gives the desired lower bound.

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.

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

[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, $w_i(0) = 1, \forall i \in [K]$, and $\tilde{p}_i = 1, i \in [K]$
2: **for** $t = 1$ to T **do**
3: Draw arm I_t randomly according to Eq. (4)
4: Observe reward $\tilde{x}_{I_t}(t)$ and set the biased estimator $\delta(t) = 0$
5: **if** $p_{I_t}(t) < \tilde{p}_{I_t}$ **then**
6: Set $\delta(t)$ based on Eq. (7)
7: Update $\tilde{p}_{I_t}$ based on Eq. (8)
8: **end if**
9: Set the reward estimate based on Eq. (5)
10: Update the weights based on Eq. (6)
11: **end for**

where $\delta(t)$ is a biased estimator that is explained in details in §4.1. Finally the algorithm updates after the weight of 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.

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 $\tilde{p}_i = 1, i \in [K]$, to record the smallest selection probability of each arm (if chosen) so far. The value of compensate variable is set as follows.

$$\delta(t) = \min\left\{\gamma\left(1 - p_{I_t}(t)/\tilde{p}_{I_t}\right), 1\right\}. \tag{7}$$

The algorithmic nuggets of setting the compensate variable are as follows: (i) as in Line 5 of ExpRb, $\delta(t)$ is set only when $p_{I_t}(t) < \tilde{p}_{I_t}$, 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 $\tilde{p}_{I_t}$, the greater the $\delta(t)$. And finally, we update $\tilde{p}_{I_t}$ 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 $\gamma\left(1 - p_{I_t}(t)/\tilde{p}_{I_t}\right) = 1$, representing the second term in Eq. (7) in which $\delta(t) = 1$. More compactly, we have

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

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 [19] 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].

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 4 *The regret under* ExpRb*, when it is run with parameters* $\gamma = \Phi$ *and* $\eta = \sqrt{\frac{K \log K}{(e^2-1)T}}$*, satisfies*

$$\textsc{Regret}(T, \texttt{ExpRb}) \leq 2\sqrt{(e^2-1)K \log KT} + \Phi \log\left(\sqrt{\frac{(e^2-1)KT}{\log K}}\right). \tag{9}$$

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. Furthermore, in view of Theorem 3, the regret of ExpRb matches the lower (up to a logarithmic factor). This shows the tightness of our results.

Remark 5.1 *The result in Theorem 4 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 4.*

Remark 5.2 *The statement of Theorem 4 focuses on the case with* $\gamma = \Phi$*. In the supplementary material, we analyze the regret of* ExpRb *with* $\gamma < \Phi$*, and show that it fails to achieve a sub-linear regret. This is in perfect alignment with the impossibility result in Theorem 1. However, with* γ*, the* ExpRb *has more flexibility to determine the level of robustness.*

In the following, we proceed to highlight the key steps to prove the regret result in Theorem 4.

5.2 Regret Analysis of ExpRb

The full proof of the theorem appears in §F 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 5. This result could be interpreted as an upper bound on "overestimation" of rewards. Then, in Lemma 6, 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.

Lemma 5 *The cumulative compensate variable in* ExpRb *satisfies* $\sum_{t=1}^{T} \delta(t) \leq \gamma K \log\left(\frac{K}{\eta}\right)$.

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 $\tilde{p}_i$, and then applying straightforward calculus to derive the bound. Details in §F.1 in the supplementary.

Before stating the next step on a lower bound for the estimated reward, we need to introduce additional notations. Let $\phi := \phi(T)$ denote the cumulative corruption injected by the attacker over T rounds,

i.e.,

$$\phi(T) = \sum_{t=1}^{T} |a(t)| = \sum_{t=1}^{T} |x_{I_t}(t) - \tilde{x}_{I_t}(t)| \,.$$

Obviously, the corruption sequence $(a(t))_{t\in[T]}$ exerted by a Φ-attacker satisfies: $0 \leq \phi \leq \Phi$. We stress that ϕ is determined by the actual realization of the corruption values, and therefore depends on the realized sample path of the game. Let $\mathbb{E}_t[\cdot]$ denote the expectation taken with respect to all the randomization generated by the algorithm up to round t. The following result characterizes the performance of the robust estimator as a function of γ and ϕ.

Lemma 6 *With a ϕ-corrupted attacker, in* ExpRb *we have*

$$\sum\nolimits_{t=1}^{T} \mathbb{E}_t \left\{\hat{x}_i(t)\right\} - \sum\nolimits_{t=1}^{T} x_i(t) \geq c(\gamma - \phi)K, \quad i \in [K],$$

where $c = 1$ if $\gamma \geq \phi$, and $c = 1/\eta$ otherwise.

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 $\sum_{t=1}^{T} \hat{x}_i(t) \geq \sum_{t=1}^{T} x_i(t)$. However, it is also worth noting that, when $\gamma < \phi$, the underestimation is inevitable and it can be as large as $(\gamma - \phi)K/\eta$. This matches our impossibility observation in Theorem 1 stating that no algorithm is able to achieve a sublinear regret when the budget of the attacker is unknown. A proof is given in §**??** in the supplementary.

Now, given the results in Lemmas 5 and 6 on the properties of the compensate variable, we are ready to sketch the proof of Theorem 4. The proof techniques for the regret analysis of ExpRb are similar to those for traditional algorithms. We stress, however, that proof 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 basic non-stochastic bandit problem (one can refer to [3, 4]), 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 $\mathbb{E}\{\sum_{t=1}^{T} \delta(t)\}$, which corresponds to the long-term sum of the compensate variable. Lemma 5 implies that the sum of the compensate variable is upper bounded by $O(\gamma K \log(K/\eta))$. In addition, in Lemma 6, we have characterized an upper bound on the difference between the cumulative reward $\sum_{t=1}^{T} x_i(t)$ and the estimation $\mathbb{E}\left\{\sum_{t=1}^{T} \hat{x}_i(t)\right\}$. Finally, applying the upper bounds in Lemma 6 concludes the proof of Theorem 4. A detailed proof is given in §**??** 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.

References

[1] J.-Y. Audibert and S. Bubeck. Minimax policies for adversarial and stochastic bandits. In *Proc. of COLT*, 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] S. Bubeck, N. Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[5] R. Combes, M. S. T. M. Shahi, A. Proutiere, et al. Combinatorial bandits revisited. In *Proc. of NIPS*, pages 2116–2124, 2015.

[6] G. V. Cormack et al. 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. C. Parkes, and H. Xu. The intrinsic robustness of stochastic bandits to strategic manipulation. *arXiv preprint arXiv:1906.01528*, 2019.

[9] A. Gupta, T. Koren, and K. Talwar. Better algorithms for stochastic bandits with adversarial corruptions. In *Proc. of COLT*, 2019.

[10] 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.

[11] K.-S. Jun, L. Li, Y. Ma, and J. Zhu. Adversarial attacks on stochastic bandits. In *Proc. of NIPS*, pages 3640–3649, 2018.

[12] 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).

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

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

[15] F. Liu and N. Shroff. Data poisoning attacks on stochastic bandits. In *Proc. of ICML*, 2019.

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

[17] T. Lykouris, V. Mirrokni, and R. Paes Leme. Stochastic bandits robust to adversarial corruptions. In *Proc. of ACM STOC*, pages 114–122, 2018.

[18] Y. Ma, K.-S. Jun, L. Li, and X. Zhu. Data poisoning attacks in contextual bandits. In *Proc. of GameSec*, pages 186–204. Springer, 2018.

[19] G. Neu. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. In *Proc. of NIPS*, pages 3168–3176, 2015.

[20] 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.

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

[22] A. Slivkins. Introduction to multi-armed bandits. *arXiv preprint arXiv:1904.07272*, 2019.

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

[24] X. Wu, Y. Dong, J. Tao, C. Huang, and N. V. Chawla. Reliable fake review detection via modeling temporal and behavioral patterns. In *Proc. of IEEE Big Data*, pages 494–499, 2017.

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

[26] J. Zimmert and Y. Seldin. An optimal algorithm for stochastic and adversarial bandits. In *Proc. of AISTATS*, pages 467–475, 2019.

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, 4]. In the following, then, we focus on reviewing the related works on stochastic MABs with oblivious corruptions.

Ma *et al.* [18] 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 [15], 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, 17], 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. [17] and a sublinear regret algorithm with respect to the corruption budget was proposed. Specifically, the proposed algorithm in [17] 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 [17] 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 [26] study the problem of optimal algorithms for stochastic and adversarial bandits that includes [9, 17] 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 [11, 15] studied the vulnerability of existing stochastic MAB algorithms against targeted corruptions. The authors in [11] 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 [15] 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 [11, 15], it investigates the vulnerability of existing bandit algorithms, however, different from [11, 15] for non-stochastic setting, e.g., `Exp3`. Similar to [9, 17], it develops a robust algorithm, called `ExpRb` for corrupted bandits, however, different form [9, 17] for non-stochastic setting. Last, our analysis on vulnerability is applicable to both stochastic and non-stochastic bandit algorithms.

B Additional Notations for Proofs

For the simplicity of analysis, we will use the following notations for the proofs in the Appendix.

By $\mathcal{T}_i \subseteq [T]$, $i = 1, 2, \ldots, K$, we denote the set of time slots that arm i is selected. The size of $\mathcal{T}_i$ is denoted by N_i, and the elements in $\mathcal{T}_i$ are denoted by $t_i(n)$, $n = 1, 2, \ldots, N_i$. Moreover, $t_i(n)$, $n = 1, 2, \ldots, N_i$ correspond to the time slots that arm i is selected by the learning algorithm. Last, considering that the `ExpRb` algorithm maintains a variable $\tilde{p}_i$ which could be updated at each time slot, we define a new notation $\tilde{p}_i(t)$ to represent the value of $\tilde{p}_i$ at the t-th time slot.

C Proof of Theorem 1

Assume there are K arms involved in an instance of MAB problem. We construct the following two input instances which can be selected by the attacker:

1. The rewards on the i-th arm, $i \neq l$, subject to an independent and identical distribution which is unknown to the learner. The mean of the rewards on the those arms are $1/2$. The reward of the l-th arm is always 0. The attacker does nothing on these arms.
2. The rewards on each arm subject to an independent and identical distribution which is unknown to the learner. The mean of the rewards on the l-th arm is 1, and that of the i-th arm, $i \neq l$, is $1/2$. The attacker does not change the rewards on the i-th arm, $i \neq l$, which still subject to the distribution of mean $1/2$. However, the reward on the l-th arm is always manipulated to 0 when it is selected.

According to our setting, the adversary can choose case (1) or case (2) based on the policy of the learner, while the algorithm receives the same observations for case (1) and case (2). That means the algorithm cannot differentiate case (1) and case (2). Assume the algorithm will select the l-th arm $O(T^{1-\beta}), \beta = [0, 1]$, times for cases (1) or (2).

If $\beta = 0$, the adversary can choose case (1). In case (1), the cumulative reward received by the algorithm is at most $0.5(T - O(T^{1-\beta}))$, and that on the best arm is $0.5T$. Thus, the regret of the algorithm is at least $O(T^{1-\beta})$. Hence, the regret of the algorithm is $O(T)$, while $\phi = O(T^0)$, matching the claim.

Now we consider the case $\beta > 0$. The adversary then chooses case (2). In this case, the cumulative reward on the l-th arm is T, and the regret of the algorithm is thus $T - 1/2(T - O(T^{1-\beta})) - O(T^{1-\beta})$. The algorithm then achieves a linear regret with $\beta > 0$. However, the attacker only needs to manipulate the rewards on the l-th arm for $O(T^{1-\beta})$ times. This completes the proof. □

D The Vulnerability of the `Exp3` Algorithm: Proof of Corollary 2

Consider an MAB instance with K arms, in which arm l is a low-reward arm with $x_l(t) = 1/2, t \in [T]$, and the rest $K - 1$ arms are high-reward with $x_i(t) = 1, i \neq l, t \in [T]$. Let $\hat{x}_l(t)$ be the estimation of `Exp3` on the low-reward arm and $p_l(t)$ be the probability of choosing low-reward arms. Further, let $\hat{X}_l(t) = \sum_{\tau=1}^{t} \hat{x}_l(\tau)$ be the aggregate estimate reward of the low-reward arm up to round t, and $W_l(t) = \exp\left(\eta \hat{X}_l(t)/K\right)$ that represents the weight parameter of low-reward arm up

$$
\begin{aligned}
&\sum_{t=1}^{T} p_l(t)\hat{x}_l(t) \\
= \;& (1-\eta)\sum_{t=1}^{T}\frac{W_l(t-1)}{K-1+W_l(t-1)}\hat{x}_l(t) + \frac{\eta}{K}\sum_{t=1}^{T}\hat{x}_l(t) \\
= \;& (1-\eta)\sum_{t=1}^{T}\frac{K-1+W_l(t-1)}{K-1+W_l(t-1)}\hat{x}_l(t) + \frac{\eta}{K}\sum_{t=1}^{T}\hat{x}_l(t) - (1-\eta)\sum_{t=1}^{T}\frac{1}{K-1+W_l(t-1)}\hat{x}_l(t) \\
\geq \;& (1-\eta)\sum_{t=1}^{T}\hat{x}_l(t) + \frac{\eta}{K}\sum_{t=1}^{T}\hat{x}_l(t) - (1-\eta)\int_0^{\hat{X}_l(T)}\frac{1}{K-1+\exp(\eta z)}dz \qquad (11) \\
& -(1-\eta)\eta\sum_{t=1}^{T}\frac{W_l(t-1)}{\left(K-1+W_l(t-1)\right)^2}\hat{x}_l^2(t) \\
\geq \;& (1-\eta)\sum_{t=1}^{T}\hat{x}_l(t) + \frac{\eta}{K}\sum_{t=1}^{T}\hat{x}_l(t) - 1 - (1-\eta)\eta\sum_{t=1}^{T}\frac{W_l(t-1)}{\left(K-1+W_l(t-1)\right)^2}\hat{x}_l^2(t) \\
\geq \;& (1-\eta)\sum_{t=1}^{T}\hat{x}_l(t) + \frac{\eta}{K}\sum_{t=1}^{T}\hat{x}_l(t) - 1 - \eta\sum_{t=1}^{T}\hat{x}_l(t) \\
= \;& \sum_{t=1}^{T}\hat{x}_l(t) - \left(2-\frac{1}{K}\right)\eta\sum_{t=1}^{T}\hat{x}_l(t) - 1. \qquad (12)
\end{aligned}
$$

to round t. By the rules of `Exp3`, we have

$$\sum_{t=1}^{T} p_l(t)\hat{x}_l(t) \geq \sum_{t=1}^{T}\hat{x}_l(t) - \left(2-\frac{1}{K}\right)\eta\sum_{t=1}^{T}\hat{x}_l(t) - 1, \qquad (10)$$

where the detailed derivation is given in Equation (12). In Equation (12), the first inequality uses the fact that for a decreasing and convex function $f(z)$, we have

$$\sum_{t=1}^{T} f(z_{t-1})(z_t - z_{t-1}) \leq \int_z f(z)dz + \sum_{t=1}^{T} f'(z_{t-1})(z_t - z_{t-1})^2.$$

The second inequality is obtained by calculating the integral. The third inequality uses the following:

$$\sum_{t=1}^{T}\frac{W_l(t-1)}{\left(K-1+W_l(t-1)\right)^2}\hat{x}_l^2(t) \leq \sum_{t=1}^{T} p_l(t)\hat{x}_l^2(t) \leq \sum_{t=1}^{T}\hat{x}_l(t)$$

Thus, using the results in Equation (10), the expected reward obtained by `Exp3` is

$$
\begin{aligned}
\mathbb{E}\left\{\sum_{t=1}^{T} p_l(t)\hat{x}_l(t)\right\} \geq & \mathbb{E}\left\{\sum_{t=1}^{T}\hat{x}_l(t)\right\} - \left(2-\frac{1}{K}\right)\eta\mathbb{E}\left\{\sum_{t=1}^{T}\hat{x}_l(t)\right\} - 1 \\
= & \frac{1}{2}T - \left(2-\frac{1}{K}\right)\sqrt{\frac{K\log K}{(e-1)T}}\cdot\frac{1}{2}T - 1.
\end{aligned}
$$

The algorithm attains a reward of $1/2$ when choosing the low-reward arm, and receive 0 when choosing the optimal arm. This means that the regret of `Exp3` is linear with this attack. Thus, the expected number of rounds that `Exp3` chooses the optimal arm is not larger than $\phi = \left(1-\frac{1}{2K}\right)\sqrt{\frac{TK\log K}{(e-1)}} - 1$. Hence, the expected budget of the attacker is $\phi = O(\sqrt{T})$. This shows that the required budget for the attacker to effectively attack `Exp3` is $O(\sqrt{T})$. This completes the proof. □

E Regret Lower Bound: Proof of Theorem 3

It is known that without attacks, the regret of any non-stochastic bandit algorithm is lower bounded by $O(\sqrt{KT})$. In particular, a worst-case instance can be constructed by assigning fixed distributions to each arm (see [3]). Without loss of generality, we can assume that the reward on the optimal arm is less than 1. Based on the above facts, we can construct a sequence of rewards, whose values after manipulating, denoted by $\tilde{x}_i(t)$, satisfy

$$\max_i \sum_{t\in[T]} \tilde{x}_i(t) - \sum_{t\in[T]} \tilde{x}_{I_t}(t) \geq \Omega(\sqrt{KT}).$$

In addition, it is possible to consider attacking the optimal arm(s) only. We thus have

$$\max_i \sum_{t\in[T]} x_i(t) - \max_i \sum_{t\in[T]} \tilde{x}_i(t) = \Phi,$$

where Φ can be from 0 to $O(T)$, since the reward on the optimal arm is less than 1. Plugging this into the definition of regret gives the desired lower bound. □

F Regret Analysis of ExpRb: Proof of Theorem 4

For each $i \in [K]$ and $t \in [T]$, let us define $W_i(t) = \exp\left(\eta \hat{X}_i(t)/K\right)$ with $\hat{X}_i(t) = \sum_{\tau\leq t} \hat{x}_i(\tau)$. Recalling that $p_i(t) = (1-\eta)\frac{w_i(t-1)}{\sum_{j=1}^K w_j(t-1)} + \frac{\eta}{K}$, we have

$$\sum_i p_i(t)\hat{x}_i(t) = \sum_i (1-\eta)\frac{W_i(t-1)}{\sum_j W_j(t-1)}\hat{x}_i(t) + \sum_i \frac{\eta}{K}\hat{x}_i(t)\,.$$

We first provide an upper bound on $\sum_i \frac{W_i(t)}{\sum_j W_j(t-1)}$ by bouning the second term in the right-hand side above from below. To this end, note that $\hat{x}_i(t) \leq 2K/\eta$ since dy definition, $p_i(t) \geq K/\eta$, and the value of $\tilde{x}_i(t) + \delta(t)$ is bounded by 2. Hence, $\frac{\eta}{K}\hat{x}_i(t) \leq 2$. Using the fact $\exp(z) \leq 1 + z + ((e^2-3)z^2)/4$ valid for all $z \geq 2$, we obtain

$$\frac{\eta}{K}\hat{x}_i(t) \leq \exp\left(\eta\hat{x}_i(t)/K\right) - 1 - \frac{c\eta^2}{K^2}\hat{x}_i^2(t)\,,$$

with $c = (e^2-3)/4$. Hence,

$$\begin{aligned}
\sum_i \frac{W_i(t-1)}{\sum_j W_j(t-1)}\hat{x}_i(t) &= \frac{K}{\eta}\sum_i \frac{W_i(t-1)}{\sum_j W_j(t-1)}\Big(\frac{\eta}{K}\hat{x}_i(t)\Big) \\
&\geq \frac{K}{\eta}\sum_i \frac{W_i(t-1)}{\sum_j W_j(t-1)}\Big(\exp\left(\eta\hat{x}_i(t)/K\right) - 1 - \frac{c\eta^2}{K^2}\hat{x}_i^2(t)\Big) \\
&= \frac{K}{\eta}\left(\frac{\sum_i W_i(t)}{\sum_j W_j(t-1)} - 1\right) - \sum_i \frac{W_i(t-1)}{\sum_j W_j(t-1)}\frac{c\eta}{K}\hat{x}_i^2(t) \\
&\geq \frac{K}{\eta}\left(\frac{\sum_i W_i(t)}{\sum_j W_j(t-1)} - 1\right) - \frac{c\eta}{K(1-\eta)}\sum_i p_i(t)\hat{x}_i^2(t)\,,
\end{aligned}$$

where the last inequality follows from $\frac{W_i(t-1)}{\sum_j W_j(t-1)} \leq \frac{p_i(t)}{1-\eta}$.

$$\sum_i p_i(t)\hat{x}_i(t) \geq \frac{K(1-\eta)}{\eta}\left(\frac{\sum_i W_i(t)}{\sum_j W_j(t-1)} - 1\right) - \frac{c\eta}{K}\sum_i p_i(t)\hat{x}_i^2(t) + \sum_i \frac{\eta}{K}\hat{x}_i(t)\,,$$

which further gives

$$\frac{\sum_i W_i(t)}{\sum_j W_j(t-1)} \leq \frac{c\eta^2}{K^2(1-\eta)}\sum_i p_i(t)\hat{x}_i^2(t) + \frac{\eta}{K(1-\eta)}\sum_i p_i(t)\hat{x}_i(t) + 1\,.$$

Taking logarithm from both sides and using the inequality $\log(z+1) \leq z$ give

$$\log \frac{\sum_i W_i(t)}{\sum_j W_j(t-1)} \leq \frac{c\eta^2}{K^2(1-\eta)} \sum_i p_i(t)\hat{x}_i^2(t) + \frac{\eta}{K(1-\eta)} \sum_i p_i(t)\hat{x}_i(t).$$

It then follows that

$$\begin{aligned}
\frac{\eta}{K}\hat{X}_i(T) =& \log W_i(T) \leq \log \sum_i W_i(T) \\
=& \sum_{t=1}^{T} \log \frac{\sum_i W_i(t)}{\sum_j W_j(t-1)} + \log K \\
\leq& \frac{c\eta^2}{K^2(1-\eta)} \sum_{t=1}^{T} \sum_i p_i(t)\hat{x}_i^2(t) + \frac{\eta}{K(1-\eta)} \sum_{t=1}^{T} (\tilde{x}_{I_t}(t) + \delta(t)) + \log K.
\end{aligned}$$

We thus obtain the following upper bound on $\sum_{t=1}^{T} \tilde{x}_{I_t}(t)$:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} \tilde{x}_{I_t}(t)\right] \\
\geq& (1-\eta)\mathbb{E}\left[\sum_{t=1}^{T} \hat{x}_i(t)\right] - \frac{c\eta(1-\eta)}{K}\mathbb{E}\left[\sum_{t=1}^{T}\sum_i p_i(t)\hat{x}_i^2(t)\right] - \frac{(1-\eta)K\log K}{\eta} - \mathbb{E}\left[\sum_{t=1}^{T}\delta(t)\right] \\
\geq& \mathbb{E}\left[\sum_{t=1}^{T} \hat{x}_i(t)\right] - 2\eta T - \frac{c\eta}{K}\mathbb{E}\left[\sum_{t=1}^{T}\sum_i p_i(t)\hat{x}_i^2(t)\right] - \frac{K\log K}{\eta} - \mathbb{E}\left[\sum_{t=1}^{T}\delta(t)\right] \\
=& \mathbb{E}\left[\sum_{t=1}^{T} \hat{x}_i(t)\right] - 2\eta T - \frac{c\eta}{K}\mathbb{E}\left[\sum_{t=1}^{T} (\tilde{x}_{I_t}(t) + \delta(t))\, \hat{x}_{I_t}(t)\right] - \frac{K\log K}{\eta} - \mathbb{E}\left[\sum_{t=1}^{T}\delta(t)\right] \\
\geq& \mathbb{E}\left[\sum_{t=1}^{T} \hat{x}_i(t)\right] - 2\eta T - \frac{2c\eta}{K}\mathbb{E}\left[\sum_{t=1}^{T} \hat{x}_{I_t}(t)\right] - \frac{K\log K}{\eta} - \mathbb{E}\left[\sum_{t=1}^{T}\delta(t)\right],
\end{aligned} \tag{13}$$

In the above equation, the second inequality is based on the fact that $\mathbb{E}\left[\sum_{t=1}^{T} \hat{x}_i(t)\right] \leq 2T$; and the last inequality is based on the fact that $\tilde{x}_{I_t}(t) + \delta(t) \leq 2$. Note that

$$\mathbb{E}\left[\sum_{t=1}^{T} \hat{x}_{I_t}(t)\right] = 2\mathbb{E}\left[\sum_{t=1}^{T}\sum_i \frac{\hat{x}_i(t)}{2}\right] \leq 2KT.$$

Plugging the above equation to Equation (13) yields

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \tilde{x}_{I_t}(t)\right] \geq& \mathbb{E}\left[\sum_{t=1}^{T} \hat{x}_i(t)\right] - 2\eta T - \frac{(e^2-3)\eta}{K} \cdot KT - \frac{K\log K}{\eta} - \mathbb{E}\left[\sum_{t=1}^{T}\delta(t)\right] \\
=& \mathbb{E}\left[\sum_{t=1}^{T} \hat{x}_i(t)\right] - (e^2-1)\eta T - \frac{K\log K}{\eta} - \mathbb{E}\left[\sum_{t=1}^{T}\delta(t)\right].
\end{aligned}$$

We will use the following lemmas (corresponding to Lemmas 5 and 6 in the main text) to further upper bound the above:

Lemma 7 *We have:* $\sum_{t=1}^{T} \delta(t) \leq \gamma K \log\left(\frac{K}{\eta}\right)$.

Lemma 8 *For all* $i \in [K]$, *and* $\gamma \geq \Phi$, *we have:* $\sum_{t\in\mathcal{T}_i} \hat{x}_i(t) \geq \sum_{t\in\mathcal{T}_i} x_i(t)/p_i(t)$.

Applying Lemmas 7 and 8 gives: For all $i \in [K]$,

$$\mathbb{E}\left[\sum_{t=1}^{T} \tilde{x}_{I_t}(t)\right] \geq \sum_{t=1}^{T} x_i(t) - (e^2-1)\eta T - \frac{K\log K}{\eta} - \gamma K \log\left(\frac{K}{\eta}\right).$$

Finally, the proof is completed by taking the maximum over i, and setting $\gamma = \Phi$ and $\eta = \sqrt{\frac{K\log K}{(e^2-1)T}}$. □

F.1 Proof of Lemma 7 (Lemma 5 in the Main Text)

According to the rules of the ExpRb algorithm, the value of $\delta(t)$ is set to a positive only when the current selection probability of the selected arm, *i.e.*, $p_{I_t}(t)$, is smaller than $\tilde{p}_{I_t}(t)$. At a time slot $t_i(n)$, $n = 1, 2, \ldots, N_i$, when the i-th arm is selected, the value of $\delta(t)$ can be represented by the following equation.

$$\delta(t_i(n)) = \gamma\left(1 - \frac{\tilde{p}_i(t_i(n))}{\tilde{p}_i(t_i(n-1))}\right).$$

The correctness of the above equation can be verified by checking the rules of the algorithm. Specifically, we prove the above equation by considering the following cases at time slot $t_i(n)$:

(1) When $p_i(t) \geq \tilde{p}_i(t_i(n-1))$, $\tilde{p}_i(t_i(n))$ will be set equal to $\tilde{p}_i(t_i(n-1))$. Then, the value of the above equation will be equal to 0, complying with the operation in Line 4 in the algorithm.

(2) When $\tilde{p}_i(t_i(n-1)) \geq p_i(t) \geq (1-1/\gamma)\tilde{p}_{I_t}(t_i(n-1))$, the value of $\tilde{p}_i(t_i(n))$ will be set equal to $p_i(t_i(n))$. Based on the above equation, the value of $\delta(t_i(n))$ will be set as $\gamma\left(1 - p_i(t_i(n))/\tilde{p}_i(t_i(n-1))\right)$. This case complies with the operation in Equation (7), since the selection probability $p_i(t)$ satisfies $\gamma\left(1 - p_{I_t}(t)/\tilde{p}_{I_t}(t_i(n-1))\right) \leq 1$.

(3) At last, when $p_i(t) < (1-1/\gamma)\tilde{p}_{I_t}(t_i(n-1))$, the value of $\tilde{p}_i(t_i(n))$ will be set equal to $(1-1/\gamma)\tilde{p}_{I_t}(t_i(n-1))$. In this case, the value of $\delta(t_i(n))$ based on the above equation will be equal to 1. Moreover, when $p_i(t) < (1-1/\gamma)\tilde{p}_{I_t}(t_i(n-1))$, we have $\gamma\left(1 - p_{I_t}(t)/\tilde{p}_{I_t}\right) > 1$, that implies the value of $\delta(t_i(n))$ in this case complies with the operation in Equation 7.

With the above results, then we have

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

In the above equation, the inequality is obtained by considering the equation in second line as a discrete form for the integral of function $1/z$.

Moreover, according to Algorithm 1, the selection probability for any arm is lower bounded by η/K. That is, $\tilde{p}_i(t_i(N_i)) \geq \eta/K$ for $\forall i$. Then, we have

$$\sum_{t\in[T]} \delta(t) = \sum_{i=1}^{K} \sum_{n\in[N_i]} \delta(t_i(n)) \leq -\gamma K \log(z)\,|_1^{\eta/K} = \gamma K \log\left(\frac{K}{\eta}\right),$$

thus completing the proof. □

F.2 Proof of Lemma 8 (Lemma 6 in the Main Text)

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 algoruthm, we have

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

To further simplify the above equation, note that for any round $t_i(n) \in [N_i]$, there are two cases: (i) If $\gamma\left(1-\frac{p_i(t_i(n))}{\tilde{p}_i(t_i(n-1))}\right) \leq 1$, we have

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

where we have used $\tilde{p}_i(t_i(n)) = p_i(t_i(n))$ (see Equation (8)).

(ii) If $\gamma\left(1-\frac{p_i(t_i(n))}{\tilde{p}_i(t_i(n-1))}\right) > 1$, according to Equation (8), we have

$$\tilde{p}_i(t_i(n)) = \frac{\gamma-1}{\gamma}\tilde{p}_i(t_i(n-1)).$$

By basic algebraic operations, we have $\gamma\left(1-\frac{\tilde{p}_i(t_i(n))}{\tilde{p}_i(t_i(n-1))}\right) = 1$, and using this, we have

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

Putting together both cases yields

$$\sum_{t\in\mathcal{T}_i}\frac{\delta(t)}{p_i(t)} = \sum_{n\in[N_i]}\frac{\gamma}{\tilde{p}_i(t_i(n))}\left(1-\frac{\tilde{p}_i(t_i(n))}{\tilde{p}_i(t_i(n-1))}\right)+\sum_{n\in[N_i]}\left(\frac{1}{p_i(t_i(n))}-\frac{1}{\tilde{p}_i(t_i(n))}\right) \tag{14}$$

Then, we have

$$\begin{aligned}\sum_{t\in[T]}\hat{x}_i(t) &= \sum_{t\in\mathcal{T}_i}\frac{\tilde{x}_i(t)+\delta(t)}{p_i(t)} \\ &= \sum_{t\in\mathcal{T}_i}\frac{\tilde{x}_i(t)}{p_i(t)}+\sum_{n\in[N_i]}\frac{\gamma}{\tilde{p}_i(t_i(n))}\left(1-\frac{\tilde{p}_i(t_i(n))}{\tilde{p}_i(t_i(n-1))}\right)+\sum_{n\in[N_i]}\left(\frac{1}{p_i(t_i(n))}-\frac{1}{\tilde{p}_i(t_i(n))}\right) \\ &= \sum_{t\in\mathcal{T}_i}\frac{\tilde{x}_i(t)}{p_i(t)}+\sum_{n\in[N_i]}\gamma\left(\frac{1}{\tilde{p}_i(t_i(n))}-\frac{1}{\tilde{p}_i(t_i(n-1))}\right)+\sum_{n\in[N_i]}\left(\frac{1}{p_i(t_i(n))}-\frac{1}{\tilde{p}_i(t_i(n))}\right) \\ &= \sum_{t\in\mathcal{T}_i}\frac{\tilde{x}_i(t)}{p_i(t)}+\frac{\gamma}{\tilde{p}_i(t_i(N_i))}+\sum_{n\in[N_i]}\left(\frac{1}{p_i(t_i(n))}-\frac{1}{\tilde{p}_i(t_i(n))}\right).\end{aligned} \tag{15}$$

In the following, we proceed to prove the final result case by case.

Case 1: $\gamma \geq \phi$. In this case, we have

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

Using $\tilde{p}_i(t_i(N_i)) \leq \tilde{p}_i(t)$ for any t, and rewriting some terms in the above equation, we have

$$\sum_{t\in[T]} \hat{x}_i(t) \geq \sum_{t\in\mathcal{T}_i} \frac{\tilde{x}_i(t)}{p_i(t)} + \frac{\gamma-\phi}{\tilde{p}_i(t_i(N_i))} + \sum_{t\in\mathcal{T}_i} \frac{|a(t)|}{\tilde{p}_i(t)} + \sum_{t\in\mathcal{T}_i} \left(\frac{1}{p_i(t)} - \frac{1}{\tilde{p}_i(t)}\right). \tag{16}$$

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

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

Putting this together with the fact that $\tilde{p}_i(t_i(N_i)) \leq 1/K$, we thus get

$$\begin{aligned}\sum_{t\in[T]} \hat{x}_i(t) &\geq \sum_{t\in\mathcal{T}_i} \frac{\tilde{x}_i(t)}{p_i(t)} + \frac{\gamma-\phi}{\tilde{p}_i(t_i(N_i))} + \sum_{t\in\mathcal{T}_i} \frac{a(t)}{p_i(t)} \\ &= \sum_{t\in\mathcal{T}_i} \frac{x_i(t)}{p_i(t)} + \frac{\gamma-\phi}{\tilde{p}_i(t_i(N_i))} \\ &\geq \sum_{t\in\mathcal{T}_i} \frac{x_i(t)}{p_i(t)} + (\gamma-\phi)K\,.\end{aligned} \tag{17}$$

Case 2: $\gamma < \phi$. In this case, we have

$$\begin{aligned}\sum_{t\in[T]} \hat{x}_i(t) &= \sum_{t\in\mathcal{T}_i} \frac{\tilde{x}_i(t)}{p_i(t)} + \frac{\gamma}{\tilde{p}_i(t_i(N_i))} + \sum_{n\in[N_i]} \left(\frac{1}{p_i(t_i(n))} - \frac{1}{\tilde{p}_i(t_i(n))}\right) \\ &= \sum_{t\in\mathcal{T}_i} \frac{\tilde{x}_i(t)}{p_i(t)} + \frac{\gamma}{\tilde{p}_i(t_i(N_i))} + \sum_{t\in\mathcal{T}_i} \left(\frac{1}{p_i(t)} - \frac{1}{\tilde{p}_i(t)}\right) \\ &\geq \sum_{t\in\mathcal{T}_i} \frac{\tilde{x}_i(t)}{p_i(t)} + \frac{1}{\tilde{p}_i(t_i(N_i))} \frac{\gamma}{\phi} \sum_{t\in\mathcal{T}_i} |a(t)| + \sum_{t\in\mathcal{T}_i} \left(\frac{1}{p_i(t)} - \frac{1}{\tilde{p}_i(t)}\right) \\ &\geq \sum_{t\in\mathcal{T}_i} \frac{\tilde{x}_i(t)}{p_i(t)} + \frac{\gamma}{\phi} \sum_{t\in\mathcal{T}_i} \frac{|a(t)|}{\tilde{p}_i(t)} + \sum_{t\in\mathcal{T}_i} \left(\frac{1}{p_i(t)} - \frac{1}{\tilde{p}_i(t)}\right),\end{aligned} \tag{18}$$

where the first inequality holds as $\sum_{t\in\mathcal{T}_i} |a(t)| \leq \phi$, and the second one uses the fact that $\tilde{p}_i(t_i(N_i)) \leq \tilde{p}_i(t)$.

Using simple calculations, we obtain

$$\begin{aligned}\frac{\gamma}{\phi} \sum_{t\in\mathcal{T}_i} \frac{|a(t)|}{\tilde{p}_i(t)} + \sum_{t\in\mathcal{T}_i} \left(\frac{1}{p_i(t)} - \frac{1}{\tilde{p}_i(t)}\right) &\geq \frac{\gamma}{\phi} \sum_{t\in\mathcal{T}_i} \frac{|a(t)|}{\tilde{p}_i(t)} + \sum_{t\in\mathcal{T}_i} \left(\frac{|a(t)|}{p_i(t)} - \frac{|a(t)|}{\tilde{p}_i(t)}\right) \\ &= \sum_{t\in\mathcal{T}_i} \frac{|a(t)|}{p_i(t)} + \left(\frac{\gamma}{\phi} - 1\right) \sum_{t\in\mathcal{T}_i} \frac{|a(t)|}{\tilde{p}_i(t)} \\ &\geq \sum_{t\in\mathcal{T}_i} \frac{|a(t)|}{p_i(t)} + \left(\frac{\gamma}{\phi} - 1\right) \sum_{t\in\mathcal{T}_i} \frac{|a(t)|}{p_i(t)} \\ &\geq \sum_{t\in\mathcal{T}_i} \frac{|a(t)|}{p_i(t)} + (\gamma-\phi)\frac{K}{\eta},\end{aligned}$$

where the last inequality is by the fact that the selection probability for any arm at any time slot is lower bounded by η/K. Then, plugging the above result into Equation (18) yields the desired result:

$$\begin{aligned}\sum_{t\in[T]} \hat{x}_i(t) &\geq \sum_{t\in\mathcal{T}_i} \frac{\tilde{x}_i(t)}{p_i(t)} + \sum_{t\in\mathcal{T}_i} \frac{|a(t)|}{p_i(t)} + (\gamma-\phi)\frac{K}{\eta} \\ &\geq \sum_{t\in\mathcal{T}_i} \frac{\tilde{x}_i(t)}{p_i(t)} + \sum_{t\in\mathcal{T}_i} \frac{a(t)}{p_i(t)} + (\gamma-\phi)\frac{K}{\eta} \\ &= \sum_{t\in\mathcal{T}_i} \frac{x_i(t)}{p_i(t)} + (\gamma-\phi)\frac{K}{\eta}.\end{aligned}$$

□