Stochastic Linear Bandits Robust to Adversarial Attacks

Ilija Bogunovic ETH Zürich
Arpan Losalka National Univ. of Singapore
Andreas Krause ETH Zürich
Jonathan Scarlett National Univ. of Singapore

Abstract

We consider a stochastic linear bandit problem in which the rewards are not only subject to random noise, but also adversarial attacks subject to a suitable budget C (i.e., an upper bound on the sum of corruption magnitudes across the time horizon). We provide two variants of a Robust Phased Elimination algorithm, one that knows C and one that does not. Both variants are shown to attain near-optimal regret in the non-corrupted case $C = 0$, while incurring additional additive terms respectively having a linear and quadratic dependency on C in general. We present algorithm-independent lower bounds showing that these additive terms are near-optimal. In addition, in a contextual setting, we revisit a setup of diverse contexts, and show that a simple greedy algorithm is provably robust with a near-optimal additive regret term, despite performing no explicit exploration and not knowing C.

1 Introduction

Over the past years, bandit algorithms have found application in computational advertising, recommender systems, clinical trials, and many more. These algorithms make online decisions by balancing between exploiting previously high-reward actions vs. exploring less known ones that could potentially lead to higher rewards. Bandit problems can roughly be categorized [18] into *stochastic bandits*, in which subsequently played actions yield independent rewards, and *adversarial bandits*, where the rewards are chosen by an adversary, possibly subject to constraints. A recent line of works has sought to reap the benefits of both approaches by studying bandit problems that are stochastic in nature, but with rewards subject to a limited amount of *adversarial corruption*. Various works have developed provably robust algorithms [12, 24, 4, 21], and attacks have been designed that cause standard algorithms to fail [10, 12, 13, 22].

While near-optimal theoretical guarantees have been established in the case of independent arms [12], more general settings remain relatively poorly understood or even entirely unexplored; see Section 1.2 for details. Our primary goal is to bridge these gaps via a detailed study of stochastic *linear* bandits with adversarial corruptions. In the case of a fixed finite (but possibly very large) set of arms, we develop an elimination-based robust algorithm and provide regret bounds with a near-optimal joint dependence on the time horizon and the adversarial attack budget, demonstrating distinct behavior depending on whether the attack budget is known or unknown. In addition, we introduce a novel *contextual* linear bandit setting under adversarial corruptions, and show that under a context diversity assumption, a simple greedy algorithm attains near-optimal regret under adversarial corruptions, despite having no built-in mechanism that explicitly encourages exploration or robustness.

1.1 Problem Setting

We consider the stochastic linear bandit setting with a given set of arms $\mathcal{A}_0 \subset \mathbb{R}^d$ of finite size k, and adversarially corrupted rewards. At each round $t \in \{1, \dots, T\}$:

- The learner chooses an action $A_t \in \mathcal{A}_0$.
- The adversary observes A_t and decides upon the attack/corruption $c_t(A_t)$; in addition, $c_t(\cdot)$ may (implicitly) depend on other problem parameters, as detailed below.
- The learner receives a corrupted reward Y_t:

$$Y_t = \langle \theta, A_t \rangle + \epsilon_t + c_t(A_t), \tag{1}$$

  where $\theta \in \mathbb{R}^d$ is an unknown parameter vector, and $(\epsilon_t)_{t=1}^T$ is a random noise term, which is assumed to be zero-mean and 1-sub-Gaussian.

We assume that the action feature vectors are unique, span $\mathbb{R}^d$, and are bounded, i.e., $\|a\|_2 \leq 1, \forall a \in \mathcal{A}_0$.

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

We similarly make the standard assumption $\|\theta\|_2 \leq 1$, which implies that $|\langle \theta, a \rangle| \leq 1, \forall a \in \mathcal{A}_0$.

We consider an adversary/attacker that has complete knowledge of the problem – it knows both $\mathcal{A}_0$ and θ, and observes both the precise arm pulled and the noise realization ϵ_t before choosing its attack. The total *attack budget* of the adversary is given by $\sum_{t=1}^{T} |c_t(A_t)| \leq C$. We will consider both the cases that C is known and unknown to the learner.

The goal of the learner is to minimize the *cumulative regret*, defined as

$$R_T = \sum_{t=1}^{T} \max_{a \in \mathcal{A}_0} \langle \theta, a - A_t \rangle. \tag{2}$$

Broadly speaking, we say that an algorithm that attains low regret (e.g., sublinear scaling $R_T = o(T)$) is *corruption-tolerant* or *robust to adversarial attacks*.

As noted in [24], one could alternatively count the corruption as being part of the reward and define regret with the corruption included. Both notions are of interest depending on the application (e.g., depending on whether a fake ad click is considered beneficial or not). The two notions differ by at most $O(C)$, whereas our upper bounds will contain at least an $O(C \log T)$ term. In addition, in the multi-armed bandit setting, $\Omega(C)$ lower bounds were shown for both notions in [24].

1.2 Related Work

Recent surveys on bandit algorithms can be found in [18, 28]; here we focus on the most relevant works considering *stochastic* settings with adversarial corruptions and bandit attacks.

Adversarial attacks on standard bandit algorithms (e.g., UCB, ϵ-greedy, and Thompson sampling) were introduced for the case of independent arms (i.e., a classical multi-armed bandit setting) in [13, 22, 23], and for linear bandits in [10]. We will use the latter in our experiments to test robustness of the proposed algorithms, along with other heuristic attacks.

In the case of *independent* arms, Lykouris *et al.* [24] show that a simple elimination algorithm with enlarged confidence bounds is robust and near-optimal when the attack budget C is known. For unknown C, randomized algorithm is given whose regret bound roughly amounts to scaling the uncorrupted regret by C, i.e., multiplicative dependence. Subsequently, Gupta *et al.* [12] gave an improved algorithm whose regret is near-optimal, with an *additive* dependence on C.

Bogunovic *et al.* [4] consider corruption-tolerant bandits for functions with a bounded RKHS norm, which includes linear bandits as a special case. The algorithm of [4] is based on that of [24], and has analogous guarantees. However, even in the case of known C, the best dependence obtained is multiplicative; the possibility of additive dependence was left as an open problem, which we resolve in this work in the linear case.

Li *et al.* [21] also study stochastic linear bandits with adversarial corruptions. A distinction in [21] is that the regret bounds are *instance-dependent*, relying on positive gaps between the function values at corner points of the polyhedral domain. These results are distinct from the *instance-independent* bounds with *a finite number of arms* that we seek in this paper, and neither can be deduced from the other; see [4, App. K] for further discussion, as well as Remark 1 below.

It is worth noting that the above-mentioned works [24, 12, 4, 21] consider a weaker adversary that cannot observe the current action, and this has often also been assumed when designing efficient bandit attacks [13, 22]. Our more powerful adversary has also been considered previously (e.g., see [22, Fig. 2]), and naturally, any given upper bound on regret is stronger the more powerful of an adversary it applies to.

In Appendix F, we discuss further existing works that are less directly related to ours compared to those above, including distinct adversarial settings (e.g., handled by the EXP2 and EXP3 algorithms), "best of both worlds" results for stochastic and adversarial bandits, model mismatch and misspecification, and fractional/Huber-like contamination models.

Remark 1. *Returning to the results in [21], one may note that instance-dependent bounds can potentially be transferred to instance-independent bounds. However, we show in Appendix G that doing this for the results in [21] would at best lead to $R_T = O(T^{2/3} + \sqrt{CT})$, which is strictly higher than than our analogous result (Theorem 2) whenever $C = o(T^{1/3})$. This is despite the fact that we are considering a stronger adversary. However, it should be kept in mind that the domains adopted are different (polyhedral vs. finite), posing another hurdle that would need to be overcome to transfer results from one setting to the other.*

1.3 Contributions

Our main contributions are as follows:

- For known C, we present a Robust Phased Elimination algorithm, and show that it recovers a near-optimal regret bound when $C = 0$, while incurring an additive $O(d^{3/2} C \log T)$ term (up to $\log\log(dT)$ factors) more generally. A standard lower bound argument [24] shows that $\Omega(C)$ dependence is unavoidable, thus certifying the upper bound as being optimal up to logarithmic factors when $d = O(1)$ (the precise d dependence is not a

main focus of our work).

- For unknown C, we modify our algorithm to gradually decrease its confidence bound enlargement term over time, and show that we only pay a further $O(C^2)$ term compared to the known C case. While this limits the regime of sublinear regret to $C = o(\sqrt{T})$ (in contrast with $C = o(T)$ when C is known), we additionally provide a novel algorithm-independent lower bound showing that this is unavoidable for any algorithm that achieves a near-optimal non-corrupted ($C = 0$) bound. Thus, we prove a fundamental difficulty in being robust against our strong adversary when C is unknown, and demonstrate a fundamental gap between the known C and unknown C settings.
- We introduce a linear contextual problem with adversarial attacks, and show that under the model of diverse contexts from [14], the greedy algorithm not only attains near-optimal regret in the uncorrupted setting (as shown in [14]), but is also *robust to adversarial attacks.*

2 Algorithm and Regret Bounds

We present our Robust Phased Elimination algorithm in Algorithm 1, which builds on non-robust elimination algorithms [18, 19, 30], with some important differences outlined in Remark 3 below. The known C vs. unknown C variants only differ on Line 1. The algorithm runs in epochs of exponentially increasing length and maintains a set of potentially optimal actions. In every epoch, the following steps are performed: (i) compute a near-optimal experimental design over a set of potentially optimal actions, and play each action from this subset in proportion to the computed design (Lines 2-4); (ii) compute an estimate of θ, and use it to eliminate actions that appear suboptimal (Lines 5-6). We proceed by describing these steps in more detail.

Action selection. To introduce the action selection procedure, consider the problem of finding a probability distribution $\zeta : \mathcal{A} \to [0, 1]$ that solves the following:

$$\text{minimize}_{\zeta} \quad \max_{a\in\mathcal{A}} \|a\|^2_{\Gamma(\zeta)^{-1}} \quad \text{s.t.} \quad \sum_{a\in\mathcal{A}} \zeta(a) = 1, \tag{3}$$

where $\Gamma(\zeta) = \sum_{a\in\mathcal{A}} \zeta(a) a a^T$, and $\|a\|_M = \sqrt{a^T M a}$. A classical result from [16] states that the optimal solution ζ^* exists, and achieves $\max_{a\in\mathcal{A}} \|a\|^2_{\Gamma(\zeta^*)^{-1}} = d$ with $|\text{supp}(\zeta^*)| \le \frac{d(d+1)}{2}$. For our purposes, however, it suffices to solve the problem in (3) only near-optimally. As noted in [19], there exists a near-optimal design of smaller support than $d(d+1)/2$. In particular, if $\mathcal{A}$ spans $\mathbb{R}^d$,[1] then we can *efficiently* compute $\zeta : \mathcal{A} \to [0, 1]$ such that

$$\max_{a\in\mathcal{A}} \|a\|^2_{\Gamma(\zeta)^{-1}} \le 2d, \quad |\text{supp}(\zeta)| \le 4d(\log\log d + 18) \tag{4}$$

This follows from [29, Proposition 3.17], who provide a polynomial-time Frank-Wolfe algorithm.

Hence, in every epoch h, the algorithm recomputes a near-optimal design from (4) over a subset of the actions that are still potentially optimal, i.e., $\mathcal{A}_h$. It then plays each action from this subset in proportion to the computed design, but it also makes sure that every arm in its support is played at least some minimal number of times $\lceil \nu m_h \rceil$, where ν is an input truncation parameter to be chosen below, and m_h is an exponentially increasing parameter with respect to the epoch length.

Parameter estimation and arm elimination. Consider the estimator given in (6). This estimator only depends on the observations received in the current epoch, and hence, it is not affected by attacks suffered during previous epochs. However, it can still be biased due to the adversarial attacks suffered in the current epoch, and we need to account for this bias. In Lemma 4 (Appendix A), for any of the remaining potentially optimal actions, we bound the difference of the true mean reward and estimated one, and show that this error grows linearly with the total attack budget C. Hence, the algorithm makes use of the enlarged confidence bounds in (8) to retain potentially optimal arms. Moreover, we show that when C is known, our estimator is guaranteed to have sufficient accuracy so that the optimal arm is always retained in (8) with high probability. For unknown C, this is not always the case, but we can control the level of suboptimality of the arms that are retained.

The estimator of θ is robust due to the fact that it averages the rewards corresponding to the same played action, reducing the effect of the attack. Intuitively, actions that have higher importance according to the found near-optimal design are played more times than others. Consequently, it is harder for the adversary to corrupt them as it needs to use more of the attack budget. In addition, due to the introduced truncation, the algorithm plays each arm in the support of the computed design a fixed minimum number of times.

Remark 2. *The following observations from [19] are useful: (i) While (4) is stated assuming the arms span $\mathbb{R}^d$, we can simply work in the lower-dimensional subspace otherwise (e.g., when $k < d$); (ii) We can extend the algorithm and its analysis to infinite-arm settings using a covering argument.*

Remark 3. *Phased elimination algorithms (without robustness to adversarial attacks) have previously been considered in various settings, including the standard setting [18, Ch. 22], misspecified setting [19], and graph*

[1] See Remark 2 below for the general case.

Algorithm 1 Robust Phased Elimination

Require: Actions $\mathcal{A}_0 \subset \mathbb{R}^d$, confidence $\delta \in (0,1)$, truncation parameter $\nu \in (0,1)$, time horizon T

1: Initialize[2] $m_0 = 4d(\log\log d + 18)$, and for each $h \in \{0, 1, \ldots, \log_2 T - 1\}$, set $\hat{C}_h = C$ for known C, or $\hat{C}_h = \min\{\frac{\sqrt{T}}{m_0 \log_2 T}, m_0\sqrt{d}2^{\log_2 T - h}\}$ for unknown C. Initialize $h = 0$.

2: Compute design $\zeta_h : \mathcal{A}_h \to [0,1]$ such that

$$\max_{a \in \mathcal{A}_h} \|a\|^2_{\Gamma(\zeta_h)^{-1}} \le 2d, \text{ and } |\mathrm{supp}(\zeta_h)| \le m_0. \quad (5)$$

3: Set $u_h(a) = 0$ if $\zeta_h(a) = 0$, and $u_h(a) = \lceil m_h \max\{\zeta_h(a), \nu\} \rceil$ otherwise.

4: Take each action $a \in \mathcal{A}_h$ exactly $u_h(a)$ times with corresponding features $(A_t)_{t=1}^{u_h}$ and rewards $(Y_t)_{t=1}^{u_h}$ (implicitly depending on h), where $u_h = \sum_{a \in \mathcal{A}_h} u_h(a)$.

5: Estimate the parameter vector $\hat{\theta}_h$:

$$\hat{\theta}_h = \Gamma_h^{-1} \sum_{t=1}^{u_h} A_t u_h(A_t)^{-1} \sum_{s \in \mathcal{T}(A_t)} Y_s, \quad (6)$$

$$\Gamma_h = \sum_{a \in \mathcal{A}_h} u_h(a) a a^T, \quad (7)$$

where $\mathcal{T}(a) = \{s \in \{1, \ldots, u_h\} : A_s = a\}$ is the set of times at which arm a is played.

6: Update the active set of arms:

$$\mathcal{A}_{h+1} \leftarrow \Big\{ a \in \mathcal{A}_h : \max_{a' \in \mathcal{A}_h} \langle \hat{\theta}_h, a' - a \rangle \le 2\sqrt{\tfrac{4d}{m_h} \log\left(\tfrac{1}{\delta}\right)} + \tfrac{2\hat{C}_h}{m_h \nu} \sqrt{4d(1 + \nu m_0)} \Big\}. \quad (8)$$

7: Set $m_{h+1} \leftarrow 2m_h$, $h \leftarrow h + 1$ and return to step 2 (terminating after T total arm pulls).

bandits [30]. Among these, our algorithm is most similar to [19], but has several important differences: (i) We use a different and more robust estimator of θ; (ii) The confidence bounds are enlarged in terms of $\hat{C}_h$ to account for adversarial corruptions; (iii) The truncation parameter is introduced to ensure that each arm is pulled enough; (iv) In the unknown C case, we need to carefully choose the sequence $\hat{C}_h$ to trade off robustness against aggressiveness in eliminating suboptimal arms; (v) In contrast to the vast majority of existing elimination algorithms, the optimal arm may *be eliminated in the unknown C setting (i.e., the confidence bounds may not be "valid"), but this only occurs when the best remaining arm is still good enough to control the regret.*

2.1 Upper Bounds on Regret

We first provide a regret bound for the known C case, proved in Appendix A.

Theorem 1. *For any attack budget $C \ge 0$, with probability at least $1 - \delta$, the Robust Phased Elimination algorithm with known C and truncation parameter $\nu = \frac{1}{4d(\log\log d + 18)}$ satisfies*

$$R_T = \tilde{O}\Big(\sqrt{dT \log\left(\tfrac{k}{\delta}\right)} + Cd^{3/2} \log T\Big), \quad (9)$$

where the notation $\tilde{O}(\cdot)$ hides $\log\log(dT)$ factors.

When $C = 0$, we recover the scaling of [18, Thm. 22.1], which is near-optimal in light of known lower bounds [8]. In Section 2.2, we will argue that the second term is also near-optimal.

Next, we consider the case that the total attack budget C is unknown to the learner. We start by discussing the choice of $\hat{C}_h$ in Algorithm 1. Let H be the number of epochs, and note that $\tilde{H} = \log_2 T$ be a deterministic upper bound on H (see Appendix A.2 for a short proof). Then, the choice in Algorithm 1 can be rewritten as $\hat{C}_h = \min\{\frac{\sqrt{T}}{m_0 \log_2 T}, m_0\sqrt{d}2^{\tilde{H} - h}\}$. Observe that the epochs' lengths u_h and corruption thresholds $\hat{C}_h$ are exponentially increasing and decreasing, respectively. It follows that the algorithm is more cautious in early epochs (i.e., uses larger thresholds). Our second main result stated is as follows, and proved in Appendix A.

Theorem 2. *For any $C \le \frac{\sqrt{T}}{4d(\log\log d + 18)\log T}$, with probability at least $1 - \delta$, the Robust Phased Elimination algorithm with unknown C and truncation parameter $\nu = \frac{1}{4d(\log\log d + 18)}$ satisfies*

$$R_T = \tilde{O}\Big(\sqrt{dT \log\left(\tfrac{k}{\delta}\right)} + Cd^{3/2} \log T + C^2\Big). \quad (10)$$

This result matches that of Theorem 1, but with an additional penalty of C^2. In fact, due to this penalty, the regret bound (10) trivially holds when $C = \Omega(\sqrt{T})$, because we have $R_T \le 2T$ due to our assumption of bounded rewards. If $d = \omega(1)$, then there still remains the regime where $\frac{\sqrt{T}}{(d \log\log d)\log T} \ll C \ll \sqrt{T}$, but in any case, one can slightly increase the final term and state that $R_T = \tilde{O}\big(\sqrt{dT \log\left(\frac{k}{\delta}\right)} + Cd^{3/2} \log T + C^2 d^2 (\log T)^2\big)$ for arbitrary C.

At this stage, observing that our regret bound is not sublinear in T when $C = \Omega(\sqrt{T})$, the natural question arises as to whether attaining such a goal is impossible for all robust bandit algorithms. In the following subsection, we use an algorithm-independent lower bound to

[2] When $d = 1$, we have $\log\log d = -\infty$, but the results hold with $\log\log d$ replaced by $\log(1 + \log d)$.

provide a partial answer to this question; specifically, such a goal is indeed impossible (up to logarithmic factors) whenever the algorithm is required to have order-optimal regret in the uncorrupted ($C = 0$) case.

2.2 Algorithm-Independent Lower Bounds on Regret

Using the same reasoning as the standard multi-armed bandit setting [24], it is straightforward to see that $\Omega(C)$ regret is unavoidable: The adversary can simply shift all rewards to zero for the first C rounds, and the learner cannot do better than random guessing. For completeness, this argument is given in more detail in Appendix C. This argument holds even when C is known, and thus, we see that the second term in Theorem 1 is optimal up to at most an $\tilde{O}(\log T)$ factor for fixed d. We expect that an improvement on the $d^{3/2}$ dependence may be possible, but the following result, proved in Appendix C, shows that at least $\Omega(Cd)$ is unavoidable.

Theorem 3. *For any dimension d, there exists an instance with $k = d$ such that any algorithm (even with knowledge of C) must incur $\Omega(Cd)$ regret with probability at least $\frac{1}{2}$.*

Next, we provide another lower bound that will allow us to show a sense in which the C^2 term appearing Theorem 2 cannot be significantly improved.

Theorem 4. *For $d = 2$ and $k = 2$, for any algorithm that guarantees $R_T \le \bar{R}_T^{(0)}$ with probability at least $1-\delta$ for a given uncorrupted regret bound $\bar{R}_T^{(0)} \le \frac{T}{16}$ when $C = 0$, there exists an instance in which $R_T = \Omega(T)$ with probability at least $1 - \delta$ when $C = 2\bar{R}_T^{(0)}$.*

The proof is given in Appendix C. While we focus on the simplest case $d = k = 2$, the proof can also be adapted to more general choices.

Discussion. Consider the general goal of attaining a regret upper bound of the form

$$R_T \le \bar{R}_T^{(0)} + f(C)\log T, \tag{11}$$

for some $f(\cdot)$ satisfying $f(0) = 0$. Here we let the second term contain a $\log T$ factor in accordance with our upper bounds, but the following discussion still applies with only minor modifications when the $\log T$ factor is changed to $\mathrm{poly}(\log T)$ or similar.

At first glance, it appears that $f(C)$ should ideally be linear in C, and $\bar{R}_T^{(0)}$ should ideally be an order-optimal regret bound for the non-corrupted setting. However, Theorem 4 shows that we cannot have both terms exhibiting their "ideal" behavior simultaneously. To see this, note that the ideal uncorrupted regret bound behaves as $\bar{R}_T^{(0)} = \tilde{\Theta}(\sqrt{T})$ (for fixed d, k, and δ) [8, 18]. Then, to be consistent with Theorem 4, we require $f(C)\log T = \tilde{\Omega}(T)$ for $C = \Theta(\sqrt{T})$, and hence $f(C) = \tilde{\Omega}\big(\frac{C^2}{\log C}\big)$.

On the other hand, it may be possible remove the C^2 term from $f(C)$ (i.e., improve robustness), and to attain sublinear regret for certain cases with $C = \Omega(\sqrt{T})$, if one is willing to pay the price of a worse uncorrupted regret bound. This idea is left for future work.

2.3 Summary of Upper vs. Lower Bounds

We conclude this section with a short summary of how the upper and lower bounds compare in various scaling regimes of C and T, when the other parameters (d, k, δ) are held fixed:

- When C is known, the optimal regret is between $\Omega\big(\sqrt{dT}+C\big)$ and $\tilde{O}\big(\sqrt{dT}+C\log T\big)$ for any $C \le T$;
- For $C = O\big(\frac{T^{1/4}}{\log T}\big)$, the optimal regret scales as $\tilde{\Theta}\big(\sqrt{dT}\big)$ for both known and unknown C;
- For $C = \Omega(\frac{\sqrt{T}}{\log T})$, we do not provide any sublinear regret bound for when C is unknown, but Theorem 4 shows that, in fact, such a bound cannot be expected for $C = \Omega(\sqrt{T})$ unless the uncorrupted regret increases significantly.
- For C in between the previous two dot points (e.g., $C = \Theta(T^a)$ with $\frac{1}{4} < a < \frac{1}{2}$), our upper bound for unknown C exhibits strictly higher scaling than the uncorrupted regret (due to the C^2 term), and it remains open as to what extent this is unavoidable.

3 Greedy Algorithm in the Contextual Setting

In this section, we consider a k-arm linear contextual bandit problem with a single unknown d-dimensional parameter vector $\theta \in \mathbb{R}^d$ (e.g., see [14]). In each round t, contexts $a_{1,t}, \dots, a_{k,t}$ are presented to the learner, each in $\mathbb{R}^d$ and associated to one action. The learner then chooses an action indexed by $I_t \in \{1, \dots, k\}$ and observes the corrupted reward

$$Y_t = \langle \theta, a_{I_t,t}\rangle + \epsilon_t + c_t(a_{I_t,t}), \tag{12}$$

where the same assumptions from Section 1.1 hold for both $(\epsilon_t)_{t=1}^T$ and $c_t(\cdot)$ (with attack budget C), and $\|\theta\|_2 \le 1$. Similar to (2), the cumulative regret is $R_T = \sum_{t=1}^T \max_{i\in\{1,\dots,k\}} \langle \theta, a_{i,t} - a_{I_t,t}\rangle$.

In general, the introduction of contexts may significantly complicate the problem, with algorithms such as the one in Section 2 being difficult to extend, particularly with unknown C. However, perhaps surprisingly, a line of recent works has demonstrated that simple

exploration-free greedy methods can provably work well (in the non-corrupted setting) under mildassumptions on the contexts. These assumptions amount to kinds of *context diversity* [3, 14, 26] ensuring that the collected samples are sufficiently informative for learning θ accurately.

Most related to this paper is [14], who analyze the greedy algorithm in the case that arbitrary context vectors undergo small random perturbations. Motivated by these results, we investigate the performance of the greedy algorithm under the same assumption on the contexts, but with the addition of adversarial attacks. Our main finding is that the context diversity assumption not only removes the need for explicit exploration [14], but also automatically inherits near-optimal robustness to adversarial attacks, with no need to know the attack budget C.

Context generation. In more detail, the setup of [14] is introduced as follows: An arbitrary tuple $\mu_{1,t}, \dots, \mu_{K,t}$ of mean context vectors is given (possibly selected by an adaptive adversary based on the history of contexts, actions, and rewards), such that $\|\mu_{i,t}\|_2 \le 1$ for all i, t. For every available action, the context vector is then generated as $a_{i,t} = \mu_{i,t} + \xi_{i,t}$, where the random perturbation vectors $\xi_{i,t}$ are drawn independently from some zero-mean distributions $D_{1,t}, \dots, D_{K,t}$. We consider perturbations that are (r, δ)-*bounded* for some $r \le 1$ according to the following definition [14]:

$$\mathbb{P}[\|\xi_{i,t}\|_\infty \le r \text{ for all arms } i \text{ and rounds } t] \ge 1 - \delta. \tag{13}$$

As outlined above, we are interested in the diversity of samples collected by the greedy algorithm (defined below). The main idea is that the observed contexts should cover all directions in order to enable good estimation of the latent vector θ. Consequently, we make use of the notion of diversity from [14], which takes into account that the learner observes rewards for contexts that are selected greedily and thus only observes a conditional distribution of contexts. Specifically, following [14], a distribution D is called (r, λ_0)-*diverse* with parameters $r > 0$ and $\lambda_0 > 0$ if, for $a = \mu + \xi$ with $\xi \sim D$ and any $\mu \in \mathbb{R}^d$, it holds for all $\hat{\theta} \in \mathbb{R}^d$ and $\hat{b} \in \mathbb{R}$ satisfying $\hat{b} \le r\|\hat{\theta}\|_2$ that

$$\lambda_{\min}\Big(\mathbb{E}_{\xi \sim D}\big[aa^T \,\big|\, \hat{\theta}^T \xi \ge \hat{b}\big]\Big) \ge \lambda_0. \tag{14}$$

The overall perturbations are (r, λ_0)-*diverse* if the distributions $D_{i,t}$ are (r, λ_0)-diverse for all i and t.

This diversity condition is the main component in [14] for proving that the minimum eigenvalue of the empirical covariance matrix $\lambda_{\min}(\sum_{\tau=1}^{t} a_{I_\tau,\tau} a_{I_\tau,\tau}^T)$ grows linearly with t. In Lemma 6 (Appendix B), we demonstrate that this is the main quantity that has an impact on the accuracy of the estimator of θ, and in turn, on the regret bounds in the corrupted setting.

Greedy algorithm. In round t, the greedy algorithm (see Algorithm 2) receives a set of contexts $\{a_{1,t}, \dots, a_{k,t}\}$, and chooses the best action according to the least squares estimate of θ:

$$I_t = \operatorname*{arg\,max}_{i \in \{1,\dots,K\}} \langle \hat{\theta}_t, a_{i,t} \rangle, \tag{15}$$

$$\hat{\theta}_t = \operatorname*{arg\,min}_{\theta'} \sum_{\tau=1}^{t-1} (\langle \theta', a_{I_\tau,\tau} \rangle - Y_\tau)^2. \tag{16}$$

Our regret bound for this setup is stated as follows, and proved in Appendix B.

Theorem 5. *Suppose that $\|a_{i,t}\|_2 \le 1$ for all i, t, the random context perturbations are $(r, 1/T)$-bounded and (r, λ_0)-diverse with $r \le 1$, the reward noise is 1-sub-Gaussian, and the attack budget is $C \ge 0$. Then with probability at least $1 - \delta$, the greedy algorithm has regret bounded by*

$$R_T = O\bigg(\tfrac{1}{\lambda_0}\Big(\sqrt{dT \log\big(\tfrac{dT}{\delta}\big)} + C \log T + \log\big(\tfrac{dT}{\delta}\big)\Big) + \sqrt{\log\big(\tfrac{k}{\delta}\big)}\bigg). \tag{17}$$

Under the mild assumptions $\delta = e^{-O(dT)}$ and $\frac{k}{\delta} = e^{O(dT)}$, this bound simplifies to

$$R_T = O\bigg(\tfrac{1}{\lambda_0}\Big(\sqrt{dT \log\big(\tfrac{Td}{\delta}\big)} + C \log T\Big)\bigg). \tag{18}$$

In addition, when $C = 0$, Theorem 5 reduces to the result of [14]. The additional $\frac{1}{\lambda_0} C \log T$ term is essentially optimal when $\lambda_0 = \Theta(1)$, since a simple argument from [24] gives an $\Omega(C)$ lower bound (see Appendix C). In Corollary 1 (Appendix B), we specialize Theorem 5 to the case that the perturbations are Gaussian, i.e., every $\xi_{i,t}$ is drawn independently from $\mathcal{N}(0, \eta^2 I)$, and show that the greedy algorithm has sublinear regret in the low-η regime.

Theorem 5 indicates that the greedy algorithm can be robust despite being extremely simple, having no explicit built-in mechanism for combating robustness, and having no knowledge C. A caveat to this is the $\frac{1}{\lambda_0}$ dependence, indicating that the regret can increase significantly when the contexts are not sufficiently diverse.

4 Experiments

In this section, we evaluate the performance of the algorithms studied in this paper, along with the baselines LinUCB [20, 18] and Thompson sampling [1].[3]

[3] We use LinUCB as described in [18, Sec. 19.2] with least-squares regularization parameter $\lambda = 1$ and confidence

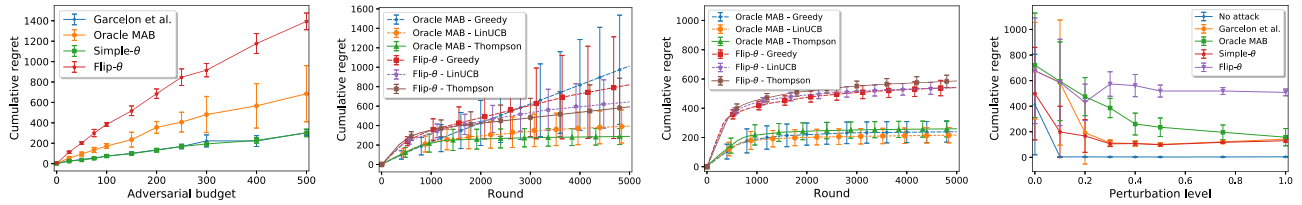

Figure 1: Contextual synthetic experiment: (Left) Regret at time $T = 3500$ as a function of C with $\eta = 0.5$; (Middle Two) Regret as a function of time with $\eta = 0$ and $\eta = 0.5$; (Right) Performance of Greedy at time $T = 3500$ with $C = 150$ and varying η.

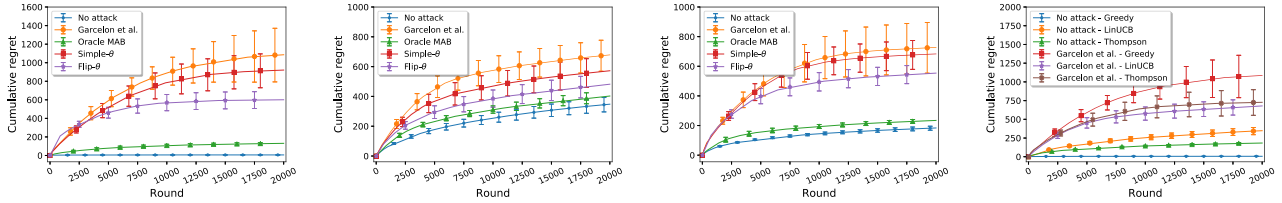

Figure 2: MovieLens experiment: (Left 3) Regret as a function of time with $C = 150$ for Greedy, LinUCB, and Thompson sampling; (Right) Regret of all algorithms under the Garcelon *et al.* attack.

We consider both the robust PE algorithm and the contextual greedy algorithm, starting with the latter.

4.1 Choices of Attacks

We consider the following attack algorithms, each depending on a target arm a_{target} and/or a target parameter vector θ_{target}. These are briefly outlined as follows, with more details in Appendix D:

- **Garcelon *et al.* attack.** This attack is a minor modification of that of [10], leaving pulls from a_{target} uncorrupted, while pushing all other rewards down to the minimum value.
- **Oracle MAB attack.** This attack from [13] pushes the reward of any $a \neq a_{\text{target}}$ to some margin ϵ_0 below that of a_{target}, or leaves the reward unchanged if such a margin is already met.
- **Simple θ-based attack.** This attack acts in the same way as that of Garcelon *et al.*, but with a_{target} always chosen as $\arg\max_a \langle a, \theta_{\text{target}} \rangle$. This is equivalent to that of [10] in the non-contextual setting, but otherwise may differ due to a_{target} varying with time.
- **Flip-θ attack.** This attack simply flips the reward from $\langle \theta, a \rangle$ to $\langle -\theta, a \rangle$.

Note that the terminology "oracle" refers to attacks that use knowledge of θ, which we assume to be permitted in this paper (the Flip-θ attack also falls in this category). We set a_{target} to be the first arm, which will have the same effect as choosing any fixed arm (since our arm feature vectors will be generated in a symmetric manner). In addition, we let θ_{target} be uniform on the unit sphere in the simple θ-based attack, and set $\epsilon_0 = 0.01$ in the Oracle MAB attack.

4.2 Contextual Setting

Synthetic Experiment. In this experiment, we consider the contextual case with contexts having uniform entries and Gaussian perturbations with variance $\eta^2 > 0$; see Appendix E for the full details. We consider $k = 25$ arms, $T = 5000$ rounds, and attack budget $C = 50$. At each time instant, we plot the cumulative regret averaged over 10 trials, and error bars indicate one standard deviation. In Appendix E, we provide analogous plots and discussion when $C = 150$.

In Figure 1 (Left), we plot the regret of Greedy at $T = 3500$ as a function of C with $\eta = 0.5$. We observe a linear increase, which is in agreement with our theory. Analogous plots for LinUCB, Thompson sampling, and $\eta \in \{0.2, 0.5\}$ can be found in Appendix E. The middle two plots in Figure 1 show the regret as a function of time with the two most effective attacks, with $\eta = 0$ and $\eta = 0.5$. We see that the regret curves are still increasing linearly under the Flip-θ attack by time $T = 5000$ when $\eta = 0$, whereas they are nearly flat when $\eta = 0.5$. While our theory only supports the robustness of Greedy, these experiments suggest that LinUCB and Thompson sampling may also enjoy similar robustness under context diversity. Finally, Figure 1 (Right) plots the regret of Greedy at $T = 3500$ as a function of η when $C = 150$. We observe that once η moves past a certain level, the performance remains fairly consistent, with a general (but not definitive) trend of decreasing regret. The greatest difference is at $\eta = 0$, particularly when the standard deviation is considered.

MovieLens Experiment. We use the MovieLens-

parameter $\delta = 0.1$, and Thompson sampling [1] uses an i.i.d. Gaussian prior with variance 0.5.

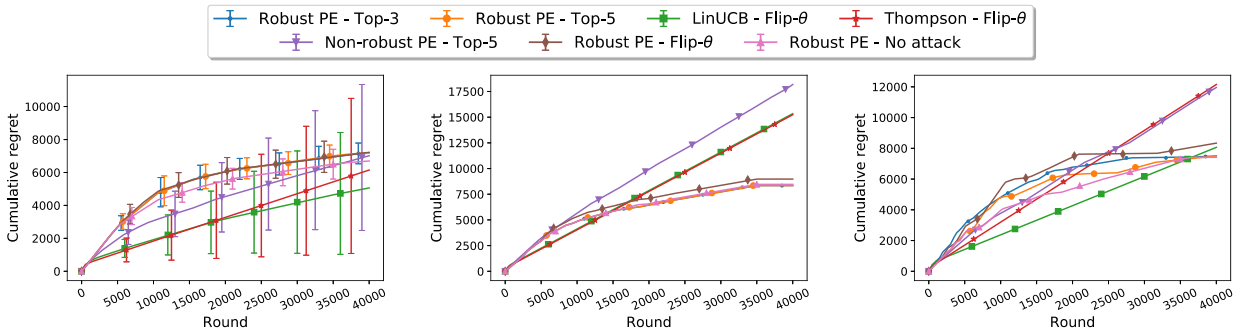

Figure 3: Non-contextual synthetic experiment with 10 trials: (Left) Average regret as a function of time; (Middle) Worst run among 10; (Right) Second-worst run among 10.

100K dataset in a similar manner to [5]; see Appendix E for details. In each trial, we select a uniformly random user and treat the 1682 movies as possible contexts. At each time instant, $k = 30$ of these movies are chosen uniformly at random and presented as the contexts. Hence, a subset of the movie vectors form the contexts, and a fixed user vector forms θ. We set $T = 20000$ and $C = 150$, and we plot the regret averaged over 10 trials (each corresponding to a different user).

In Figure 2, we plot the regret as a function of time, for Greedy, LinUCB, and Thompson sampling. Despite the lack of explicit context perturbation in this experiment, we see that the algorithms are again able to recover from the attacks, suggesting that the various movies in the data set are sufficiently diverse. On the other hand, we do not claim the attacks here to be optimal, and it is possible that stronger attacks may incur linear regret. In Figure 2 (Right), we plot all three algorithms under the strongest attack and under no attack. We see that Greedy has very low regret when there is no attack, but has slightly higher regret when attacked.

4.3 Non-Contextual Setting

We now turn to experiments for the robust PE algorithm (Algorithm 1), with some minor practical changes detailed in Appendix E. We use the above synthetic experimental setup with the context perturbations removed (i.e., $\eta = 0$), and with $d = 5$, $k = 50$, $T = 40000$, and $C = 150$. For comparison, we also include non-robust PE, which removes the second term in (8).

For LinUCB, Thompson sampling, and non-robust PE, we continue to attack right from the start. However, for robust PE, this is a poor attack strategy, since the algorithm initially uses a very stringent condition for elimination. Instead, following insight from the proof of Theorem 2, we start the attack at the first epoch for which $\hat{C}_h < C$. We consider the Flip-θ attack of Section 4.1, as well as an additional *Top-N attack* targeted at eliminating good arms: Whenever any of the top N remaining arms are pulled, push the reward to -1. We consider both $N = 3$ and $N = 5$. We focus on the case of unknown C here, and present similar plots for known C in Appendix E.

In Figure 3 (Left), we see that the average regret of all algorithms is similar by the end of the time horizon; however, an inspection of the error bars reveals that this is not the full story. In particular, the regret of LinUCB and Thompson sampling vary considerably depending on whether the attack was successful or not, whereas robust PE exhibits much lower variation. To highlight this, we plot the regret from the worst and second-worst runs out of 10 (as measured at time T) in Figure 3 (Middle) and Figure 3 (Right). In Appendix E, we provide analogous plots in the case of 40 trials, showing the worst 4-out-of-40 runs and observing similar behavior to Figure 3.

We see that LinUCB and Thompson sampling visibly have linear regret, whereas the regret of robust PE flattens out by the end of the time horizon even for these worst-2-of-10 curves, indicating better *high-probability* behavior. In contrast, these results suggest the possibility of algorithms with improved *finite-time* performance guarantees, which was not the focus of our work.

5 Conclusion

We have considered the linear stochastic problem in the presence of adversarial attacks/corruptions. We provided novel algorithms in both the standard and contextual settings that are provably robust against such attacks. We demonstrated near-optimal regret bounds in all cases, and to our knowledge, we are the first to do so in each case. A possible direction for future work is to consider a setting in which both rewards and contexts can be altered by the adversary subject to a limited attack budget.

Acknowledgments

We are grateful to Akshay Krishnamurthy for helpful discussions regarding the existing literature on contextual bandit model selection.

This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme grant agreement No 815943 and ETH Zürich Postdoctoral Fellowship 19-2 FEL-47. J. Scarlett was supported by the Singapore National Research Foundation (NRF) under grant number R-252-000-A74-281.

References

[1] S. Agrawal and N. Goyal, "Thompson sampling for contextual bandits with linear payoffs," in *International Conference on Machine Learning*, 2013.

[2] P. Auer and C.-K. Chiang, "An algorithm with nearly optimal pseudo-regret for both stochastic and adversarial bandits," in *Conference on Learning Theory*, 2016.

[3] H. Bastani, M. Bayati, and K. Khosravi, "Mostly exploration-free algorithms for contextual bandits," *arXiv preprint arXiv:1704.09011*, 2017.

[4] I. Bogunovic, A. Krause, and J. Scarlett, "Corruption-tolerant Gaussian process bandit optimization," in *Conference on Artificial Intelligence and Statistics*, 2020.

[5] I. Bogunovic, J. Scarlett, S. Jegelka, and V. Cevher, "Adversarially robust optimization with Gaussian processes," in *Advances in Neural Information Processing Systems*, 2018.

[6] S. Bubeck, N. Cesa-Bianchi, and S. M. Kakade, "Towards minimax policies for online linear optimization with bandit feedback," in *Conference on Learning Theory*, 2012.

[7] S. Bubeck and A. Slivkins, "The best of both worlds: Stochastic and adversarial bandits," in *Conference on Learning Theory*, 2012.

[8] V. Dani, T. P. Hayes, and S. M. Kakade, "Stochastic linear optimization under bandit feedback," in *Conference on Learning Theory*, 2008.

[9] D. Foster and A. Rakhlin, "Beyond ucb: Optimal and efficient contextual bandits with regression oracles," in *International Conference on Machine Learning*. PMLR, 2020, pp. 3199–3210.

[10] E. Garcelon, B. Roziere, L. Meunier, O. Teytaud, A. Lazaric, and M. Pirotta, "Adversarial attacks on linear contextual bandits," *arXiv preprint arXiv:2002.03839*, 2020.

[11] A. Ghosh, S. R. Chowdhury, and A. Gopalan, "Misspecified linear bandits," in *AAAI Conference on Artificial Intelligence*, 2017.

[12] A. Gupta, T. Koren, and K. Talwar, "Better algorithms for stochastic bandits with adversarial corruptions," *arXiv preprint arXiv:1902.08647*, 2019.

[13] K.-S. Jun, L. Li, Y. Ma, and J. Zhu, "Adversarial attacks on stochastic bandits," in *Advances in Neural Information Processing Systems*, 2018.

[14] S. Kannan, J. H. Morgenstern, A. Roth, B. Waggoner, and Z. S. Wu, "A smoothed analysis of the greedy algorithm for the linear contextual bandit problem," in *Advances in Neural Information Processing Systems*, 2018.

[15] S. Kapoor, K. K. Patel, and P. Kar, "Corruption-tolerant bandit learning," *Machine Learning*, vol. 108, no. 4, pp. 687–715, Apr. 2019.

[16] J. Kiefer and J. Wolfowitz, "The equivalence of two extremum problems," *Canadian Journal of Mathematics*, vol. 12, pp. 363–366, 1960.

[17] A. Krishnamurthy, Z. S. Wu, and V. Syrgkanis, "Semiparametric contextual bandits," in *International Conference on Machine Learning*, 2018.

[18] T. Lattimore and C. Szepesvári, "Bandit algorithms," *preprint*, vol. 28, 2018.

[19] T. Lattimore and C. Szepesvári, "Learning with good feature representations in bandits and in RL with a generative model," in *International Conference on Machine Learning*, 2020.

[20] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *International Conference on World Wide Web*, 2010.

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

[22] F. Liu and N. Shroff, "Data poisoning attacks on stochastic bandits," in *International Conference on Machine Learning*, 2019.

[23] G. Liu and L. Lai, "Action-manipulation attacks on stochastic bandits," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[24] T. Lykouris, V. Mirrokni, and R. Paes Leme, "Stochastic bandits robust to adversarial corruptions," in *ACM SIGACT Symposium on Theory of Computing*, 2018.

[25] G. Neu and J. Olkhovskaya, "Efficient and robust algorithms for adversarial linear contextual bandits," in *Conference on Learning Theory*. PMLR, 2020, pp. 3049–3068.

[26] M. Raghavan, A. Slivkins, J. W. Vaughan, and Z. S. Wu, "The externalities of exploration and how data diversity helps exploitation," *arXiv preprint arXiv:1806.00543*, 2018.

[27] Y. Seldin and A. Slivkins, "One practical algorithm for both stochastic and adversarial bandits." in *International Conference on Machine Learning*, 2014.

[28] A. Slivkins *et al.*, "Introduction to multi-armed bandits," *Foundations and Trends in Machine Learning*, vol. 12, no. 1-2, pp. 1–286, 2019.

[29] M. J. Todd, *Minimum-Volume Ellipsoids*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2016.

[30] M. Valko, R. Munos, B. Kveton, and T. Kocák, "Spectral bandits for smooth graph functions," in *International Conference on Machine Learning*, 2014.

[31] A. Zanette, A. Lazaric, M. Kochenderfer, and E. Brunskill, "Learning near optimal policies with low inherent Bellman error," *arXiv preprint arXiv:2003.00153*, 2020.