Robust Near-Optimal Arm Identification With Strongly-Adaptive Adversaries

Mayuri Sridhar and Srinivas Devadas (MIT CSAIL)

***Abstract*—In this work, we study the best arm identification problem in the adversarial multi-armed bandits framework. We define a strongly-adaptive adversarial model in this framework, based on strongly-adaptive adversaries in security and distributed systems. On the negative side, we show the increased strength of the adversarial model by proving that it is impossible for any best-arm identification algorithm to return an arm with rank $\leq \left\lfloor \frac{\epsilon K}{1+\epsilon_0} \right\rfloor$, where K is the number of arms, ϵ is the adversary's budget and ϵ_0 is the breaking point of the robust mean estimation subroutine. On the positive side, we construct a novel sequential elimination algorithm which returns a near-optimal arm (with rank $\leq \left\lceil (1+\lambda) \left\lfloor \frac{\epsilon K}{\epsilon_0} \right\rfloor \right\rceil$ where $\lambda > 0$ is a function of ϵ and ϵ_0 and tends to 0 for small ϵ) with high probability. We evaluate our algorithm on both synthetic and real-world datasets and empirically demonstrate that our algorithm returns a near-optimal arm under a strongly-adaptive adversarial model.**

***Index Terms*—Multi-armed bandits, best arm identification, strongly-adaptive adversaries, sequential decision-making.**

I. Introduction

The multi-armed bandits framework is used to make decisions under uncertainty. Originally formalized by Robbins [1], the multi-armed bandits problem models a gambler that has to choose between K different slot machines, each of which has an arm to pull, and produces a reward following some probability distribution. In each round, the gambler chooses an arm and collects the reward for the chosen arm. In the stochastic bandits model, the reward for an arm i is drawn from an unknown distribution and the reward for each pull is independent and identically distributed [2, 3]. This framework is used to model many problems, including dynamic pricing, medical trials and many more (see e.g., Bouneffouf et al. [4] for an overview of practical applications) [5, 6, 7, 8, 9]. Typically, the goal of the algorithm is to maximize the total reward over n rounds.

We focus on the best arm identification (BAI) problem, a variant which solves the pure exploration problem of finding the best arm to pull, while also minimizing sample complexity [3, 10, 11, 12]. This targets an important class of problems where there is a separate exploration phase before deployment and we are trying to find the optimal arm as quickly as possible. For instance, consider a company trying to choose between K different products to launch. Before launch, the company tests each product in a *preliminary* phase with the goal of choosing the best product as quickly as possible (minimal sample complexity). Often, in real-world settings, it is practically important to consider complex environments. For instance, simple decisions, like choosing the optimal route to transmit packets in a communication network (see e.g., Rekhter and Li [13]), may have nonstationarity among the costs of each route and complex dependencies [3]. These real-world environments can be modeled adversarially; in this work, we focus on the best arm identification problem in a strong adversarial setting.

Please contact authors at mayuri@mit.edu and devadas@mit.edu for any additional information.

A. Related Work

Early work on best arm identification focuses on bounding overall regret and minimizing sample complexity [3, 10, 11, 12]. Audibert and Bubeck [12] define the *regret* of an algorithm as the gap between the mean reward of the optimal arm and the mean reward of the ultimately chosen arm. Mannor and Tsitsiklis [10] show that finding an ϵ-optimal arm (where the difference between mean rewards of the chosen arm and the best arms is $\leq \epsilon$) with probability at least $1-\delta$ requires $\Omega(\frac{K \log(1/\delta)}{\epsilon^2})$ samples in expectation when the reward of each arm is drawn from an i.i.d. Bernoulli distribution. Even-Dar et al. [14] shows that this is tight up to a constant factor by constructing a sequential elimination algorithm to identify an ϵ-optimal arm, with high probability.

In contrast, we focus on the best arm identification problem with *adversarial corruptions*. In this setting, the goal of the algorithm is to return an optimal (or near-optimal) arm with minimal sample complexity while an adversary tries to fool it by corrupting a subset of the rewards. The goal is to design an algorithm that is *robust* (in terms of the mean reward of the returned arm and sample complexity) to such adversarial manipulations. A rich line of work explores robust best arm identification in adversarial contexts, and considers both adaptive and non-adaptive adversaries.

a) Non-adaptive Adversaries: Auer et al. [3] considers a setting where an adversary has complete control over the sequence of rewards from each slot machine. That is, each slot machine is assigned the sequence of rewards *before* the algorithm is run, but no other constraint is placed on the sequence [3]. They do not consider "best arm identification" since there is no underlying ordering on the arms.

Subsequent work by Zhang and Shen [15] transforms the ideas of Auer et al. to the best arm identification setting by introducing additional assumptions; they consider a variant where the corruption of the rewards for each arm converges to 0 in the limit. Recent work by Mukherjee et al. [16] considers an adversarial bandit model where the adversary can replace rewards drawn from each arm with probability ϵ.

In all of these settings, the adversary is *non-adaptive*. Unfortunately, assuming that the adversary is non-adaptive can

be unrealistic because it would require all of the adversary's decisions to be *independent* of the algorithm's execution. In the real world, the adversary is likely to see the algorithm's prior choices which it could incorporate into its decisions. Therefore, these works can be insufficient to guarantee robustness against such adversaries.

b) Adaptive Adversaries: We then consider works which assume that adversaries can adaptively corrupt rewards based on the algorithm's past choices. This introduces several challenges when it comes to guaranteeing robustness. The seminal work by Lykouris et al. [17] considers an adaptive adversarial model. In their model, the adversary can corrupt the rewards for each arm in a given round of the algorithm, and does so *after* observing the algorithm's past choices and the uncorrupted rewards for the current iteration. Gupta et al. [18] considers a similar adaptive adversarial model and improve the regret bounds of Lykouris et al. [17]. Zhong et al. [19] analyzes this model for the best arm identification problem. Follow-up work by Feng et al. [20] explores more complex adaptive adversarial strategies where each arm can independently corrupt samples.

We note that these prior works allow regret to scale with the total magnitude of corruptions introduced by the adversary. Another line of work considers adversaries that can make arbitrary corruptions, where the corruptions are not limited in their total magnitude. Altschuler et al. [21] considers a setting with unbounded corruptions and an adaptive adversary. In particular, they model each arm pull as drawing from an arbitrary corrupted distribution with probability ϵ.

c) Motivation for Strongly-Adaptive Adversaries: With the prevalence of multi-armed bandits problems in large-scale systems (including healthcare, finance and more [4, 5, 6, 7, 8, 9]), best-arm identification algorithms will need to be deployed in a distributed manner, making robustness in distributed settings crucial. In particular, the literature in distributed systems and security considers *strongly*-adaptive adversaries to model real-world adversarial capabilities (see e.g., Feldman and Micali [22], Dutta et al. [23], Lewko and Lewko [24], Goldwasser et al. [25], Abraham et al. [26, 27]). Unlike adaptive adversaries, strongly-adaptive adversaries first *observe* all the messages passed in a given round *before* deciding on the corruption strategy for the round [22, 23, 24, 25, 26, 27].

For example, Auer et al. [3] discusses applications of the multi-armed bandits problem to communication networks where the algorithm is tasked with finding an optimal route for packet traffic. Meanwhile, an adversary can try to force suboptimal or adversarial paths to be chosen. In the real world, the adversary might have a history of the algorithm's past choices and use them to predict the algorithm's current choice; in fact, learning-based algorithms have shown significant potential in similar tasks [28, 29, 30]. These settings can only be realistically captured by considering a strongly-adaptive adversary, which motivates the formalization of a more powerful adversarial model than the one considered in prior work.

Similar adversarial models have been explored in the multi-armed bandits literature before; however, prior work typically focused on designing optimal attacks against specific algorithms. For instance, Jun et al. [31] considers an adversary which observes the algorithm's chosen arm and reward, then chooses an adaptive corruption strategy, modeling the distributed setting. The cost of the attack was measured over the horizon T, where the cost is defined as

$$\sum_{t=1}^{T} \alpha_t = \sum_{t=1}^{T} (\hat{r_t} - r_t),$$

where r_t is the true reward and $\hat{r_t}$ is the observed reward. This work considers two algorithms, ϵ-greedy and UCB, and constructs efficient attack strategies. Zuo [32] later improved the cost of their attack from $O(\log T)$ to $\hat{O}(\sqrt{\log T})$ for the UCB algorithm. Similarly, Liu and Shroff [33] design an online adaptive attack strategy, which minimizes the magnitude of the reward corruption while ensuring any black-box algorithm incurs linear regret.

Inspired by these works, we investigate the multi-armed bandits problem under a strongly-adaptive adversarial model. We consider the problem of constructing a robust algorithm that defends against the strong potential attacks that a strongly-adaptive adversarial model enables. Recent work by Rangi et al. [34] constructs algorithms to combat these attacks by allowing the algorithms a limited number of queries to the uncorrupted reward distribution; our work constructs a robust near-optimal BAI algorithm without any access to uncorrupted data.

We model a strongly-adaptive adversary as an adversary who first observes the sequence of arms the algorithm chooses to sample in a given stage and the corresponding rewards *before* deciding on a subset of samples to corrupt in that stage. We allow the adversary to know, a priori, the underlying reward distribution and the algorithm's strategy, allowing for much more complex and realistic adversarial strategies than prior work [31, 33]. We further allow the adversary to be bounded only by the *fraction* of corruptions introduced, rather than the magnitude; to the best of our knowledge, this is the first work that considers arbitrary corruptions in the strongly-adaptive adversarial model.

d) Comparison to Prior Models: In this section, we show that the strongly-adaptive adversarial model with arbitrary corruptions is significantly stronger than previous adversarial models. We denote the total number of samples as n. In prior adversarial models, for each sample, an adversary with budget ϵ can decide whether to corrupt the reward either randomly or based on past history, while maintaining the guarantee that the total number of times that the reward is corrupted is at most ϵn. In this setting, it is difficult for the adversary to corrupt all samples from a particular arm, which is key to the convergence guarantees of prior work.

Mukherjee et al. [16] makes this assumption by only analyzing corruptions that are Bernoulli distributed and independent; as such, this approach divides the corruptions across all arms and ensures that no arm has too many corrupted samples in total. Similarly Altschuler et al. [21] models the corrupted distribution by a mixture model *per arm* where corrupted samples are drawn with probability ϵ. Gupta et al. [18] and Zhong et al. [19] provide an adaptive corruption model but assume that corruption strategy in round i is chosen before the algorithm chooses the arm to pull in round i. That is, the rewards for all arms are generated and then the adversary generates a corrupted rewards vector for the iteration. The

algorithm chooses an arm to pull *after* the corrupted rewards vector is generated [18, 19]. In all these settings, at each round of sampling, the algorithm can choose a uniformly random arm to pull. While the adversary can see all the rewards and the past pulled arms, if the current arm sampled is a uniformly random choice, there is implicitly a similar guarantee that no arm has too many corrupted samples with high probability. We show in Section II that this guarantee cannot be satisfied in the strongly-adaptive adversarial model. We also note that while Feng et al. [20] allows for adaptive adversaries, each arm has no knowledge of the realized rewards among other arms, which is a weaker setting than our model.

Furthermore, like Mukherjee et al. [16] and Altschuler et al. [21], we allow for arbitrary corruptions that are **not** bounded in magnitude. This is motivated by adversarial bandits applications like click fraud, fake ratings and email spam [17, 35, 36], where corruptions have no obvious bound in magnitude. For instance, in the click fraud setting, botnets can corrupt the click rate associated with ads. With enough parallelism and compute power, adversaries can create extremely large corruptions. Designing a meaningful bound on the corruption introduced by an adversary is often domain-specific and thus, arbitrary corruptions are an important setting to consider. This setting makes our model significantly stronger than prior works with strongly-adaptive adversaries [32, 33, 34].

B. Paper Overview

In Section II, we formalize the strongly-adaptive adversarial model for the adversarial best arm identification problem and the corresponding assumptions. Then, we define a naïve algorithm which solves the problem for previous adversarial models and show why it fails under our adversarial model. In Section III, we prove upper bounds on the optimality of any best arm identification algorithm in the strongly-adaptive adversarial setting. In Section IV, we first construct a simple robust sequential elimination algorithm that returns a near-optimal arm with constant probability. Then, we improve upon this algorithm to construct a robust algorithm which returns a near-optimal arm with high probability even in settings where the adversary can observe the algorithm's randomness. In Section V, we empirically evaluate the performance of our algorithms on synthetic and real-world datasets. Section VI discusses our conclusions and areas for future work.

II. Model and preliminaries

A. Strongly-Adaptive Adversarial Best Arm Identification

We consider the best arm identification problem with adversarial corruptions. In this problem, the algorithm has to choose among K arms, where each arm i has an unknown reward distribution with mean μ_i. We make minimal (and natural) assumptions on the reward distribution of these arms, which are further discussed in Section II-B.

To formally define our adversarial model, we first define a *stage* of sampling. We construct this as an analogue to a round of communication in the distributed systems world [22, 23, 24, 25, 26, 27]. In the distributed setting, we expect samples to be allocated *in batches* across servers with the goal of minimizing communication overhead. Thus, the algorithm first chooses a sampling strategy (e.g., allocating 100 samples, where each arm in contention is sampled with probability proportional to their current estimated reward), then receives the results for the batch (in our example, 100 samples). We denote this as a stage of sampling. In order to minimize communication overhead, we assume that at least one arm is eliminated after each stage.

Our adversarial model proceeds as follows in each stage i:

1) The adversary observes the sequence of arms that the algorithm pulls in Stage i and the reward (or feedback) received at each pull. We define a single pull and its corresponding feedback as a sample.
2) We define n_i as the total number of samples received by the algorithm in Stage i. The adversary can then arbitrarily change the feedback from the arms for up to $\lfloor \epsilon n_i \rfloor$ samples.
3) The algorithm observes the (possibly corrupted) samples from Stage i and eliminates some subset of arms.

We assume there exists some realization of samples from any arm in contention which would lead to the arm being eliminated.

Thus, our model defines a stage of a BAI algorithm such that at least one arm is eliminated at the end of each stage. We believe this is a realistic definition due to the communication overhead required in distributed settings. The class of sequential elimination algorithms (including successive elimination, action elimination, successive rejects, and racing algorithms [12, 14, 37, 38, 39, 40, 41]) for best arm identification naturally fits our adversarial model. These algorithms operate in stages or rounds, where some subset of arms is eliminated at the end of each stage [40]. However, we note that our adversarial model applies to the broader class of BAI algorithms, as well. If we consider other algorithms (e.g., Exp3 [3]) where arms are never fully eliminated until termination, the entire execution of the algorithm occurs in a single stage and our impossibility results would *remain unchanged*.

On the other hand, if we weaken our definition of a stage (e.g., define a stage as a single *sample*), then the adversary may never corrupt any samples; however, this would incur a significant communication overhead. In general, algorithms may define stages differently (e.g., not always requiring arms to be removed from contention) to trade off between communication overhead and sample complexity. Empirically, we show that even with a relaxed definition of stages where arms are never fully eliminated, we still observe weak performances indicating the significant increase in the strength of our adversarial model, compared to prior work. We discuss these results further in Section V-A.

B. Assumptions

Our algorithms rely on a few main assumptions, that we outline below.

1) **Minimum Gap:** All pairs of arms i and j satisfy $|\mu_i - \mu_j| > \Delta$ for some known Δ.
2) **Robust Estimator:** There exists an algorithm that can provide a robust mean estimate $\hat{\mu}$ where $|\hat{\mu} - \mu| \leq \Delta/2$ with probability at least $1 - \delta$ for any $\delta \in (0, 1)$

given $g(\Delta, \epsilon, \delta)$ samples from a distribution with mean μ with at most ϵ-fraction of corruptions for all $\epsilon < \epsilon_0$. Here, we define ϵ_0 as the breaking point of the robust mean estimator; if ϵ_0 fraction of samples are corrupted arbitrarily, there are no guarantees on the recovered mean. Note that such an estimator can only exist for $\epsilon_0 < 0.5$.

3) **Upper Bound on Corruption:** The algorithm knows an upper bound on ϵ, which represents the fraction of samples the adversary can corrupt.

Assumption 1 is a standard assumption made in the literature (see e.g., Audibert and Bubeck [12], Gupta et al. [18]). We observe that typically Δ is defined as the minimum gap between the optimal arms and the set of suboptimal arms, which is a looser requirement than ours. We relax this assumption in Section IV to mirror previous literature.

Assumption 2 allows us to generalize our work without relying on specific reward distributions. That is, we assume that we are in the *fully identifiable* best arm identification setting, where we can order the arms correctly with arbitrarily high probability given a sufficient number of samples [21]. We then abstract the sample complexity into a function $g(\Delta, \epsilon, \delta)$, which can vary based on the underlying reward distribution. For instance, in the setting where $\epsilon = 0$ and binary rewards, Even-Dar et al. [14] describes a simple naïve algorithm which uses $g(\Delta, 0, \delta) = \frac{4}{\Delta^2} \ln(\frac{2K}{\delta})$ samples per arm to identify the best arm. The extension of Assumption 2 to the adversarial setting with a constant fraction of corruptions is well-motivated by works on robust mean estimation that recover the true mean under a constant fraction of adversarial corruptions, with varying assumptions on the underlying distributions (e.g., sub-Gaussian distributions or distributions with bounded second moments) [42, 43, 44, 45, 46, 47]. For our experiments, we use the α-trimmed mean as our robust estimator which removes the top and bottom α-fraction of samples and returns the mean of the remaining samples [46]. For this to satisfy our robust estimator guarantees, we use the concentration guarantees of the Gaussian distribution as discussed in Appendix E-A.

Finally, Assumption 3 is a classic assumption in the literature, in order to bound the power of the adversary [16, 21].

With this context, we define a **fully corrupted arm** as an arm which is sampled s times and the adversary has corrupted at least $\lceil \epsilon_0 s \rceil$ samples. We consider any arm which has fewer than $\lceil \epsilon_0 s \rceil$ samples corrupted as an **uncorrupted arm** since a robust mean estimate can be recovered by Assumption 2. Finally, we define the underlying ordering of arms $G_{true} = [\mu_1, \mu_2 ... \mu_K]$ as an ordered list satisfying $G_{true}[i] > G_{true}[i+1]$ for all $i \in [1, K-1]$, μ_i is the mean of arm i, and the **rank** of $G_{true}[i] = i$. Thus, we can use the robust estimator to create an ordering G where $G = G_{true}$ with high probability in the uncorrupted setting. Finally, we denote $\epsilon_1 = \frac{\epsilon}{\epsilon_0}$.

C. A Naïve Best Arm Identification Algorithm

Assumption 2 states that there exists a subroutine which can distinguish between any distributions with means that are separated by at least Δ with high probability (at least $(1-\delta)$ for any $\delta \in (0,1)$) using $g(\Delta, \epsilon, \delta)$ samples per distribution. We can leverage this to construct a naïve algorithm, Algorithm 1.

Algorithm 1 Naïve Best Arm Identification

Input: K arms, robust estimator subroutine, and $g(\Delta, \epsilon, \delta)$.
Output: Arm with highest observed reward.

1) Sample arms uniformly at random until all K arms have $g(\Delta, \epsilon, \delta)$ samples per arm.
2) Create an ordering G of the arms by calculating a robust mean estimate $\hat{\mu}_i$ for each arm using our robust estimator.
3) Return the arm with the highest value of $\hat{\mu}_i$.

Algorithm 1 would return the optimal arm in expectation under previous adversarial models, with an expected sample complexity of $Kg(\Delta, \epsilon, \delta)$. That is, this algorithm would succeed against the adversarial models of Mukherjee et al. [16], Zhong et al. [19], and Altschuler et al. [21]; although, we note that they provide more efficient algorithms. In particular, under prior adversarial models, choosing a uniformly random arm to sample in each round ensures that each arm is corrupted (in expectation) an ϵ fraction of the time. Since we assume that $\epsilon < \epsilon_0$, Assumption 2 implies that there exists a robust mean estimation algorithm that can construct the correct ordering given $g(\Delta, \epsilon, \delta)$ samples per arm. Thus, in expectation, Algorithm 1 will return the best arm; we can simply increase the number of samples to provide a high probability guarantee ($\geq 1 - \delta_0$) for any $\delta_0 \in (0, 1)$.

However, Algorithm 1 cannot always find the optimal arm with high probability with a strongly-adaptive adversary. In particular, Algorithm 1 relies on the adversary's corruption strategy being *independent* of the algorithm's current choices which does not hold under a strongly-adaptive adversarial model. The algorithm's success depends on the premise that the adversary cannot corrupt any estimate $\hat{\mu}_i$ fully with high probability. If we assume that each arm has at most an ϵ-fraction of corrupted samples for $\epsilon < \epsilon_0$, we can use robust mean estimators to correctly order the arms. However, if even one arm is fully corrupted, then the mean estimate for the arm can be arbitrarily far from the true value and no algorithm can recover the true mean. Thus, the adversary can fully corrupt the arm with the lowest mean and ensure it is returned by Algorithm 1, which implies that the naïve algorithm provides no performance guarantees under a strongly-adaptive adversary.

III. Impossibility Results

In this section, we prove that it is impossible for any best arm identification algorithm to always recover any of the top $\left\lfloor \frac{\epsilon K}{1+\epsilon_0} \right\rfloor$ arms with high probability under our adversarial model.

To prove this, we first prove a preliminary lemma to show that there is no best arm identification algorithm that can recover the best arm with high probability under a strongly-adaptive adversarial model.

Lemma 1. *There exists $\delta_0 \in (0,1)$ such that there is no best arm identification algorithm that can return the arm with maximal reward with probability at least $1 - \delta_0$ in the strongly-adaptive adversary model as long as $K \geq \frac{\epsilon_0 + 1}{\epsilon}$.*

This proof is in Appendix A; we use this result to prove Theorem 1.

Theorem 1. *There exists $\delta_0 \in (0,1)$, such that there is no best arm identification algorithm that can return an arm with rank $\leq \left\lfloor \frac{\epsilon K}{1+\epsilon_0} \right\rfloor$ with probability at least $1-\delta_0$ in the strongly-adaptive adversarial model if*

$$K \geq \frac{\epsilon_0 + 1}{\epsilon}.$$

Proof. We define a "good" arm as an arm with rank $\leq \left\lfloor \frac{\epsilon K}{1+\epsilon_0} \right\rfloor$ in the underlying order G_{true}. We will prove this by induction on the number of "good" arms in contention. We consider our base case where there is one "good" arm. In this case, in Stage 1, there are at least $K = \frac{\epsilon_0+1}{\epsilon}$ arms in contention and by Lemma 1, we know there is no best arm identification algorithm that can recover the best arm with high probability.

Now, we assume that there are m "good" arms with $K = \lceil \frac{m(\epsilon_0+1)}{\epsilon} \rceil$. We assume a strong inductive hypothesis that there is no best arm identification algorithm that can recover one of the top m' arms out of K' arms in contention with high probability ($> 1-\delta_0$ for any $\delta_0 \in (0,1)$) for any $m' < m$ and $K' \geq \frac{m'(\epsilon_0+1)}{\epsilon}$ for the remainder of this argument. Again, we assume all the arms are sampled equally with n samples per arm since this is the worst-case setting for the adversary per Lemma 1.

The adversary can then fully corrupt

$$\begin{aligned}\left\lfloor \frac{\epsilon n K}{\lceil \epsilon_0 n \rceil} \right\rfloor &\geq \left\lfloor K \frac{\epsilon n}{1+\epsilon_0 n} \right\rfloor \\ &\geq \left\lfloor \frac{\epsilon K}{1+\epsilon_0} \right\rfloor \geq m\end{aligned}$$

arms in the first stage since $n \geq 1$. Thus, the adversary can fully corrupt *all* good arms in Stage 1.

Fact 1: By a simple generalization of Lemma 1, we note that if the adversary has enough budget to fully corrupt i arms in any stage of a best arm identification algorithm, then any algorithm that eliminates i arms in this stage eliminates the i best arms with constant probability. That is, since the corruptions are arbitrary, the adversary can place the i best arms anywhere in the perceived ordering; then, the best strategy for any algorithm is to eliminate i arms uniformly at random from the set of arms. Thus, any best arm identification that succeeds with high probability for *any* $\delta_0 \in (0,1)$ must assume i "good" arms were eliminated.

We consider two strategies for the algorithm:

1) After Stage 1, eliminate $\geq m$ arms.
2) After Stage 1, eliminate $< m$ arms.

Case 1: Since the adversary can place the m good arms arbitrarily in the observed ordering, the algorithm has no information about their placement. By Fact 1, this strategy will remove all the good arms with probability at least δ_0 for some $\delta_0 \in (0,1)$.

Case 2: Next, we consider the case where the algorithm eliminates $0 < i < m$ arms in the first stage. The adversary has arbitrarily placed the top m arms within the ordering and therefore the algorithm's best strategy is to eliminate i arms uniformly at random. However, by Fact 1, all i arms are good with constant probability. Hence, we must assume that we have $K - i$ arms left with $m - i$ good arms.

Now, after Stage 1, if we set $m' = m - i \geq 1$, we have m' good arms in contention and there are

$$K - i \geq \frac{m(\epsilon_0+1)}{\epsilon} - i > \frac{(m-i)(\epsilon_0+1)}{\epsilon}$$

arms in contention. Thus, since $m' \geq 1$, our claim holds from our inductive hypothesis.

Hence, if we have at least $K \geq \frac{\epsilon_0+1}{\epsilon}$ arms, then there is no best arm identification algorithm that can return an arm with rank $\leq \left\lfloor \frac{\epsilon K}{1+\epsilon_0} \right\rfloor$ with probability $1-\delta_0$ for any $\delta_0 \in (0,1)$. □

Theorem 1 shows why our adversarial model is significantly stronger than previously studied ones. We note that our theorem is a *lower* bound, which applies even if each arm is only sampled once per stage. In practical settings, we expect each arm to be sampled n times where $n \gg 1$ in each stage, since at least one arm must be eliminated at each stage. In this case, our impossibility result states that no BAI algorithm can find the top

$$\left\lfloor K \frac{\epsilon n}{1+\epsilon_0 n} \right\rfloor \approx \lfloor \epsilon_1 K \rfloor$$

arms for large n, where n represents the minimum number of times an arm is sampled in any stage of the algorithm. In the real-world dataset analyzed in Section V,

$$n = g(\Delta, \epsilon, \delta) \geq \frac{1}{\Delta^2} \approx 10,000.$$

IV. Robust Sequential Elimination Algorithms

In this section, we design an algorithm that can return a near-optimal arm with high probability. We complement our impossibility results in Section III, by constructing an algorithm that recovers an arm with rank $\leq \lceil (1+\lambda)\lfloor \epsilon_1 K \rfloor \rceil$ with high probability for $\lambda = \frac{1-2\epsilon_1-\sqrt{1-4\epsilon_1}}{2\epsilon_1}$.

A. A Simple Algorithm

We first construct a single-stage baseline algorithm (Algorithm 2) which makes a small modification to Algorithm 1 for a significant increase in the expected rank of the returned arm. In particular, we choose a constant c where the choice of c allows us to trade off between the probability of returning an uncorrupted arm and the optimality guarantee of the uncorrupted arms.

Algorithm 2 Single-Stage Best Arm Identification

Input: K arms, robust estimator subroutine and additional parameters: $g(\Delta, \epsilon, \delta)$, ϵ, ϵ_0, c.
Output: Near-optimal arm with nearly constant probability ($\geq (1-1/c)(1-\delta_0)$).

1) Sample all K arms $g(\Delta, \epsilon, \delta)$ times per arm.
2) Create an ordering G of the arms by calculating a robust mean estimate $\hat{\mu}_i$ for each arm using our robust estimator.
3) Return a random arm chosen from the top $\lceil c\lfloor \epsilon_1 K \rfloor \rceil$ arms.

Theorem 2. *Algorithm 2 returns an arm with rank* $\leq \lceil (c+1)\lfloor \epsilon_1 K \rfloor \rceil$ *with probability* $\geq (1-1/c)(1-\delta_0)$ *for* $c > 1$*, where* δ_0 *is the probability that any robust estimator invocation fails.*

Proof. With probability $(1-\delta_0)$, the robust estimator produces a correct robust estimate of all the arms that are not fully corrupted in every stage of the algorithm. For the rest of this argument, we assume that the robust estimator always produces the correct robust estimate.

Denote $n = g(\Delta, \epsilon, \delta)$ to represent the number of samples per arm. The adversary can fully corrupt at most

$$\left\lfloor \frac{\epsilon n K}{\lceil \epsilon_0 n \rceil} \right\rfloor \leq \lfloor \epsilon_1 K \rfloor$$

arms. Thus, in the ordering G, the top $\lceil c\lfloor \epsilon_1 K \rfloor \rceil$ arms (for $c > 1$) contain at most $\lfloor \epsilon_1 K \rfloor$ fully corrupted arms. A randomly chosen arm from this set has probability at most $1/c$ of being fully corrupted; in this case, we have no guarantee about the rank of the returned arm.

However, with probability at least $1 - 1/c$, the chosen arm is not fully corrupted. In this case, the remaining arms must be correctly ranked, which implies they have rank at most $\lceil (c+1)\lfloor \epsilon_1 K \rfloor \rceil$; we note that the $(c+1)$ factor is due to the fact that the top $\lfloor \epsilon_1 K \rfloor$ arms can be arbitrarily corrupted to the bottom of the ordering, while the remaining arms are correctly ranked. Thus, this implies that we return an arm with rank at least $\lceil (c+1)\lfloor \epsilon_1 K \rfloor \rceil$ with probability at least $(1-1/c)(1-\delta_0)$, which completes our proof. □

Thus, Algorithm 2 matches the expected sample complexity of Algorithm 1 and returns a near-optimal arm with near-constant probability for a fixed value of c.

We note that we can bound δ_0 in our high probability guarantee in terms of $1-\delta$ (the probability the robust estimator succeeds in a single iteration). In general, if the robust estimator succeeds on a single arm with probability $(1-\delta)$, then it succeeds on all K arms with probability at least $(1-\delta)^K$. In this setting, with a single stage, this implies that $\delta_0 = 1-(1-\delta)^K$. In the general case, the estimator succeeds on all arms in all rounds with probability $\geq (1-\delta)^{K^2}$, since there are at most K rounds in any sequential elimination algorithm. Thus, $\delta_0 \leq 1-(1-\delta)^{K^2}$ for any sequential elimination algorithm. While this appears large, our experiments use $\delta = 10^{-12}$; with at most 1000 arms, this allows us a δ_0 of 10^{-7} overall.

Next, we improve upon this to provide a high probability guarantee.

B. A Robust Sequential Elimination Algorithm

A key component of our adversary's power relies on having a large number of arms in contention, and thus a large adversarial budget. We use this insight to design Algorithm 3, which reduces the set of arms in contention while ensuring that most of the arms satisfy our definition of optimality.

We first prove a few preliminary lemmas about the guarantees of Algorithm 3.

Lemma 2. *Algorithm 3 will terminate and return an arm* i *with probability* $\geq 1-\delta_0$*, where* δ_0 *is the probability that our robust estimator fails at some point in the algorithm's execution,* $0 \leq \epsilon_1 = \epsilon/\epsilon_0 \leq 0.25$*,* $c = 1+\lambda$*, and*

$$\lambda = \frac{1-2\epsilon_1-\sqrt{1-4\epsilon_1}}{2\epsilon_1}.$$

Algorithm 3 Robust Best Arm Identification

Input: K arms, robust estimator subroutine and additional parameters: $g(\Delta, \epsilon, \delta)$, ϵ, ϵ_0, c, Δ.

Output: Near-optimal arm with high probability ($\geq 1-\delta_0$).

1) Add all K arms to a set s_1 which are in contention for the best arm.
2) Set $i = 1$.
3) Repeat:
   a) Sample each arm in s_i $g(\Delta, \epsilon, \delta)$ times.
   b) Create an ordering G_i of the arms by calculating a mean estimate $\hat{\mu_j}$ for each arm j using our robust estimator.
   c) If $i > 1$, using our robust estimator, check if any arm j shows a distribution shift (change in mean of $> \Delta/2$) between $\hat{\mu_j}$ from any previous stage of sampling. Remove any arm with a distribution shift from the ordering G_i.
   d) If $f_i = \left\lfloor \frac{\epsilon |s_i|}{\epsilon_0} \right\rfloor < 1$, return the best arm from G_i.
   e) Create a new set s_{i+1} consisting of the top $\lceil cf_i \rceil$ arms in G_i that are still in contention.
   f) Set $i = i+1$.

The proof of Lemma 2 is provided in Appendix B.

Corollary 1. *The arm* i^* *returned by Algorithm 3 has never been fully corrupted through any stage of sampling with probability* $\geq (1-\delta_0)$ *where* δ_0 *is the probability that our robust estimator fails at some point in the algorithm's execution,* $0 \leq \epsilon_1 = \epsilon/\epsilon_0 \leq 0.25$*,* $c = 1+\lambda$*, and*

$$\lambda = \frac{1-2\epsilon_1-\sqrt{1-4\epsilon_1}}{2\epsilon_1}.$$

The proof of Corollary 1 is in Appendix C.

We then use these to prove the following performance guarantee for Algorithm 3.

Theorem 3. *Regardless of the adversary's strategy, Algorithm 3 returns an arm with rank* $\leq \lceil (1+\lambda)\lfloor \epsilon_1 K \rfloor \rceil$ *with probability* $\geq 1-\delta_0$ *where* δ_0 *is the probability that any robust estimator invocation fails,* $0 \leq \epsilon_1 = \epsilon/\epsilon_0 \leq 0.25$*,* $c = 1+\lambda$*, and*

$$\lambda = \frac{1-2\epsilon_1-\sqrt{1-4\epsilon_1}}{2\epsilon_1}.$$

Proof. With probability $(1-\delta_0)$, the robust estimator produces a correct robust estimate of all the arms that are not fully corrupted in every stage of the algorithm. For the rest of this argument, we assume that the robust estimator always produces the correct robust estimate.

We define G_1 as the corrupted ordering that the adversary can create from G_{true} after Stage 1 of sampling. By Lemma 2,

we know the loop in lines (a)-(f) will terminate and return an arm i with probability at least $1-\delta_0$. By Corollary 1, we know that the arm that is returned was never fully corrupted. In Stage 1, we assume the adversary corrupts x_1 arms "up" into the top $\lceil c\lfloor\epsilon_1 K\rfloor\rceil$ ranks and x_2 arms "down" into the rest of the ordering. For instance, a sample execution is shown below with $K=12$, $\epsilon_1=0.25$, $c=2$, $x_1=2$ and $x_2=1$.

$$\begin{aligned} G_{true} &= \boxed{1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10,\ 11,\ 12} \\ G_1 &= \boxed{12,\ 11,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10,\ 1} \\ s_2 &= \boxed{12,\ 11,\ 2,\ 3,\ 4,\ 5} \end{aligned}$$

We observe that there are $\lceil c\lfloor\epsilon_1 K\rfloor\rceil - x_1$ uncorrupted arms in s_2. From Lemma 2, we observe that the adversary can corrupt at most $\lceil (c-1)\lfloor\epsilon_1 K\rfloor\rceil - 1$ new arms until the algorithm terminates. Thus, at the end of the algorithm, there are at least $\lfloor\epsilon_1 K\rfloor - x_1 + 1$ arms which have never been corrupted. We observe that since these arms were never corrupted, they either remain in contention at the end of the algorithm's execution or are pruned for being low-rank. In either case, these arms are always correctly ordered and thus, our algorithm returns the best ranked arm among them. The uncorrupted arms in contention have rank between x_2+1 and $\lceil c\lfloor\epsilon_1 K\rfloor\rceil + x_2 - x_1$. Thus, the adversary returns an arm of rank at most

$$\begin{aligned} &(\lceil c\lfloor\epsilon_1 K\rfloor\rceil + x_2 - x_1) - (\lfloor\epsilon_1 K\rfloor - x_1 + 1) + 1 \\ &\leq \lceil (1+\lambda)\lfloor\epsilon_1 K\rfloor\rceil, \end{aligned}$$

since $x_2 \leq \lfloor\epsilon_1 K\rfloor$ and $c = 1+\lambda$, completing our proof. □

Theorem 3 shows that we can still apply sequential arm elimination in order to guarantee *near-optimal* best arm identification even for a strongly-adaptive adversary. We note that for small $\epsilon \ll \epsilon_0$, λ tends to 0; thus, our construction returns an arm with rank $\approx \lceil\epsilon_1 K\rceil + 1$ with high probability, which is nearly tight with our impossibility result.

However, we note that our definition of Δ corresponding to the minimum pairwise gap between all possible arms is quite strong and could lead to unnecessarily large sample complexity. In fact, intuitively, the robust estimator only needs to distinguish between arms that satisfy our optimality condition and arms that do not. This aligns well with previous work (e.g., Audibert and Bubeck [12], Gupta et al. [18]) where the minimum gap is defined as the gap between the best and second-best arm. This insight leads to a much simpler requirement on Δ.

Proposition 1. *Regardless of the adversary's strategy, Algorithm 3 returns an arm of rank* $\leq \lceil (2+\lambda)\lfloor\epsilon_1 K\rfloor\rceil$ *with probability* $\geq 1-\delta_0$ *if we use a minimum gap of* $\Delta_* = \mu_t - \mu_{t+1}$ *where* $t = \lceil (2+\lambda)\lfloor\epsilon_1 K\rfloor\rceil$, δ_0 *is the probability that any robust estimator invocation fails,* $0 \leq \epsilon_1 = \epsilon/\epsilon_0 \leq 0.25$, $c = 1+\lambda$, *and*

$$\lambda = \frac{1 - 2\epsilon_1 - \sqrt{1-4\epsilon_1}}{2\epsilon_1}.$$

This proof is provided in full in Appendix D.

Intuitively, when we let Δ be the pairwise gap between arms, our robust estimator guarantees that all uncorrupted arms are ordered correctly. That is, in a setting with no corruptions, we will always return the best arm. With $\Delta = \Delta_*$, we make no guarantees on the ordering of the top t arms; instead we only require that the top t arms are separated from the remaining arms in contention, which weakens the optimality guarantee of the returned arm. However, depending on the distribution of rewards, $\Delta = \Delta_*$ can be more practical to use, since it requires less knowledge of the rewards of all the arms and can significantly reduce sample complexity.

One key distinction between this algorithm and prior work is the use of distribution shifts to allow the algorithm to recover from significant shifts in the observed ordering of arms in each stage. Previous algorithms can rely on weaker adversarial models to safely eliminate suboptimal arms much faster [16, 21]. However, the convergence of our algorithm relies solely on the budget reduction of the adversary between stages. In fact, Algorithm 3 will satisfy our guarantees even in the setting where the adversary *can observe the internal randomness* of the algorithm, as long as the adversary's budget is limited per stage. To the best of our knowledge, past work on adaptive adversaries (including Algorithms 1 and 2) requires unobserved randomness for any convergence guarantees [17, 19, 21].

Finally, we analyze the sample complexity of our algorithm.

Theorem 4. *Algorithm 3 has a sample complexity of*

$$\leq \frac{g(\Delta,\epsilon,\delta)(K+\epsilon_0/\epsilon-1)(K-\epsilon_0/\epsilon+2)}{2}$$

where δ *is chosen to guarantee that the probability that any robust estimator invocation fails is at most* δ_0.

Proof. We note that the sampling done in each stage of the algorithm is the same; we simply sample each arm in contention $g(\Delta,\epsilon,\delta)$ times and recover the robust mean estimate. Thus, at any given stage i of the algorithm, there are $|s_i|g(\Delta,\epsilon,\delta)$ samples required.

We observe that each stage eliminates at least one arm and thus, $|s_i| \leq K-(i-1)$. Further, we observe that the algorithm terminates when $f_i < 1$. This occurs when

$$|s_i| < \frac{\epsilon_0}{\epsilon} \iff i > K + 1 - \frac{\epsilon_0}{\epsilon}.$$

Thus, we can bound our sample complexity as

$$\begin{aligned} &g(\Delta,\epsilon,\delta) \sum_{i=1}^{K+2-\frac{\epsilon_0}{\epsilon}} |s_i| \\ &= \frac{g(\Delta,\epsilon,\delta)(K+\epsilon_0/\epsilon-1)(K-\epsilon_0/\epsilon+2)}{2}, \end{aligned}$$

matching our bound above. □

V. Experimental Results

We evaluate Algorithms 2 and 3 on both synthetic and real-world datasets. Throughout this section, we use the α-trimmed mean estimator as our robust estimator subroutine with $\alpha = \epsilon_0 = 0.49$. We measure the performance of our algorithms using the average rank of the returned arm. We

run 250 trials for each value of K and ϵ and we normalize the rank of the returned arm. In terms of sample complexity, previous work shows that $O(\Delta^{-2}\log\log\Delta^{-2})$ samples are required to distinguish between two arms which are separated by a gap of Δ [48]. We observe that this is a lower bound for distinguishing between a *single pair* of arms. The simple algorithm described in Even-Dar et al. [14] requires $O(\frac{\ln 2K/\delta}{\Delta^2})$ samples per arm to identify the best arm in a single round. In the distributed setting, we choose to use Even-Dar's sample complexity for $g(\Delta,\epsilon,\delta)$ with $\delta = 10^{-12}$, since we cannot allocate samples adaptively within a stage. Further details about the experimental setup and the design of the robust estimator are given in Appendix E.

A. Baselines

We first compare the performance of our algorithm to prior work. We choose a specific adversarial strategy, where the adversary chooses to corrupt samples from the lowest ranked arms, making their feedback arbitrarily large in Stage 1. In this setting, Algorithm 1 (which returns a near-optimal arm under prior adversarial models) will consistently return the arm with rank K (the lowest mean) regardless of $g(\Delta,\epsilon,\delta)$ due to the increased strength of our adversarial model; thus, we expect previous algorithms to have similar performances. We implement 4 baseline algorithms as described below:

1) Mukherjee et al. [16] Algorithm 2: This was designed for a non-adaptive adversary and allows for arbitrary corruptions. The performance guarantee is to return the optimal arm with probability $\geq 1-\delta$ for any $\delta \in (0,1)$.
2) Altschuler et al. [21] Algorithm 3: This was designed for an adaptive adversary with arbitrary corruptions. The performance guarantee is to return the optimal arm with probability $\geq 1-\delta$ for any $\delta \in (0,1)$.
3) Zhong et al. [19] Algorithm 1: This was designed for an adaptive adversary with bounded corruptions. The performance guarantee is to return an ϵ-optimal arm with probability $\geq 1-\delta$ for $\delta = \exp(-\Theta(T))$ and $\epsilon = O(\frac{C\log K}{T})$, where T is the horizon and C is the corruption budget.
4) Auer et al. [3] Exp3: This was designed for a non-adaptive setting, where each arm is assigned an arbitrary sequence of rewards before the algorithm executes. Exp3 is *not* a sequential elimination algorithm since arms are never eliminated from contention; we analyze Exp3 by defining a stage as a fixed number of samples (for our experiments, we use 40 samples per stage). Exp3 guarantees that the expected regret (compared to the single best arm) is $O(\sqrt{G^* K\ln K})$, where G^* is the total reward from the best arm over the time horizon.

Our results across these baselines are summarized in Figure 1. We first observe that with $\epsilon \geq 0.04$, all the baselines consistently return the worst possible arm. This indicates that their performance guarantees on weaker adversarial models do not hold against strongly-adaptive adversaries. We further note that Exp3 performs competitively at $\epsilon = 0.02$. However, this is due to the weakened definition of a stage. Unlike the other baselines, Exp3 does not eliminate any arms at the end of a stage; instead, a stage is simply defined as 40 samples. Thus, for $\epsilon \leq 0.02$, the adversary cannot corrupt a sufficient number of samples and the algorithm performs well. However, as the adversary's budget increases (or the number of samples per stage increases), we observe that Exp3 also consistently returns one of the worst arms.

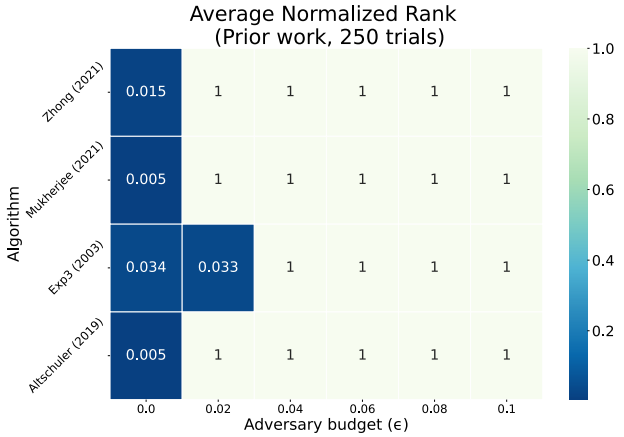


Fig. 1: Experiments with baseline algorithms on Gaussian synthetic data under the strongly-adaptive adversarial model. Results are averaged over 250 trials with 200 arms. All the baselines return one of the worst arms under the strongly-adaptive adversarial model for $\epsilon \geq 0.04$. All standard deviations are ≤ 0.05.

Since the prior baselines are not designed for our adversarial model, we compare our algorithms directly to each other.

B. Adversarial Strategies

For our algorithms, we investigate two possible adversarial strategies. We assume that the true ordering of arms is $G_{true} = [\mu_1, \mu_2, ...\mu_K]$, which satisfies $\mu_i > \mu_{i+1}$ for all $i \in [1, K-1]$. We define G_i as the perceived ordering after adversarial corruptions in Stage i and we assume the adversary initially has the budget to corrupt at most f_i arms in Stage i.

We observe that the best outcome for the adversary occurs when the algorithm returns the arm with the highest rank. An intuitive first adversarial strategy would try to keep the set of highest ranked arms in contention as long as possible. However, from Theorem 3, we know that Algorithm 3 will eventually return an originally uncorrupted arm and eventually all the high-ranked arms will be removed from contention. In fact, we tested this strategy empirically and Algorithm 3 consistently returned the optimal arm. Thus, we consider a variant of this strategy, where the adversary introduces as many high-ranked arms into contention as possible in Stage 1. However, in future stages, the adversary uses the remaining budget to eliminate as many low-ranked arms from contention as possible. We also consider an "optimistic" setting, where the adversary simply removes as many of the best arms from contention as possible in each stage. We formalize this as follows:

- **Strategy 1 (Aggressive):** The adversary chooses the worst f_1 arms to corrupt and places them in the lowest ranks in Stage 1. In every subsequent stage i, the adversary corrupts the best f_i arms that were not previously corrupted.

- **Strategy 2 (Optimistic):** At each stage i, the adversary chooses the best f_i arms to corrupt and places them in the highest ranks.

Throughout this section, we compare Algorithm 3 to Algorithm 2. Algorithm 2 provides a reasonable performance guarantee, where we have a nearly constant probability of returning a near-optimal arm. Moreover, we note that in more optimistic settings, Algorithm 2 can provide comparable performance, while reducing sample complexity and providing convergence in a single stage. However, Algorithm 2 will also fail with probability 1 if we assume the adversary has access to the algorithm's internal randomness. For the rest of this section, we investigate the weaker adversarial setting, where the algorithm has independent randomness, although this is not required for Algorithm 3's convergence.

C. Synthetic Experiments

We first evaluate our algorithms on synthetic data. We construct datasets consisting of K arms, where the reward for arm i follows a Gaussian distribution with mean μ_i, and generate data such that

$$\mu_{i-1} - \mu_i = \Delta \quad \forall i \in [2, K].$$

Throughout this section, we use $\Delta = 1$ and we experiment with $K \in [100, 1000]$. For Algorithm 2, we use $c = 2$, so that we return a high-rank arm with probability ≈ 0.5. For Algorithm 3, we solve for c using Theorem 3. We vary $\epsilon \in [0.01, 0.1]$.

We first analyze the results for Strategy #1 as summarized in Figure 2. Algorithm 2 consistently returns an arm with rank ≈ 0.45, which matches our expectations, while Algorithm 3 consistently returns arms of rank at most 0.07 for $\epsilon \leq 0.1$. This indicates that under the aggressive adversarial strategy, Algorithm 3 significantly outperforms our baseline and returns a near-optimal arm for all $\epsilon \leq 0.1$. Furthermore, we also compute the average increase in sample complexity between Algorithm 2 and Algorithm 3. The increase in sample complexity is relatively small ($\leq 1.5\times$ for $\epsilon \leq 0.1$), which indicates that most of the sample complexity is due to the first stage of our algorithm which provides an initial ordering.

We then consider Strategy #2; our results are summarized in Figure 3. We note that Algorithm 3 outperforms Algorithm 2 for all values of ϵ; however, the margins are much smaller than under Strategy #1. Intuitively, we observe that the adversary simply removes the best arms from contention in Stage 1. Thus, Algorithm 2 simply returns a random arm within the top $\lceil 2f_1 \rceil$ ranks, which has rank $\leq \lceil 3f_1 \rceil$. This is only slightly worse than the theoretical performance guarantees of Algorithm 3, which returns an arm with rank $\leq \lceil cf_1 \rceil$, where c satisfies Theorem 3. For large ϵ, c tends to 2 and the performance gap is relatively small. The sample complexity results for Strategy #2 are similar to Strategy #1 ($\leq 1.5\times$ increase).

For completeness, we also provide the distributions of the returned arms for each strategy in Appendix E.

D. PKIS2 Experiments

Following the work of Mukherjee et al. [16], we analyze our algorithm's performance on the PKIS2 dataset, which is a collection of protein kinase and kinase inhibitors. We consider the best arm identification problem of finding the best inhibitor for ACVRL1 [49]. After the data is appropriately normalized and transformed, we have $K = 63$ distinct inhibitors to test. Further details about the dataset and transformations are available in Appendix E.

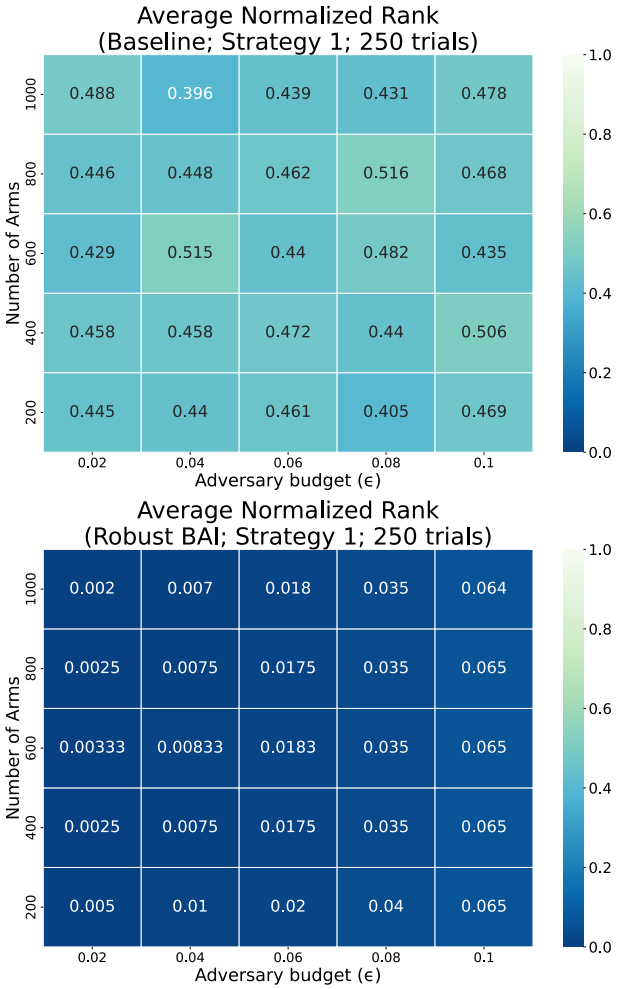


Fig. 2: **Top:** The average normalized rank of the returned arm under Algorithm 2. We observe that the average rank ranges between 0.39 and 0.52 for all values of ϵ. **Bottom:** The average normalized rank of the returned arm under Algorithm 3. We observe that the average rank ranges between 0.002 and 0.07 for all values of ϵ.

Again, we first analyze Strategy #1. Our results are summarized in Figure 4. For $\epsilon \leq 0.1$, Algorithm 3 significantly outperforms Algorithm 2, mirroring the synthetic dataset. Algorithm 3 returns arms with rank ≤ 0.1, while Algorithm 2 returns arms with rank ≈ 0.45. This indicates that our algorithm's performance translates to real-world data. We note that the increase in sample complexity is $\leq 1.5\times$, matching our observations on the synthetic dataset.

We then consider Strategy #2 on the PKIS2 dataset; the results are summarized in Figure 5. Again, we see a much smaller gap between Algorithm 2 and Algorithm 3 under

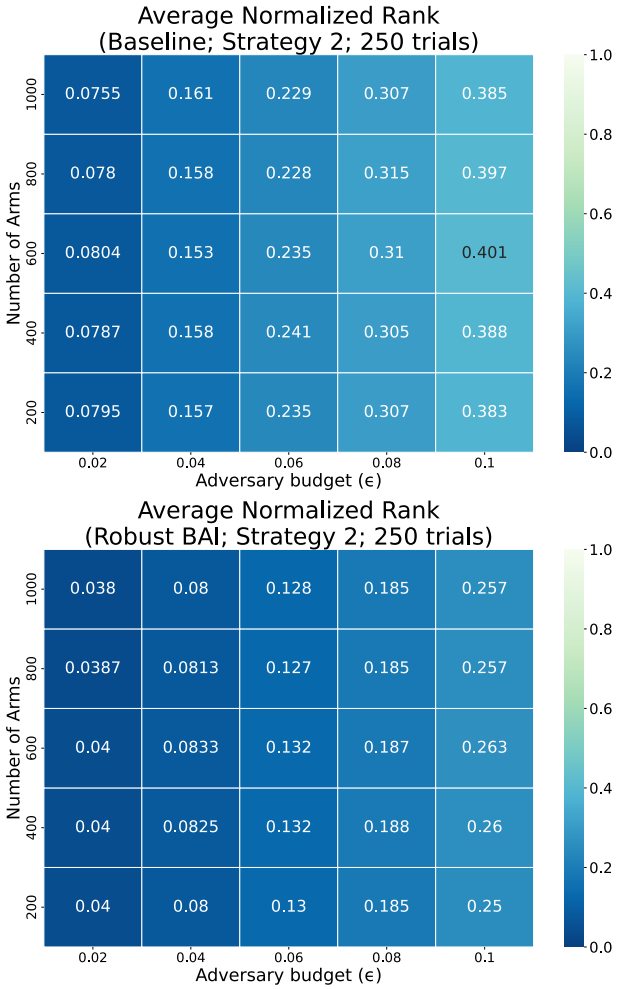


Fig. 3: Average normalized rank of returned arm under Algorithm 2 (**top**) and under Algorithm 3 (**bottom**) with Strategy #2. For all $\epsilon \leq 0.1$, Algorithm 3 outperforms Algorithm 2 although with a smaller margin than Strategy #1.

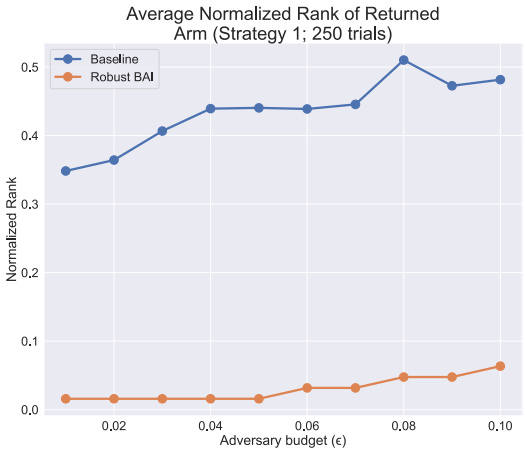


Fig. 4: Experiments with PKIS2 data with adversarial strategy #1. Algorithm 3 returns a low-rank arm (≤ 0.1) and outperforms Algorithm 2 for all values of ϵ.

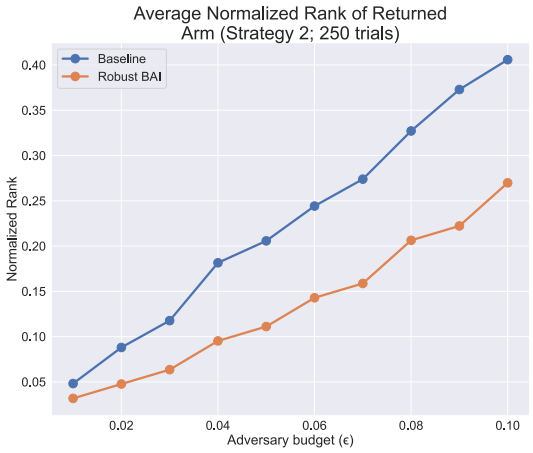


Fig. 5: Experiments with PKIS2 data with adversarial Strategy #2. Algorithm 3 consistently outperforms Algorithm 2 for $\epsilon \leq 0.1$.

this adversarial setting. However, robust BAI consistently outperforms the baseline and returns an arm of rank ≤ 0.3 even at $\epsilon = 0.1$. The sample complexity results are similar to Strategy #1.

VI. Discussion and Conclusions

a) Future Work: We note that our algorithms are not necessarily sample efficient, since the exploration across arms is uniform in each phase. For instance, we observe that we can improve upon the sample complexity by randomly sampling a small subset of arms after Stage 1 and bounding the probability that there are enough "good" arms remaining in contention to provide a high-probability guarantee of returning a near-optimal arm. However, it is still an interesting area of future research to further improve on the sample complexity guarantees under the strongly-adaptive adversarial model. For instance, while Algorithm 3 outperforms Algorithm 2 significantly in Strategy #1, there is a smaller performance difference in more optimistic settings such as Strategy #2. Designing an algorithm which minimizes sample complexity under weaker adversarial settings like Strategy #2 while simultaneously providing robustness against stronger adversarial settings (e.g., to provide best-of-both-worlds or data-dependent regret guarantees such as Abbasi-Yadkori et al. [50], Wei and Luo [51], Ito [52], Ito et al. [53]) is an important avenue for future work.

b) Contributions: We conclude with a summary of our contributions. We define a novel strongly-adaptive adversarial model with arbitrary corruptions which is significantly stronger than the adversarial models considered in prior work and matches real-world adversarial capabilities in distributed settings. We show that with minimal restrictions on K, the best arm can be corrupted beyond recovery and we formally prove an impossibility result for the best $\left\lfloor \frac{\epsilon}{1+\epsilon_0} \right\rfloor$ fraction of arms for any best arm identification algorithm. We then provide a novel sequential elimination algorithm that recovers an arm of rank $\leq \left\lceil (1+\lambda)\left\lfloor \frac{\epsilon K}{\epsilon_0} \right\rfloor \right\rceil$, for $\lambda = \frac{1-2\epsilon-\sqrt{1-4\epsilon_1}}{2\epsilon_1}$, with high

probability under our strongly-adaptive adversarial model. In fact, our algorithm provides robustness even if the adversary can observe the algorithm's internal randomness. Finally, we evaluate our algorithm on synthetic and real-world datasets and empirically demonstrate its robustness against a strongly-adaptive adversary.

REFERENCES

[1] H. E. Robbins, "Some aspects of the sequential design of experiments," *Bulletin of the American Mathematical Society*, vol. 58, pp. 527–535, 1952.

[2] A. Slivkins, "Introduction to multi-armed bandits," *CoRR*, vol. abs/1904.07272, 2019. [Online]. Available: http://arxiv.org/abs/1904.07272

[3] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM J. Comput.*, vol. 32, no. 1, p. 48–77, Jan 2003. [Online]. Available: https://doi.org/10.1137/S0097539701398375

[4] D. Bouneffouf, I. Rish, and C. Aggarwal, "Survey on applications of multi-armed and contextual bandits," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, 2020, pp. 1–8.

[5] K. Misra, E. M. Schwartz, and J. Abernethy, "Dynamic Online Pricing with Incomplete Information Using Multiarmed Bandit Experiments," *Marketing Science*, vol. 38, no. 2, pp. 226–252, March 2019. [Online]. Available: https://ideas.repec.org/a/inm/ormksc/v38y2019i2p226-252.html

[6] S. S. Villar, J. Bowden, and J. Wason, "Multi-armed Bandit Models for the Optimal Design of Clinical Trials: Benefits and Challenges," *Statistical Science*, vol. 30, no. 2, pp. 199 – 215, 2015. [Online]. Available: https://doi.org/10.1214/14-STS504

[7] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation." in *WWW*, M. Rappa, P. Jones, J. Freire, and S. Chakrabarti, Eds. ACM, 2010, pp. 661–670. [Online]. Available: http://dblp.uni-trier.de/db/conf/www/www2010.html#LiCLS10

[8] A. Anandkumar, N. Michael, A. K. Tang, and A. Swami, "Distributed algorithms for learning and cognitive medium access with logarithmic regret," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 731–745, 2011.

[9] J. Jiang, R. Das, G. Ananthanarayanan, P. A. Chou, V. Padmanabhan, V. Sekar, E. Dominique, M. Goliszewski, D. Kukoleca, R. Vafin, and H. Zhang, "Via: Improving internet telephony call quality using predictive relay selection," in *Proceedings of the 2016 ACM SIGCOMM Conference*, ser. SIGCOMM '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 286–299. [Online]. Available: https://doi.org/10.1145/2934872.2934907

[10] S. Mannor and J. N. Tsitsiklis, "The sample complexity of exploration in the multi-armed bandit problem," *J. Mach. Learn. Res.*, vol. 5, p. 623–648, Dec 2004.

[11] S. Bubeck, R. Munos, and G. Stoltz, "Pure exploration in multi-armed bandits problems," in *Proceedings of the 20th international conference on Algorithmic learning theory*, 2009, pp. 23–37.

[12] J.-Y. Audibert and S. Bubeck, "Best arm identification in multi-armed bandits," in *Conference on Learning Theory*, 2010.

[13] Y. Rekhter and T. Li, "Rfc1771: A border gateway protocol 4 (bgp-4)," 1995.

[14] E. Even-Dar, S. Mannor, and Y. Mansour, "Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems," *Journal of Machine Learning Research*, vol. 7, pp. 1079–1105, 2006.

[15] H. Zhang and C. Shen, "Best arm identification for both stochastic and adversarial multi-armed bandits," in *2018 IEEE Information Theory Workshop (ITW)*, 2018, pp. 1–5.

[16] A. Mukherjee, A. Tajer, P.-Y. Chen, and P. Das, "Mean-based best arm identification in stochastic bandits under reward contamination," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 9651–9662. [Online]. Available: https://proceedings.neurips.cc/paper/2021/file/500ee9106e0e4d8f769fadfdf9f2837e-Paper.pdf

[17] T. Lykouris, V. Mirrokni, and R. Paes Leme, "Stochastic bandits robust to adversarial corruptions," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, ser. STOC 2018. New York, NY, USA: Association for Computing Machinery, 2018, p. 114–122. [Online]. Available: https://doi.org/10.1145/3188745.3188918

[18] A. Gupta, T. Koren, and K. Talwar, "Better algorithms for stochastic bandits with adversarial corruptions," in *Proceedings of the Thirty-Second Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, A. Beygelzimer and D. Hsu, Eds., vol. 99. PMLR, 25–28 Jun 2019, pp. 1562–1578. [Online]. Available: https://proceedings.mlr.press/v99/gupta19a.html

[19] Z. Zhong, W. C. Cheung, and V. Tan, "Probabilistic sequential shrinking: A best arm identification algorithm for stochastic bandits with corruptions," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 12 772–12 781. [Online]. Available: https://proceedings.mlr.press/v139/zhong21a.html

[20] Z. Feng, D. C. Parkes, and H. Xu, "The intrinsic robustness of stochastic bandits to strategic manipulation," in *International Conference on Machine Learning*, 2019.

[21] J. Altschuler, V.-E. Brunel, and A. Malek, "Best arm identification for contaminated bandits," *Journal of Machine Learning Research*, vol. 20, no. 91, pp. 1–39, 2019. [Online]. Available: http://jmlr.org/papers/v20/18-395.html

[22] P. Feldman and S. Micali, "Optimal algorithms for byzantine agreement," in *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, ser. STOC '88. New York, NY, USA: Association for Computing Machinery, 1988, p. 148–161. [Online]. Available: https://doi.org/10.1145/62212.62225

[23] C. Dutta, G. Pandurangan, R. Rajaraman, Z. Sun, and E. Viola, *On the Complexity of Information Spreading in Dynamic Networks*, 2013, pp. 717–736. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611973105.52

[24] A. Lewko and M. Lewko, "On the complexity of asynchronous agreement against powerful adversaries," in *Proceedings of the 2013 ACM Symposium on Principles of Distributed Computing*, ser. PODC '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 280–289. [Online]. Available: https://doi.org/10.1145/2484239.2484250

[25] S. Goldwasser, Y. T. Kalai, and S. Park, "Adaptively secure coin-flipping, revisited," in *Automata, Languages, and Programming*, M. M. Halldórsson, K. Iwama, N. Kobayashi, and B. Speckmann, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 663–674.

[26] I. Abraham, T.-H. H. Chan, D. Dolev, K. Nayak, R. Pass, L. Ren, and E. Shi, "Communication complexity of byzantine agreement, revisited," in *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, ser. PODC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 317–326. [Online]. Available: https://doi.org/10.1145/3293611.3331629

[27] I. Abraham, S. Devadas, D. Dolev, K. Nayak, and L. Ren, "Synchronous byzantine agreement with expected O(1) rounds, expected O(n^2) communication, and optimal resilience," in *Financial Cryptography and Data Security*, I. Goldberg and T. Moore, Eds. Cham: Springer International Publishing, 2019, pp. 320–334.

[28] M. Hashemi, K. Swersky, J. A. Smith, G. Ayers, H. Litz, J. Chang, C. Kozyrakis, and P. Ranganathan, "Learning memory access patterns," *CoRR*, vol. abs/1803.02329, 2018. [Online]. Available: http://arxiv.org/abs/1803.02329

[29] C.-Y. Hsu, P. Indyk, D. Katabi, and A. Vakilian, "Learning-based frequency estimation algorithms," in *ICLR*, 2019.

[30] T. Eden, P. Indyk, S. Narayanan, R. Rubinfeld, S. Silwal, and T. Wagner, "Learning-based support estimation in sublinear time," *arXiv preprint arXiv:2106.08396*, 2021.

[31] K. Jun, L. Li, Y. Ma, and X. Zhu, "Adversarial attacks on stochastic bandits," *CoRR*, vol. abs/1810.12188, 2018. [Online]. Available: http://arxiv.org/abs/1810.12188

[32] S. Zuo, "Near optimal adversarial attack on UCB bandits," *CoRR*, vol. abs/2008.09312, 2020. [Online]. Available: https://arxiv.org/abs/2008.09312

[33] F. Liu and N. Shroff, "Data poisoning attacks on stochastic bandits," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 4042–4050. [Online]. Available: https://proceedings.mlr.press/v97/liu19e.html

[34] A. Rangi, L. Tran-Thanh, H. Xu, and M. Franceschetti, "Secure-ucb: Saving stochastic bandits from poisoning attacks via limited data verification," in *AAAI Conference on Artificial Intelligence*, 2021.

[35] T. Jung, S. Martin, D. Ernst, and G. Leduc, "Contextual multi-armed bandits for the prevention of spam in voip networks," 06 2012.

[36] L. Yang, M. Hajiesmaili, M. S. Talebi, J. C. S. Lui, and W. S. Wong, "Adversarial bandits with corruptions: Regret lower bound and no-regret algorithm," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 19 943–19 952. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/e655c7716a4b3ea67f48c6322fc42ed6-Paper.pdf

[37] A. Carpentier and A. Locatelli, "Tight (lower) bounds for the fixed budget best arm identification bandit problem," in *29th Annual Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, vol. 49. Columbia University, New York, New York, USA: PMLR, 23–26 Jun 2016, pp. 590–604. [Online]. Available: https://proceedings.mlr.press/v49/carpentier16.html

[38] Z. Gao, Y. Han, Z. Ren, and Z. Zhou, "Batched multi-armed bandits problem," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/20f07591c6fcb220ffe637cda29bb3f6-Paper.pdf

[39] O. Maron and A. W. Moore, "The racing algorithm: Model selection for lazy learners," *Artificial Intelligence Review*, vol. 11, no. 1–5, pp. 193–225, 1997. [Online]. Available: http://www.autonlab.org/autonweb/14694/version/3/part/5/data/maron-racing.pdf?branch=main&language=en

[40] S. Shahrampour, M. Noshad, and V. Tarokh, "On sequential elimination algorithms for best-arm identification in multi-armed bandits," *IEEE Transactions on Signal Processing*, vol. 65, pp. 4281–4292, 2017.

[41] M. L. Puterman and M. C. Shin, "Action elimination procedures for modified policy iteration algorithms," *Oper. Res.*, vol. 30, no. 2, p. 301–318, apr 1982.

[42] P. J. Huber, "Robust Estimation of a Location Parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73 – 101, 1964. [Online]. Available: https://doi.org/10.1214/aoms/1177703732

[43] J. W. Tukey, "Mathematics and the picturing of data," in *Proceedings of the International Congress of Mathematicians*. Vancouver: Canadian Mathematical Congress, 1975, p. 523–531.

[44] Y. Cheng, I. Diakonikolas, and R. Ge, "High-dimensional robust mean estimation in nearly-linear time," in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '19. USA: Society for Industrial and Applied Mathematics, 2019, p. 2755–2771.

[45] S. B. Hopkins and J. Li, "How hard is robust mean estimation?" *CoRR*, vol. abs/1903.07870, 2019. [Online]. Available: http://arxiv.org/abs/1903.07870

[46] G. Lugosi and S. Mendelson, "Robust multivariate mean estimation: The optimality of trimmed mean," *The Annals of Statistics*, vol. 49, no. 1, pp. 393 – 410, 2021. [Online]. Available: https://doi.org/10.1214/20-AOS1961

[47] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 5650–5659. [Online]. Available: https://proceedings.mlr.press/v80/yin18a.html

[48] K. Jamieson, M. Malloy, R. Nowak, and S. Bubeck, "lil' ucb : An optimal exploration algorithm for multi-armed bandits," in *Proceedings of The 27th Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, M. F. Balcan, V. Feldman, and C. Szepesvári, Eds., vol. 35. Barcelona, Spain: PMLR, 13–15 Jun 2014, pp. 423–439. [Online]. Available: https://proceedings.mlr.press/v35/jamieson14.html

[49] D. H. Drewry, C. I. Wells, D. M. Andrews, R. Angell, H. Al-Ali, A. D. Axtman, S. J. Capuzz, J. M. Elkins, P. Ettmayer, M. Frederiksen, O. Gileadi, N. Gray, A. Hooper, S. Knapp, S. Laufer, U. Lücking, M. Michaelides, S. Müller, E. Muratov, R. A. Denny, K. S. Saikatendu, D. K. Treiber, W. J. Zuercher, and T. M. Willson, "Progress towards a public chemogenomic set for protein kinases and a call for contributions," *PLoS one*, vol. 12, no. (8): e0181585, pp. 1 – 20, 2017.

[50] Y. Abbasi-Yadkori, P. Bartlett, V. Gabillon, A. Malek, and M. Valko, "Best of both worlds: Stochastic & adversarial best-arm identification," in *Proceedings of the 31st Conference On Learning Theory*, ser. Proceedings of Machine Learning Research, S. Bubeck, V. Perchet, and P. Rigollet, Eds., vol. 75. PMLR, 06–09 Jul 2018, pp. 918–949. [Online]. Available: https://proceedings.mlr.press/v75/abbasi-yadkori18a.html

[51] C.-Y. Wei and H. Luo, "More adaptive algorithms for adversarial bandits," in *Proceedings of the 31st Conference On Learning Theory*, ser. Proceedings of Machine Learning Research, S. Bubeck, V. Perchet, and P. Rigollet, Eds., vol. 75. PMLR, 06–09 Jul 2018, pp. 1263–1291. [Online]. Available: https://proceedings.mlr.press/v75/wei18a.html

[52] S. Ito, "Hybrid regret bounds for combinatorial semi-bandits and adversarial linear bandits," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 2654–2667. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/15a50c8ba6a0002a2fa7e5d8c0a40bd9-Paper.pdf

[53] S. Ito, T. Tsuchiya, and J. Honda, "Adversarially robust multi-armed bandit algorithm with variance-dependent regret bounds," in *Proceedings of Thirty Fifth Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, P.-L. Loh and M. Raginsky, Eds., vol. 178. PMLR, 02–05 Jul 2022, pp. 1421–1422. [Online]. Available: https://proceedings.mlr.press/v178/ito22a.html

[54] S. Bubeck, N. Cesa-Bianchi, and G. Lugosi, "Bandits with heavy tail," *IEEE Transactions on Information Theory*, vol. 59, no. 11, pp. 7711–7717, 2013.

[55] D. Ruppert, *Statistics and data analysis for Financial Engineering*. Springer Science & Business Media, 2010.

APPENDIX A
PROOF OF LEMMA 1

Proof. Without loss of generality, let A denote any best arm identification algorithm. By the definition of our adversarial model, A has to eliminate at least one arm after the first stage of sampling. We denote G as the ordering of arms that A observes after Stage 1 and i^* as the true best arm.

Now, we observe that if the adversary can fully corrupt i^* in Stage 1 of sampling, then the observed mean can be arbitrarily far from the true mean. In particular, since the corruptions are not bound in magnitude, the perceived rank of the arm can be arbitrarily modified if the arm is fully corrupted. In this setting, the best strategy for A to minimize the probability of eliminating i^* is to eliminate a single arm uniformly at random. That is, for any non-uniform strategy, we note that the adversary knows A's strategy and can thus appropriately place i^* in G where it has the maximum probability of being eliminated (probability $> 1/K$). Thus, if the adversary can fully corrupt i^*, any algorithm must have at least probability $1/K$ of eliminating arm i^*.

Now, we show that the adversary can fully corrupt i^* with minimal restrictions on K. We first argue that the algorithm's best strategy is to sample all the arms an equal number of times. In particular, A does not have any information about any of the arms. For any A which follows a non-uniform sampling distribution, there exists a problem instance such that with constant probability, arm i^* will be sampled *less* than in the uniform strategy and will thus require fewer samples to fully corrupt. Thus, we analyze the setting where all arms are sampled n times in Stage 1.

The adversary has a budget of ϵnK and requires $\lceil \epsilon_0 n \rceil$ samples to fully corrupt arm i^*. Thus, the adversary can fully corrupt i^* when

$$\begin{aligned} &\lceil \epsilon_0 n \rceil \leq \epsilon n K \\ \impliedby\ &\epsilon_0 n + 1 \leq \epsilon n K \impliedby K \geq \frac{\epsilon_0 + 1}{\epsilon}, \end{aligned}$$

since $n \geq 1$.

Therefore, there is no best arm identification algorithm which can return the optimal arm with probability $\geq 1 - \delta_0$, for all $\delta_0 \in (0, 1)$ and $K \geq \frac{\epsilon_0 + 1}{\epsilon}$. □

Appendix B
Proof of Lemma 2

Proof. With probability $(1 - \delta_0)$, the robust estimator produces a correct robust estimate of all the arms that are not fully corrupted in every stage of the algorithm. For the rest of this argument, we assume that the robust estimator always produces a correct robust estimate throughout the algorithm's execution.

To prove that the algorithm terminates, we need to prove that the size of the set s_i decreases with i and will eventually satisfy the termination condition in step (d) that $\lfloor \frac{\epsilon |s_i|}{\epsilon_0} \rfloor < 1$. We also need to show that we never remove all the arms from contention (s_i is never empty). Here, we define a "good" arm as an arm that was not fully corrupted in Stage 1.

We will first prove that $|s_i|$ is decreasing. From the definition of s_{i+1} in step (e), this is satisfied when

$$\begin{aligned} |s_{i+1}| < |s_i| &\iff \lceil c \lfloor \epsilon_1 |s_i| \rfloor \rceil < |s_i|, \\ &\impliedby c\epsilon_1 |s_i| + 1 < |s_i|. \end{aligned}$$

For $c = 1 + \lambda = \frac{1 - \sqrt{1 - 4\epsilon_1}}{2\epsilon_1}$, we observe that $c\epsilon_1 \leq 0.5$. This implies that the above equation is satisfied for $|s_i| > 2$. We note that if $|s_i| \leq 2$, then line (d) returns an arm and s_{i+1} is never created.

Now, we can prove that the set s_i always contains an arm. The set s_i can be empty if we create an empty set on step (e) without termination or if we remove all the arms from G_i per step (c). In order to satisfy the first condition, we must create a set s_{i+1} in step (e) that is empty. However, step (e) is only reached if the condition in step (d) that $f_i < 1$ is not satisfied. Combining these two statements together implies that

$$\lceil c \lfloor \epsilon_1 |s_i| \rfloor \rceil < 1 \textbf{ and } \lfloor \epsilon_1 |s_i| \rfloor \geq 1$$

must both be true. This is a contradiction since $c = 1 + \lambda$ for $\lambda > 0$.

We can now show that we will never remove all the arms from G_i, leading to an empty set s_{i+1} in step (e). Arms are only removed in two steps: step (c) and step (e). In particular, we consider the last good arm j in contention. First, we assume that arm j is removed at some stage, Stage i.

We note that it cannot be removed in step (e), since arms are only removed here if there are more than $\lceil c \lfloor \epsilon_1 |s_i| \rfloor \rceil$ arms that will remain in contention. Since every newly corrupted arm is removed after the first time it is corrupted (first distribution shift), this implies that there are more than $\lceil (c-1) \lfloor \epsilon_1 |s_i| \rfloor \rceil \geq 1$ good arms (which have never been corrupted) in contention, which contradicts our definition that j is the last good arm in contention.

Thus, arm j must be removed in step (c). We note that we prune an arm every time there is a distribution shift. In particular, this implies that either an arm was previously corrupted and is no longer corrupted, or vice-versa. This implies that every arm that is pruned in step (c) has to be fully corrupted at least once through the execution of the algorithm. Thus, we can compute the total number of arms that are fully corrupted through all iterations of our loop (lines (a)-(f) in Algorithm 3) and argue that this is less than the number of good arms in contention after Stage 1. For convenience, we denote for $i \geq 1$, $f^*_{i+1} = \epsilon_1 |s^*_{i+1}|$, $s^*_{i+1} = c f^*_i$, and $f^*_1 = \epsilon_1 K$. That is, f^*_i and s^*_i are the non-integer versions of f_i and s_i. We show that arm j cannot be removed using f^*_i and s^*_i and later argue that choosing $f_i = \lfloor \epsilon_1 |s_i| \rfloor$ and $s_{i+1} = \lceil c f_i \rceil$ does not affect correctness.

To do this, we first compute an upper bound on the total number of arms the adversary can fully corrupt through the algorithm's execution. We note that when s^*_2 is created, there are at least $(c-1)\epsilon_1 |s^*_1|$ "good" arms in contention by construction. We will show that the total number of arms that the adversary can fully corrupt from Stage i until termination for any $i \geq 2$ is less than $(c-1)\epsilon_1 |s^*_1|$. In particular, we note that the number of arms that the adversary can fully corrupt in stage i is

$$f^*_i = \epsilon_1 |s^*_i| \leq \epsilon_1 c f^*_{i-1},$$

since $s^*_i \leq c f^*_i$. Thus, the total number of arms that are fully corrupted from Stage i until the algorithm terminates is

$$\sum_i^n f^*_i < \frac{f^*_i}{(1 - c\epsilon_1)},$$

where n is the total number of rounds until termination. This inequality uses the fact that we can upper bound the sum of f^*_i by an infinite geometric series with the same initial term and common ratio.

By construction, there are $(c-1)f_1^*$ good arms available at the start of Stage 2 and any arm is only removed in step (c) if it is fully corrupted at least once. Thus, the adversary can never fully corrupt all "good" arms in contention as long as

$$\sum_{i=2}^{n} f_i^* < (c-1)f_1^* \iff \frac{c\epsilon_1 f_1^*}{1-c\epsilon_1} \leq (c-1)f_1^*.$$

Solving for c, with the constraint that $0 \leq \epsilon_1 \leq 0.25$, this is satisfied when

$$c \in \frac{1 \pm \sqrt{1-4\epsilon_1}}{2\epsilon_1}.$$

To minimize the rank of the returned arm, we can thus choose $c = \frac{1-\sqrt{1-4\epsilon_1}}{2\epsilon_1}$. Thus, the adversary can never fully corrupt all the arms for the chosen value of c.

Finally, we observe that this analysis does not account for the rounding required in f_i and s_i. In particular, since $s_i = \lceil cf_{i-1} \rceil$, s_2 can be larger than s_2^* by 1. This implies that f_2 may be larger than f_2^* by 1 as well (we note that the floor of f_i is irrelevant since it only decreases the adversary's budget). However, we observe that for any i,

$$f_i - f_i^* \leq \epsilon_1(|s_i| - |s_i^*|).$$

This implies that the increase in the true number of corruptions $(f_i - f_i^*)$ in Stage i is less than the increase in the number of arms in contention $(|s_i| - |s_i^*|)$ in Stage i. Thus, our analysis still holds with the required integer relaxations; if any relaxation increases the number of corruptions the adversary can induce, it must correspondingly increase the number of arms in contention at least as much. This indicates that the adversary can never corrupt all the arms in contention.

Thus, we conclude that the last good arm cannot be removed in step (c), which then guarantees that as long as the robust estimator succeeds throughout the algorithm, we will terminate and return an arm. □

APPENDIX C
PROOF OF COROLLARY 1

Proof. With probability $(1-\delta_0)$, the robust estimator produces a correct robust estimate of all the arms that are not fully corrupted in every stage of the algorithm. For the rest of this argument, we assume that the robust estimator always produces a correct robust estimate.

From Lemma 2, we know that Algorithm 3 will terminate and return an arm. For each arm a, we can define r_a as the stage where a was fully corrupted for the last time. By the time the algorithm terminates in step (d), we know that there is no arm in G_i which is still fully corrupted since the adversary no longer has the budget to fully corrupt an arm. Thus, we know that in the last round, any arm that was previously fully corrupted in some round r_a will show a distribution shift between the mean estimate in r_a and the final mean estimate. Thus, this arm will be removed from the set in step (c).

Hence, at termination, Algorithm 3 will return an arm which has never been fully corrupted with probability $(1-\delta_0)$. □

APPENDIX D
PROOF OF PROPOSITION 1

Proof. We denote the pairwise minimum gap between arms as Δ_0. Theorem 3 proves that if we use a minimum gap of $\Delta = \Delta_0$, Algorithm 3 will succeed with high probability. We now consider the performance of Algorithm 3 with $\Delta = \Delta_*$.

We denote $f_1 = \lfloor \epsilon_1 K \rfloor$, which is an upper bound on the number of arms the adversary can fully corrupt in Stage 1. We will show that all the uncorrupted arms in s_2, created from the top $\lceil (1+\lambda)f_1 \rceil = \lceil cf_1 \rceil$ arms in G_1, will have rank $\leq \lceil (2+\lambda)f_1 \rceil = \lceil (c+1)f_1 \rceil$ if we use a minimum gap $\Delta = \Delta_*$.

To show this, we can analyze the ordering G_1 constructed in Stage 1 for any fixed adversarial strategy. We denote G_1^0 as the ordering constructed with $\Delta = \Delta_0$ and G_1^* as the ordering constructed with $\Delta = \Delta_*$.

For a fixed adversarial strategy, we first observe that we can assume G_1^0 and G_1^* contain the same number of fully corrupted arms in the top $\lceil cf_1 \rceil$ ranks, since these arms are placed arbitrarily and the number of fully corrupted arms is only limited by the adversary's budget. Thus, we only need to consider the number of uncorrupted arms that do not satisfy our optimality guarantee. The correctness of Algorithm 3 relies on the fact that there are no uncorrupted arms that do not satisfy our optimality guarantee in the top $\lceil cf_1 \rceil$ ranks.

In particular, we assume there exists an uncorrupted arm j which is in the top $\lceil cf_1 \rceil$ ranks in G_1^*, where the rank of j is not optimal (i.e., j has rank $\geq \lceil (c+1)f_1 \rceil$). This implies that there exists at least one uncorrupted arm q where q satisfies our optimality guarantee but j is ranked above q in G_1^*. That is, if such a q does not exist, then every uncorrupted arm q which satisfies our optimality guarantees is ranked above j; there are at least $\lceil cf_1 \rceil$ such uncorrupted arms (even if the best $\lfloor f_1 \rfloor$ arms are fully corrupted), which indicates that j cannot be in the top $\lceil cf_1 \rceil$ ranks in G_1^*.

Thus, there exists an arm q which is uncorrupted and ranked below j in G_1^*. However,

$$\begin{aligned}\mu_q - \mu_j &\geq \mu_t - \mu_{t+1} \\ &\geq \Delta_*,\end{aligned}$$

where we use the fact that $\mu_j \leq \mu_{t+1}$ since j does not satisfy our optimality guarantees and $\mu_q \geq \mu_t$ since q must satisfy our optimality guarantees.

However, in turn, this implies that our robust estimator did not rank q and j correctly although they have a gap $\geq \Delta_*$, which violates our robust estimator assumption. Therefore, all the uncorrupted arms in the top $\lceil cf_1 \rceil$ ranks of G_1^* have rank at most $\lceil (c+1)f_1 \rceil$.

Now, we show that Algorithm 3 will still terminate and return an uncorrupted arm. We observe that the size of the sets $|s_i|$ decreases in the same manner, regardless of Δ, which shows that the algorithm must terminate.

Thus, for correctness, we only need to show that the algorithm will not remove all uncorrupted arms. We observe that any originally corrupted arm i in s_2 which did not satisfy our optimality definition will still show a distribution shift $> \Delta_*/2$ when it is no longer fully corrupted, by the definition

of Δ_*. Therefore, in the final stage of the algorithm, all the remaining arms will be uncorrupted. Moreover, we note that the "good" arms pruned with $\Delta = \Delta_*$ are a subset of the arms pruned with $\Delta = \Delta_0$ since $\Delta_* \geq \Delta_0$. Thus, we can use the same arguments as in Lemma 2 to show that we will never prune all the good arms. This is sufficient to show that the correctness and termination guarantees of Theorem 3 hold.

This implies that we can use $\Delta = \Delta_*$ in Algorithm 3, completing our argument. □

APPENDIX E
EXPERIMENTAL DETAILS

A. Experimental Setup

We construct experiments to test the performance of our sequential elimination algorithms using first synthetic, then real-world datasets. All experiments were run on an AWS machine of EC2 Instance Type c7g.4xlarge. This instance provides 16vCPUs and an AWS Graviton3 processor. Experiments were parallelized across CPUs for varying values of ϵ. All experiments were completed in ≈ 48 hours of CPU time.

a) Designing a Simple Robust Estimator: For our experiments, we use the α-trimmed mean as our robust estimator and use $\alpha = \epsilon_0 = 0.49$. As before, Δ represents the minimum pairwise gap between the arms. We note that knowing a lower bound on Δ and an upper bound on ϵ is sufficient for our algorithms, though looser bounds can lead to an unnecessary increase in sample complexity. For efficiency, we assume our algorithms have knowledge of Δ and ϵ throughout.

Here, we show that a simple robust estimator that would satisfy Assumption 2 (from Section II-B) under minimal additional conditions is the α-trimmed mean. This estimator simply removes the top and bottom α fraction of samples and calculates the trimmed mean. Now, we can analyze how this estimator performs in a setting where the algorithm has access to an upper bound on ϵ which is less than ϵ_0 and chooses $\alpha = \epsilon_0$. In particular, we will consider a setting where the adversary *does not* fully corrupt an arm and show how to calculate $g(\Delta, \epsilon, \delta)$ to satisfy our guarantees. We note that we will require knowledge of the variance of the underlying probability density distribution of the rewards to appropriately choose g, which we believe is a reasonable assumption and one that is required in prior works (e.g., Mukherjee et al. [16], Altschuler et al. [21], Bubeck et al. [54]).

Our first lower bound on $g(\Delta, \epsilon, \delta)$ is the number of samples required to distinguish between two arms with no corruptions with probability $\geq 1 - \delta$; we denote this number of samples as m_1 and we note that it is finite since we are in the fully identifiable best arm identification setting. We can calculate m_1 in several ways, assuming we have access to the mean and variance of the distributions (e.g., see the work of Even-Dar et al. [14]). Now, we show that for a fraction of corruptions $\epsilon < \epsilon_0$, we can calculate a finite sample size that allows us to correctly order the arms with high probability.

1) **Corruption Strategy A: Trimmed Corruptions** If the adversary introduces an ϵ-fraction of arbitrarily large or small corruptions (outside the $(\alpha, 1-\alpha)$ quantiles) they will be trimmed with high probability given a sufficiently large number of samples (denoted as m_2) by the quantile central limit theorem.
2) **Corruption Strategy B: Untrimmed Corruptions** If the adversary introduces an ϵ-fraction of corruptions within the $(\alpha, 1-\alpha)$ quantiles, the effective impact is to bring the trimmed mean arbitrarily close to the α or $(1-\alpha)$ quantile. By the quantile version of the central limit theorem, we know that these quantiles converge to their expected values, following a standard normal distribution with a variance inversely related to the density of the function. In particular, the q^{th} sample quantile is normally distributed with mean equal to the population quantile (denoted $F^{-1}(q)$) and variance of $\frac{q(1-q)}{nf(F^{-1}(q))^2}$ where $f(x)$ is the density function for the underlying distribution (Result 4.1 of Ruppert [55]). With a variant of the fully identifiable best arm identification assumption where the α-quantile of arm i is identifiable as larger than the $(1-\alpha)$-quantile of arm j for any arms i and j with $\mu_i > \mu_j$ and sufficiently large sample size (denoted as m_3), we can correctly order the arms [21].

Now, we can choose $g(\Delta, \epsilon, \delta) = \max\{m_1, m_2, m_3\}$ to satisfy our guarantees for the α-trimmed mean to order the arms correctly when the fraction of corrupted samples is $\epsilon < \epsilon_0$. We note there may be better robust estimators with varying assumptions on the underlying distributions. We provide the analysis and experimental results to demonstrate the practicality of our algorithms in realistic settings.

In our experiments, we used the α-trimmed mean with $g(\Delta, \epsilon, \delta) = \frac{\log(2K/\delta)}{\Delta^2}$, based on the work of Even-Dar et al. [14]. Further, to ensure we are in the fully identifiable setting, we choose the variance of all distributions to be much smaller than the minimum gap between arms. Throughout this section, we use $\delta = 10^{-12}$, which implies that $\delta_0 \leq 10^{-7}$; if our algorithm incorrectly eliminates all the arms due to a failure in the estimator, we simply return a uniformly random arm from the set of arms.

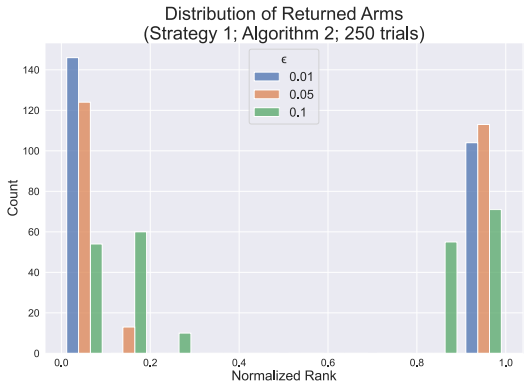
b) Synthetic Dataset Construction: To construct a simple synthetic dataset, we assume that K arms are drawn from a Gaussian distribution. Without loss of generality, we construct the arms as follows:

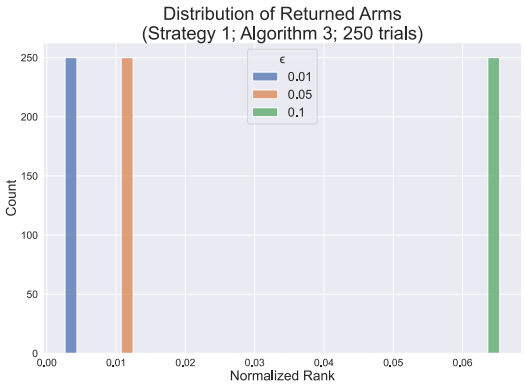
$$\mu_K = 0 \quad \text{and} \quad \mu_{i-1} = \mu_i + \Delta \quad \forall i \in [2, K].$$

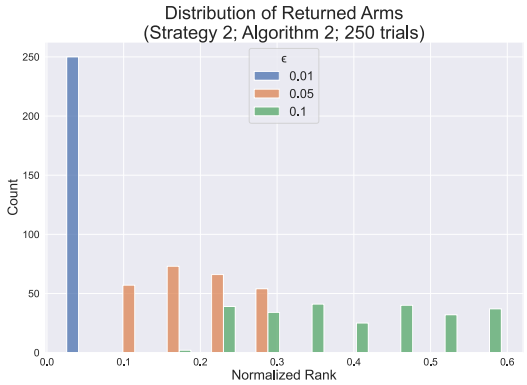
We use $\Delta = 1$ and we vary K from 100 to 1000.

c) PKIS2 Dataset: The PKIS2 dataset, constructed by Drewry et al. [49], is a dataset for drug discovery; it is available online under the Creative Commons Attribution License and can be downloaded in full from this link. This dataset contains a list of kinase, corresponding molecular inhibitors, and a "percentage inhibition", which is a measure of how strongly each inhibitor responds to the kinase. This dataset contains no personally identifying information about how these metrics were calculated.

Following the setup of Mukherjee et al. [16], we choose the kinase ACVRL1. The dataset tests 645 inhibitors against different kinase. There are 189 inhibitors which are tested against ACVRL1 and we initially select all of them. In order to establish a meaningful best inhibitor, we arbitrarily remove inhibitors which have the same percentage inhibition, which







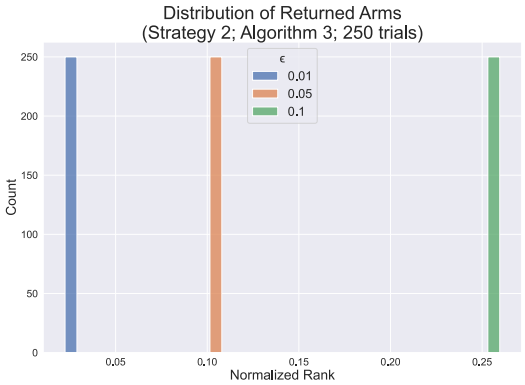


Fig. 6: Under Strategy #1 *(top)*, Algorithm 2 returns a low-rank arm with probability 0.5. We observe ≈ 125 trials out of 250 which have low rank for all ϵ, matching our theoretical estimate. Under Strategy #2 *(bottom)*, we observe a nearly-uniform distribution over the set of arms in contention which all satisfy our theoretical guarantees.

Fig. 7: Algorithm 3 has very low variance and returns a near-optimal arm for all values of ϵ under Strategy #1 *(top)*. Under Strategy #2 *(bottom)*, Algorithm 3 still shows low variance, but the returned arms have higher rank, matching our previous results.

leaves 63 distinct inhibitors. We normalize the percentage inhibition and calculate the percentage control by subtracting the normalized percentage inhibition from 1. Finally, we take the logarithm of the percentage control; this represents the mean of each arm and is observed to follow a Gaussian distribution [16]. The best "arm" in this setting is the one with the largest percentage control against ACVRL1. We model each sample for an inhibitor as a draw from a Gaussian distributions centered at its computed mean.

B. Distribution Analysis

For completeness, we also provide histograms showing the distribution of the returned arm across the 250 trials, for both our algorithms and adversarial strategies. For the synthetic data, we analyze the distributions for $K = 500$. Error bars for performance can be misleading since the baseline distributions are often bimodal. For instance, Algorithm 2 under Strategy #1 has probability $\approx 1/c$ of returning an originally corrupted arm and probability $\approx 1 - 1/c$ of returning a near-optimal arm. This creates two distinct modes where the former has high rank and the latter has low rank. Thus, our results are summarized in Figure 6 and Figure 7.

First, in Figure 6, we observe that Algorithm 2 is bimodal under Strategy #1 as expected. We observe that the distribution of counts matches our theoretical estimates, as ≈ 125 out of 250 trials return an arm of rank ≤ 0.3 for all ϵ. This matches our theoretical estimate since $c = 2$ implies that Algorithm 2 returns an originally uncorrupted arm with probability 0.5. Under Strategy #2, we observe a smooth, nearly uniform distribution over the set of arms in contention, which does not include the lowest-ranked arms for sufficiently large ϵ. This again matches our theoretical results.

In Figure 7, we observe that Algorithm 3 has extremely low variance. This is expected, since the only randomization in the algorithm is from the randomized robust estimator subroutine, which we expect to succeed with high probability. The results from Figure 6 and Figure 7 match our previous results from Figures 2 and 3. Finally, we note that we observe similar results for the PKIS2 dataset.