On the Problem of Best Arm Retention

Houshuang Chen, Yuchen He, and Chihao Zhang*

Shanghai Jiao Tong University, Shanghai 200240, China
{chenhoushuang,yuchen_he,chihao}@sjtu.edu.cn

Abstract. This paper presents a comprehensive study on the problem of Best Arm Retention (BAR), which requires retaining m arms with the best arm included from n after some trials, in stochastic multi-armed bandit settings. We explore many perspectives of the problem.

- We begin by revisiting the lower bound for the (ε, δ)-PAC algorithm for Best Arm Identification (BAI), where we remove the previously imposed restriction of $\delta < 0.5$ in the lower bound found in the literature.
- By refining the technique above, we obtain optimal bounds for (ε, δ)-PAC algorithms for BAR.
- We further study another variant of the problem, called r-BAR, which has recently found applications in streaming algorithms for multi-armed bandits. The goal of the r-BAR problem is to ensure the expected gap between the best arm and the optimal arm retained is less than r. We prove tight sample complexity for the problem.
- We explore the regret minimization problem for r-BAR and develop algorithm beyond pure exploration. We also propose a conjecture regarding the optimal regret in this setting.

Keywords: Best arm identification · PAC learning · Multi-armed bandits.

* Corresponding author.

1 Introduction

The multi-armed bandit (MAB) framework, pioneered by [26], has emerged as a powerful paradigm for modeling sequential decision-making under uncertainty in various real-world applications, ranging from clinical trials to online advertising. Among myriad MAB problems, the *Best Arm Identification* (BAI), as the pure exploration version of stochastic MAB, stands out as a critical task, where the objective is to identify the best arm based on their rewards.

At the beginning of the stochastic MAB game, the player confronts n arms, each associated with an unknown distribution. At each round $t \in [T]$[1], the player chooses an arm and receives a reward. The player is trying to obtain higher accumulated rewards. Equivalently, the goal is to minimize the expected regret, which is the expected accumulated reward difference between playing the best arm with the highest mean and playing with the algorithm chosen arms. In contrast, BAI, the pure exploration version of MAB, seeks to swiftly identify the arm with the highest mean, ignoring reward considerations during the decision-making process.

While achieving a high-probability identification of the best arm remains an unsolved challenge [6,11], an alternative approach involves designing an (ε, δ)-probably approximately correct (PAC) algorithm for BAI. An (ε, δ)-PAC algorithm can find an arm whose mean reward is at most ε from the optimal one with probability at least $1 - \delta$. For this family of algorithms, [24,9] established that the sample complexity is $\Theta\left(\frac{n}{\varepsilon^2} \log \frac{1}{\delta}\right)$.

Recent research has explored streaming algorithms [1,12] employing multiple passes to retain m arms due to memory constraints. These works emphasize retaining the best arm to optimize rewards in subsequent passes. This particular setting naturally leads to the *Best Arm Retention* (BAR) problem, a pragmatic extension of BAI accommodating scenarios with limited memory or computational resources. The name of the problem was coined in [12] and was also known as "arm trapping problem" in literature [1]. However, previous study for the problem is either incomplete or suboptimal.

In BAR, the objective shifts from identifying the arm with the highest expected reward to retain a subset of size m containing the best arm for further exploration or exploitation. In practice, this subset may be subject to constraints like fixed memory capacity, making BAR an adaptable framework for addressing real-world considerations such as uncertainty, dynamic environments, and regret minimization over time. Notably, BAR reduces to the classic BAI problem when $m = 1$, and becomes easier as m increases.

Another similar extension of the BAI problem is to identify and retain the top m arms [16]. However, this extension poses greater complexity and in practice, retaining the best arm alone often suffices. For instance, if one would like to perform some regret minimization algorithm on the m arms, retaining the optimal one already yields optimal regret. Notably, (ε, δ)-PAC algorithm requires $\Omega\left(\frac{n}{\varepsilon^2} \log \frac{m}{\delta}\right)$ samples to retain the top m arms [17], which is worse than our bounds for BAR in Theorem 1.

In this work, we call an arm ε-optimal if the mean gap between the best arm and this arm is less than ε. We address the (ε, δ)-BAR problem, the PAC setting of BAR,

[1] T can be a stopping time.

where the objective is to ensure that the set of m retained arms contains an ε-optimal arm with at least $1-\delta$ probability after observing as few samples as possible. The least number of samples to fulfill the requirement is called the sample complexity of (ε,δ)-BAR.

Theorem 1. *For any (ε,δ)-PAC algorithm for BAR satisfying $\varepsilon \le \frac{1}{8}$ and $\delta \le \frac{n-m}{n}(1-\beta)$, where $\beta \in (0,1)$ is a universal constant, the sample complexity is*

$$\Theta\left(\frac{n-m}{\varepsilon^2}\log\frac{n-m}{n\delta}\right).$$

It is trivial that the sample complexity is zero when $\delta \ge \frac{n-m}{n}$ because we can choose m arms uniformly at random. In fact, the lower bound in Theorem 1 addresses almost all feasible δ except $\frac{n-m}{n}-\delta = o\left(\frac{1}{n}\right)$, as explained in Remark 1.

If the subsequent exploitation only requires obtaining a low regret, as in [12], then it suffices for the expected gap between the mean of the best and the optimal in the retained arm to be small. That is a weaker requirement than (ε,δ)-PAC learnability. To capture the complexity of this requirement, we define the problem r-BAR, where the goal is to guarantee the expected gap is less than r. As before, when $m=1$, it is equivalent to identifying an arm whose mean is at most r from the optimal one in expectation. We call it r-BAI problem, which has been investigated in [3]. We determine the sample complexity of this problem.

Theorem 2. *The sample complexity of r-BAR is $\Theta\left(\frac{(n-m)^3}{(nr)^2}\right)$.*

We further consider the decision-making process beyond pure exploration. Like the classic MAB, we prove both upper bounds and lower bounds for regret minimization. To this end, we introduce a new complexity measure called *regret complexity*, which intuitively measures how much regret one has to pay to retain an arm whose expected mean reward is at most r from the best. The formal definition of the regret complexity is in Section 3.3

Theorem 3. *There exists an algorithm for r-BAR such that the regret complexity is no more than*

$$O\left(\frac{(n-m)^2}{nr}\left(1+\sqrt{\frac{m}{n-m}}\right)\right),$$

and for any algorithm, the regret complexity is no less than

$$\Omega\left(\frac{(n-m)^2}{nr}\right).$$

The gap between the upper and the lower bounds is $\left(1+\sqrt{\frac{m}{n-m}}\right)$. Thus our bounds are tight for $n-m=\Omega(n)$. When m is very close to n, the gap is $\sqrt{n}$, and we will explain in Section 5.3 that eliminating this gap is not easy because different instances requires different sample size and therefore a more sophisticated adaptive strategy is required for an optimal algorithm.

2 Related Work

Pure exploration in stochastic MAB (see [22]) has garnered significant attention and has been explored in various settings, with the most prominent being BAI. Research in this area has investigated sample complexity under fixed confidence [9,14,11,6,32,27], success probability of identifying the best arm with a fixed budget [2,18,34], and (ε, δ)-PAC algorithms for BAI, aiming to identify an ε-optimal arm with fixed confidence [24,13,15]. These settings naturally extend to finding the top m arms [16,17,33,4,30], or other structured arm groups [7,20,29,8,10,23,25].

The concept of *Best Arm Retention* (BAR) was introduced by [12], with a similar idea of “trapping the best arm” first appearing in [1] in the context of stream algorithms. The notion of r-BAI (as a special case of r-BAR when $m = 1$), also referred to as “simple regret”, was discussed by [3]. To the best of our knowledge, this paper is the first systematic investigation of the BAR problem.

3 Notation and Preliminaries

For any integer $n > 0$, let $[n]$ denote the set $\{1, 2, \ldots, n\}$. If $x, y \in \mathbb{R}$ and $x \le y$, $[x, y]$ denotes the closed interval $\{z : x \le z \le y\}$. Let $\texttt{Ber}(x)$ denote the Bernoulli distribution with mean $x \in [0, 1]$. The Kullback-Leibler (KL) divergence between two Bernoulli distributions with means x and y is given by $\mathfrak{d}(x, y) := x \log \frac{x}{y} + (1 - x) \log \frac{1-x}{1-y}$ for brevity. There are some properties of the KL divergence:

Fact 1 *(a) $\mathfrak{d}(\cdot, y)$ (or $\mathfrak{d}(x, \cdot)$) is convex for any fixed y (or x);*
(b) for any $0 \le a \le x \le y \le b \le 1$, $\mathfrak{d}(a, b) \ge \mathfrak{d}(x, y)$;

Proof. (a) Directly calculating $\frac{\partial^2 \mathfrak{d}(x,y)}{\partial x^2} \ge 0$ implies $\mathfrak{d}(\cdot, y)$ is convex and $\mathfrak{d}(x, \cdot)$ is similar.
(b) Since $\left.\frac{\partial \mathfrak{d}(x,y)}{\partial x}\right|_{x=y} = 0$, $\mathfrak{d}(\cdot, y)$ achieves the minimum in $x = y$. Similarly, $\mathfrak{d}(x, y)$ is the minimum of $\mathfrak{d}(x, \cdot)$. Therefore, $\mathfrak{d}(a, b) \ge \mathfrak{d}(x, b) \ge \mathfrak{d}(x, y)$.

3.1 Mutil-Armed Bandits

In this paper, we exclusively consider the stochastic *Multi-Armed Bandit* (MAB) problem, which can be represented by an n-dimensional product distribution $\nu = (\nu_1, \nu_2, \ldots, \nu_n)$. Each distribution corresponds to an arm. At each round/day $t \in [T]$, the player selects an arm $a_t \in [n]$ and receives a reward $r_t \sim \nu_{a_t}$ independently.

Let $\mu = (\mu_1, \mu_2, \ldots, \mu_n)$ denote the mean vector of ν. Define $i^* = \arg\max_{i \in [n]} \mu_i$ as the best arm with the highest mean, and let $\Delta_i = \mu_{i^*} - \mu_i$ represent the mean gap between the best arm and arm i. Additionally, let $T_i = \sum_{t=1}^{T} \mathbb{1}_{a_t = i}$ denote the number of times arm i is pulled. The player's objective is to maximize the accumulated reward $\sum_{t=1}^{T} r_t$, or equivalently, to minimize the regret, defined as $R(n, T) = T\mu_{i^*} - \mathbf{E}\left[\sum_{t=1}^{T} r_t\right] = \mathbf{E}\left[\sum_{i=1}^{n} \Delta_i T_i\right]$, which measures the difference between the accumulated expected reward of the best arm and that of the algorithm. We abbreviate $R(n, T)$ as R when the context is clear and refer to a product distribution ν as an

MAB instance. If each ν_i is a Bernoulli distribution, we also use μ to denote an MAB instance.

For the regret of the MAB problem, there exist several algorithms that achieve tight bounds of $\Theta\left(\sqrt{nT}\right)$ up to a constant factor, such as the *Online Stochastic Mirror Descent* (OSMD) or the *Follow the Regularized Leader* (FTRL) algorithm. The following result from [21] provides a refined constant factor and refer to Appendix C for the detail of the algorithm.

Proposition 1 ([21], Theorem 11). *Running the procedure MirrorDescent on any MAB instance with specific parameters, the regret is at most $R(n,T) \leq \sqrt{2nT}$.*

3.2 Best Arm Identification

Given an MAB instance ν, the *Best Arm Identification* (BAI) problem aims to identify the arm i^* with the highest mean based on as few samples as possible. Different from the fixed T in MAB, the sample size T here is a stopping time with respect to the filtration $(\mathcal{F}_t)_{t\in\mathbb{N}}$ where $\mathcal{F}_t = \sigma(a_1, r_1, a_2, r_2, \ldots, a_t, r_t)$. In our paper, we focus on the (ε,δ)-PAC setting of BAI, denoted as (ε,δ)-BAI, which requires the algorithm to output an ε-optimal arm with probability at least $1-\delta$ for any MAB instance. Here, an arm i is considered ε-optimal if $\mu_{i^*} - \mu_i < \varepsilon$.

The first tight bound for (ε,δ)-BAI was achieved by a median elimination algorithm in [9]. We will provide the details of this algorithm in Appendix D for the completeness.

Proposition 2 ([9], Theorem 10). *Given ε and δ, and an arm set S, the median elimination algorithm $\hat{i} = $ MedianElimination(ε,δ,S), taking ε, δ, and S as input and outputting an ε-optimal arm, is an (ε,δ)-BAI algorithm with sample size $T = O\left(\frac{n}{\varepsilon^2}\log\frac{1}{\delta}\right)$.*

3.3 Best Arm Retention

Given an MAB instance ν, the *Best Arm Retention* (BAR) problem involves retaining m arms S_T out of n after T samples, ensuring that the best arm is included in the retained set $i^* \in S_T$. Similar to BAI, in our paper the sample size T is a stopping time with respect to $(\mathcal{F}_t)_{t\in\mathbb{N}}$. Correspondingly, the (ε,δ)-PAC version of BAR, denoted as (ε,δ)-BAR, requires that the retained m arms contain at least one ε-optimal arm with probability at least $1-\delta$.

In practice, to achieve low regret in a multiple-pass streaming algorithm, it suffices for the *expected gap*[2] between the mean of the best arm and the optimal arm in the retained m arms to be small. We define r-BAR as the problem to guarantee that the expected gap is at most r, namely $\mathbf{E}\left[\mu_{i^*} - \max_{i\in S_T}\mu_i\right] < r$.

The regret for a r-BAR algorithm, $R(n) = T\mu_{i^*} - \mathbf{E}\left[\sum_{t=1}^{T} r_t\right]$, is similar to the regret defined in Section 3.1 except that T is a stopping time. When referring to the sample (or regret) complexity of r-BAR, we denote the minimum samples (or regret) required by any algorithm capable of solving r-BAR for any MAB instance.

[2] We use *mean gap* to refer to the mean difference between i^* and the other fixed arm, and *expected gap* to denote the expected difference in means between i^* and the optimal arm of an arm subset, where the randomness of the expectation arises from the arm subset.

4 An (ε, δ)-PAC Algorithm for BAR

In this section, we will design a simple algorithm with the assistance of the median elimination algorithm to establish an upper bound, followed by a lower bound based on likelihood ratio.

4.1 Upper Bound for (ε, δ)-BAR

Our algorithm presented in Algorithm 1 is straightforward. We first uniformly at random choose $n-m+1$ arms from the set of n arms. Next, we execute the MedianElimination algorithm (Algorithm 6) to obtain an ε-optimal arm (with respect to the chosen arms) with probability $1-\frac{n}{n-m+1}\cdot\delta$. Finally, we output this arm along with the remaining $m-1$ arms that were not chosen in the first stage.

Algorithm 1 (ε, δ)-PAC Algorithm for BAR

```
    Input: $\varepsilon, \delta, m$, arm set $S$
    Output: $m$ arms
Choose $n-m+1$ arms denoted as $S'$ from $n$ arms uniformly at random.
Run the median elimination algorithm $i' = \textsc{MedianElimination}(\varepsilon, \frac{n}{n-m+1}\delta, S')$.
return $S \setminus S' \cup \{i'\}$.
```

Theorem 4 (Part of Theorem 1). *Algorithm 1 is an (ε, δ)-BAR algorithm with a sample size of $O\left(\frac{n-m+1}{\varepsilon^2}\log\frac{n-m+1}{n\delta}\right)$.*

Proof. Our algorithm fails if and only if:

(1) the optimal arm was chosen in the first stage, and
(2) the procedure MedianElimination failed to return a nearly optimal arm.

Therefore, the failure probability is given by $\frac{n-m+1}{n}\cdot\frac{n}{n-m+1}\cdot\delta=\delta$. Furthermore, the sample complexity due to the MedianElimination procedure is $O\left(\frac{n-m+1}{\varepsilon^2}\log\frac{n-m+1}{n\delta}\right)$ by Proposition 2.

4.2 Lower Bounds

Recall the mean vector $\mu = (\mu_1, \mu_2, \cdots, \mu_n)$ denotes a MAB instance, where each arm i follows a Bernoulli distribution with mean μ_i. Consider the following n instances: $\mathcal{H}_1 = (\frac{1}{2}+\varepsilon, \frac{1}{2}, \ldots, \frac{1}{2})$, and for $j \neq 1$, $\mathcal{H}_j$ differs from $\mathcal{H}_1$ only in $\mathcal{H}_j(j) = \frac{1}{2}+2\varepsilon$. We use $\mathbf{Pr}_i[\cdot]$ and $\mathbf{E}_i[\cdot]$ to denote probability and expectation of the algorithm running on instance $\mathcal{H}_i$.

At a high level, if an algorithm outputs arm j as the best arm with a higher probability in $\mathcal{H}_j$ than in $\mathcal{H}_1$, then this algorithm, to some extent, distinguishes the two instances, indicating that arm j should be pulled enough times.

The well-known lower bound for (ε,δ)-BAI, $\Omega\left(\frac{n}{\varepsilon^2}\log\frac{1}{\delta}\right)$, is tight but with the restriction of $\delta < 0.5$ [24]. The lower bound proof techniques from previous literature (e.g. [24]) are mainly based on the following observation: given two instances $\mathscr{H}_1$ and $\mathscr{H}_j$, any proper algorithm should retain arm j with probability at least $1-\delta$ on $\mathscr{H}_j$, but at most δ on $\mathscr{H}_1$. When $1-\delta > \delta$, enough pulls for arm j are required to distinguish between these two instances, which necessitates $\delta < \frac{1}{2}$. However, when $\delta \geq \frac{1}{2}$, a more refined argument is required.

We will begin with a warm-up lower bound for BAR with $m = 1$ (or equivalently, BAI) to demonstrate how to eliminate this restriction in Theorem 5. For $\delta \geq 0.5$, we know that $\Theta\left(\frac{n}{\varepsilon^2}\log\frac{1}{\delta}\right) = \Theta\left(\frac{(1-\delta)n}{\varepsilon^2}\right)$. Therefore our result shows that the lower bound $\Omega\left(\frac{n}{\varepsilon^2}\log\frac{1}{\delta}\right)$ holds for almost all feasible δ. Subsequently, we will extend this method to BAR with general m.

BAI: Warm-up

Theorem 5. *For any (ε,δ)-BAI algorithm satisfying $1-\delta = \frac{1+\Omega(1)}{n}$ and $\varepsilon \leq \frac{1}{8}$, the sample size when running on $\mathscr{H}_1$ is $\mathbf{E}_1[T] = \Omega\left(\frac{(1-\delta)n-1}{\varepsilon^2}\right)$.*

We will prove Theorem 5 in this part. Given any algorithm $\mathcal{A}$, we define B as $\left\{ i \in \{2,\ldots,n\} : \mathbf{Pr}_1[\mathcal{A} \text{ outputs arm } i] \leq \frac{\delta}{n-k} \right\}$, where k is an integer to be determined later such that $\frac{\delta}{n-k} \leq 1-\delta$. It is evident that $|B| \geq k$. Otherwise, there are $n-k$ arms $j \in \{2,\ldots,n-1\}$ satisfying $\mathbf{Pr}_1[\mathcal{A} \text{ outputs arm } j] > \frac{\delta}{n-k}$ and this contradicts that $\mathcal{A}$ is an (ε,δ)-PAC algorithm. However, for each $i \in B$, we have $\mathbf{Pr}_i[\mathcal{A} \text{ outputs arm } i] \geq 1-\delta$. The following lemma obtained by likelihood ratio shows that arm i must be pulled enough times. The proof is given in Appendix A for completeness.

Lemma 1 ([19], Lemma 1). *For any two MAB instances μ, μ' with n arms, and for any algorithm with almost-surely finite stopping time T and event $\mathcal{E} \in \mathcal{F}_T$, $\sum_{i=1}^{n}\left(\mathbf{E}_\mu[T_i]\cdot\mathfrak{d}(\mu_i,\mu'_i)\right) \geq \mathfrak{d}(\mathbf{Pr}_\mu[\mathcal{E}], \mathbf{Pr}_{\mu'}[\mathcal{E}])$.*

Here we only consider the algorithm with almost-surely finite stopping time. Otherwise the sample complexity is infinite and the theorem obviously holds. Therefore, for any $i \in B$, we can apply Lemma 1 to instances $\mathscr{H}_1$ and $\mathscr{H}_i$ with $\mathcal{E}_i = \{\mathcal{A} \text{ outputs arm } i\}$ to obtain $\mathbf{E}_1[T_i]\cdot\mathfrak{d}(0.5, 0.5+2\varepsilon) \geq \mathfrak{d}(\mathbf{Pr}_1[\mathcal{E}_i], \mathbf{Pr}_i[\mathcal{E}_i])$. Since $\mathfrak{d}(0.5, 0.5+2\varepsilon) \leq 12\varepsilon^2$ and $\mathfrak{d}(\mathbf{Pr}_1[\mathcal{E}_i], \mathbf{Pr}_i[\mathcal{E}_i]) \geq \mathfrak{d}\left(\frac{\delta}{n-k}, 1-\delta\right)$ by Fact 1, then $12\varepsilon^2 \cdot \mathbf{E}_1[T_i] \geq \mathfrak{d}\left(\frac{\delta}{n-k}, 1-\delta\right)$. Summing up all $i \in B$, we have $12\varepsilon^2\mathbf{E}_1[T] \geq k \cdot \mathfrak{d}\left(\frac{\delta}{n-k}, 1-\delta\right)$. Here we choose $k = n - \frac{\delta}{\frac{1-\delta}{2}+\frac{1}{2n}} = \Omega(n)$, and thus $\frac{\delta}{n-k} = \frac{1-\delta}{2} + \frac{1}{2n}$. The following lemma, which will be proven in Appendix B, assists us in bounding the KL divergence:

Lemma 2. $\mathfrak{d}(\frac{1-\delta}{2}+\frac{1}{2n}, 1-\delta) = \Omega\left(\frac{1-\delta}{2} - \frac{1}{2n}\right)$ *if* $1-\delta = \frac{1+\Omega(1)}{n}$.

Therefore sample complexity $\mathbf{E}_1[T] = \Omega\left(\frac{(1-\delta)n-1}{\varepsilon^2}\right)$ if $1-\delta = \frac{1+\Omega(1)}{n}$.

Lower Bound for (ε, δ)-BAR In this part, we establish a more general lower bound for the (ε, δ)-PAC algorithms for best arm retention (BAR) by refining arguments in the proof of Theorem 5. Similar to the proof for BAI, we only need to consider the algorithm with the almost-surely finite stopping time for the sample complexity. The following theorem is stronger than the lower bound in Theorem 1.

Theorem 6. *For any (ε, δ)-BAR algorithm with almost-surely finite stopping time such that $\varepsilon \le \frac{1}{8}$ and $\delta \le \frac{n-m}{n}(1-\beta)$, where β is a constant, its sample complexity on the input $\mathcal{H}_1$ satisfies $\mathbf{E}_1\left[T - T_1\right] = \Omega\left(\frac{n-m-\delta}{\varepsilon^2} \log \frac{n-m-\delta}{(n-1)\delta}\right)$.*

We reserve the notations introduced in the previous parts except $\mathcal{E}_i = \{\, \mathcal{A} \text{ retains arm } i \,\}$. For any algorithm, we have $m = \mathbf{E}_1\left[\sum_{i=1}^n \mathbb{1}_{\mathcal{E}_i}\right] = \sum_{i=1}^n \mathbf{Pr}_1\left[\mathcal{E}_i\right]$.

If we directly apply an argument similar to that in the proof of Theorem 5 here, then there exists at least k arms retained with probability at most $\frac{m-(1-\delta)}{n-k}$. Therefore the lower bound is $12\varepsilon^2 \mathbf{E}_1\left[T - T_1\right] \ge k \cdot \mathfrak{d}\left(\frac{m-(1-\delta)}{n-k}, 1-\delta\right)$. We should choose a k satisfying $\frac{m-(1-\delta)}{n-k} < 1-\delta$ to maximize $k \cdot \mathfrak{d}\left(\frac{m-(1-\delta)}{n-k}, 1-\delta\right)$. Consider a special case with $m = n-1$ and $\delta = \frac{1}{2n}$, where we can only choose $k = 1$. Thus $k \cdot \mathfrak{d}(\frac{m-(1-\delta)}{n-k}, 1-\delta) = \mathfrak{d}(1 - \frac{1-\frac{1}{2n}}{n-1}, 1 - \frac{1}{2n}) = \Theta\left(\frac{1}{n}\right)$, which leads to $\mathbf{E}_1\left[T\right] = \Omega\left(\frac{1}{\varepsilon^2 n}\right)$. However, the upper bound is $T \le O\left(\frac{1}{\varepsilon^2}\right)$ in this case.

The above analysis is too pessimistic as we classify suboptimal arms (those in B and those not in B) via a single threshold. The following lemma proved in Appendix B allows us to argue about their sum directly.

Lemma 3. *For any $x_1, x_2 \ldots, x_n \in [0,1]$ with average $a := \frac{\sum_i x_i}{n} < b \in [0,1]$, then $\sum_{i: x_i < b} \mathfrak{d}(x_i, b) \ge n \cdot \mathfrak{d}(a, b)$.*

Armed with Lemma 3, we can sum up all i such that $\mathbf{Pr}_1\left[\mathcal{E}_i\right] \le 1-\delta$ to which Lemma 1 can be applied.

$$\begin{aligned}
12\varepsilon^2 \mathbf{E}_1\left[T - T_1\right] &\ge \sum_{i: \mathbf{Pr}_1\left[\mathcal{E}_i\right] < 1-\delta} \mathfrak{d}\left(\mathbf{Pr}_1\left[\mathcal{E}_i\right], \mathbf{Pr}_i\left[\mathcal{E}_i\right]\right) && \text{(By Lemma 1)} \\
&\ge \sum_{i: \mathbf{Pr}_1\left[\mathcal{E}_i\right] < 1-\delta} \mathfrak{d}\left(\mathbf{Pr}_1\left[\mathcal{E}_i\right], 1-\delta\right) && \text{(By Fact 1)} \\
&\ge (n-1) \cdot \mathfrak{d}\left(\frac{m-(1-\delta)}{n-1}, 1-\delta\right). && \text{(By Lemma 3)}
\end{aligned}$$

Finally, we use a lemma proved in Appendix B to help analyze the KL divergence:

Lemma 4. *For any $0 < a < b < 1$, if $\frac{b-a}{a} = \Omega(1)$, then $\mathfrak{d}(b, a) = \Omega\left(b \cdot \log \frac{b}{a}\right)$.*

Now we are ready to bound $\mathfrak{d}(\frac{m-(1-\delta)}{n-1}, 1-\delta)$. Let $\delta = \frac{n-m}{n}(1-\beta)$ where β is a universal constant.

$$\begin{aligned}\mathfrak{d}\left(\frac{m-(1-\delta)}{n-1}, 1-\delta\right) &= \mathfrak{d}\left(\frac{n-m-\delta}{n-1}, \delta\right) \qquad (\mathfrak{d}(x,y) = \mathfrak{d}(1-x, 1-y))\\ &= \mathfrak{d}\left(\frac{n-m}{n}\left(1+\frac{\beta}{n-1}\right), \frac{n-m}{n}(1-\beta)\right)\\ &\triangleq \mathfrak{d}(B, A).\end{aligned}$$

Here $\frac{B-A}{A} = \frac{\beta/(n-1)+\beta}{1-\beta} = \Omega(1)$, thereby $\mathfrak{d}\left(\frac{m-(1-\delta)}{n-1}, 1-\delta\right) = \Omega\left(\frac{n-m-\delta}{n-1}\log\frac{n-m-\delta}{(n-1)\delta}\right)$. Thus, $\mathbf{E}_1\left[T - T_1\right] = \Omega\left(\frac{n-m-\delta}{\varepsilon^2}\log\frac{n-m-\delta}{(n-1)\delta}\right)$.

Remark 1. This approach encounters limitations when δ approaches the boundary $\frac{n-m}{n}$, specifically when $\frac{n-m}{n} - \delta = o\left(\frac{1}{n}\right)$. For instance, consider the scenario where $m = n-1$ and $\delta = \frac{1}{n} - \frac{1}{n^2}$. In this case, $\mathfrak{d}(\frac{m-(1-\delta)}{n-1}, 1-\delta) = \mathfrak{d}(\frac{n-1}{n} - \frac{1}{n^2(n-1)}, \frac{n-1}{n} + \frac{1}{n^2}) = \Theta\left(\frac{1}{n^2}\right)$. Consequently, the resulting lower bound is $\Omega\left(\frac{1}{\varepsilon^2 n}\right)$.

Suppose there exists an algorithm that achieves this lower bound, making it an $(\varepsilon, \frac{1}{n} - \frac{1}{n^2})$-BAR algorithm with a sample complexity of $c\frac{1}{\varepsilon^2 n}$, where c is a universal constant independent of n. However, as n grows sufficiently large, such that $c\frac{1}{\varepsilon^2 n} \leq 0$, this algorithm is paradoxical. It allows for retaining an ε-optimal arm with a higher probability than $\frac{n-1}{n}$ but without exploration, which is logically impossible.

5 *r*-BAR

Recall that r-BAR requires the mean difference between the best arm from n arms and the optimal arm from the retained pool of size $m < n$ is less than r. In this section, we study both the sample complexity and the minimum regret of this problem. Our results reveal some connections and distinctions between these two optimization objectives.

5.1 Sample Complexity for *r*-BAR

Exploration Algorithm for *r*-BAR Directly adapting the (ε, δ)-PAC algorithm to a r-BAR setting would imply an expected gap bounded by $\delta + (1-\delta)\varepsilon \leq r$. This translates to $\delta \leq r$ and $\varepsilon \leq 2r$ for $\delta \leq 0.5$. Consequently, the sample complexity of this algorithm becomes $O\left(\frac{n-m}{r^2}\log\frac{n-m}{nr}\right)$. However, when r is small, this bound is not tight compared to the optimal bound in Theorem 2. To address this, we leverage the insight that a lower expected gap suggests lower regret. Thus, we employ the procedure MirrorDescent and choose arms with probabilities proportional to their pull counts. A similar approach has been explored in [3,5,12]. Let us restate the algorithm for completeness:

Lemma 5. *Let i^* be the best arm among $\mathcal{S}$, then for n arms with any mean μ as input, Algorithm 2 satisfies* $\mathbf{E}\left[\mu_{i^*} - \mu_{i'}\right] \leq \sqrt{\frac{2n}{T}}$.

Algorithm 2 Find best arm with online stochastic mirror descent

```
Input: arm set $\mathcal{S}$ of size $n$ and time horizon $T$
Output: a good arm
procedure FindBest($S, T$)
    Run MirrorDescent on $\mathcal{S}$ with $T$ rounds
    Compute $T_i, \forall i \in [n]$: the number of times arm $i$ is pulled during $T$ rounds
    Choose arm $i'$ from $\mathcal{S}$ with probability $\frac{T_{i'}}{T}$
    return arm $i'$
```

Proof. Direct calculation yields

$$\begin{aligned}\mathbf{E}\left[\mu_{i^*} - \mu_{i'}\right] &= \mathbf{E}\left[\mathbf{E}\left[\mu_{i^*} - \mu_{i'} | T_1, T_2, \ldots, T_n\right]\right] \\ &= \mathbf{E}\left[\sum_{\text{arm } j \in \mathcal{S}} \Delta_j \cdot \mathbf{Pr}\left[i' = j | T_1, T_2, \ldots, T_n\right]\right] = \mathbf{E}\left[\sum_{\text{arm } j \in \mathcal{S}} \Delta_j \cdot \frac{T_j}{T}\right] \leq \sqrt{\frac{2n}{T}},\end{aligned}$$

where the last inequality follows from Proposition 1.

We can employ the previously described subroutine to devise our final algorithm. Firstly, we randomly select $n - m + 1$ arms from the set of n arms, denoted as S'. We then run Algorithm 2 with a sufficient number of rounds. Finally, we add the output arm to the remaining unchosen arms to form the output set.

Algorithm 3 Optimal sampling for r-BAR

```
Input: arm set $\mathcal{S}$ of size $n \geq m$ and expectation gap $r$
Output: $m$ arms
Sample $n - m + 1$ arms, denoted as $S'$, uniformly at random from $\mathcal{S}$
$i' = $ FindBest($S', T^*$) where $T^* = \frac{2(n-m+2)^3}{(nr)^2}$
return $\{ i' \} \cup S \backslash S'$
```

Theorem 7 (Part of Theorem 2). *Algorithm 3 is an algorithm for r-BAR with sample complexity $O\left(\frac{(n-m)^3}{(nr)^2}\right)$.*

Proof. The sample complexity is straightforward. To demonstrate that the expected gap between the best arm in $\{ i' \} \cup S \backslash S'$, denoted by $\hat{i}$, and the best arm i^* among all n arms is less than r, note that i^* is excluded only if $i^* \in S'$. Thus,

$$\begin{aligned}\mathbf{E}\left[\mu_{i^*} - \mu_{\hat{i}}\right] &= \mathbf{Pr}\left[i^* \notin S'\right] \mathbf{E}\left[\mu_{i^*} - \mu_{\hat{i}} | i^* \notin S'\right] + \mathbf{Pr}\left[i^* \in S'\right] \mathbf{E}\left[\mu_{i^*} - \mu_{\hat{i}} | i^* \in S'\right] \\ &\leq 0 + \frac{n - m + 1}{n} \sqrt{\frac{2(n - m + 1)}{T^*}} < r,\end{aligned}$$

where the inequality follows from Lemma 5.

Lower Bound of Theorem 2 Let $\hat{i}$ be the best arm among the retained m arms. For any r-BAR algorithm, we have $\mathbf{E}\left[\mu_{i^*} - \mu_{\hat{i}}\right] \leq r$. By Markov inequality, we have $\mathbf{Pr}\left[\mu_{i^*} - \mu_{\hat{i}} \geq cr\right] \leq \frac{1}{c}$ for any $c > 0$. This implies that an r-BAR algorithm is also a $(cr, \frac{1}{c})$-PAC algorithm for BAR. Thus then sample complexity is bounded below by $\Omega\left(\frac{n-m}{(cr)^2} \log \frac{c(n-m)}{n}\right)$, as per Theorem 6. Choosing $c = \frac{2n}{n-m}$ completes the proof for the lower bound of Theorem 2

5.2 Regret Minimization for r-BAR

Upper Bound While a pure exploration algorithm suffices for r-BAR, it may yield large regret. For instance, if i^* is not chosen by Algorithm 3 initially, a low-regret algorithm like MirrorDescent running on the suboptimal arm can still result in significant regret. To address this issue, we first run FindBest on all n arms for a few rounds. We then add the output arm to the randomly chosen $n-m+1$ arms. Subsequently, we run FindBest on these $n-m+2$ arms again, with the subsequent process identical to that of Algorithm 3, except for retaining the optimal arm from the initial process. Define $L_2 = \frac{2(n-m+2)^3}{(n-1)^2 r^2}$ and $L_1 = \frac{m-2}{n-1} L_2$. Our algorithm outlined in Algorithm 4 follows a similar approach as proposed in [12], albeit with distinct objectives.

Algorithm 4 Low-regret sampling for BAR

```
Input: arm set $\mathcal{S}$ of size $n \geq m$ and expectation gap $r$
Output: $m$ arms
$i_1 = \textsc{FindBest}(S, L_1)$
Sample $n - m + 1$ arms, denoted as $S'$, uniformly at random from $S \setminus i_1$
$i_2 = \textsc{FindBest}(S' \cup \{ i_1 \}, L_2)$
Uniformly at random choose $n - m$ arms from $S' \setminus \{ i_1, i_2 \}$ to drop
return the remaining arms
```

Theorem 8. *Algorithm 4 is an algorithm for r-BAR with regret $O\left(\sqrt{\frac{(n-m)^3}{nr^2}}\right)$.*

Proof. The proof follows a similar structure to that of Theorem 7. Let $\hat{i}$ denote the best arm among the retained arms. Since i^* will be dropped only if $i^* \in S'$, thus

$$\begin{aligned}
\mathbf{E}\left[\mu_{i^*} - \mu_{\hat{i}}\right] &= \mathbf{Pr}\left[i^* \in S'\right] \mathbf{E}\left[\mu_{i^*} - \mu_{\hat{i}} | i^* \in S'\right] \\
&= \mathbf{Pr}\left[i^* \neq i_1\right] \mathbf{Pr}\left[i^* \in S' | i^* \neq i_1\right] \mathbf{E}\left[\mu_{i^*} - \mu_{\hat{i}} | i^* \in S'\right] \\
&\leq \frac{n-m+1}{n-1} \mathbf{E}\left[\mu_{i^*} - \mu_{\hat{i}} | i^* \in S'\right] \\
&\leq \frac{n-m+1}{n-1} \sqrt{\frac{2(n-m+2)}{L_2}} < r,
\end{aligned}$$

where the last inequality follows from Lemma 5.

Regarding regret, the initial FindBest procedure incurs a regret of $\sqrt{2nL_1}$. In the subsequent step, let i' denote the best arm in $S' \cup i_1$. The regret between playing i' and the algorithm over L_2 rounds amounts to $\sqrt{2(n-m+2)L_2}$. If i' is not i^*, then

$$\begin{aligned}
\mathbf{E}\left[\mu_{i^*} - \mu_{i'}\right] &= \mathbf{Pr}\left[i^* \notin S' \cup \{ i_1 \}\right] \mathbf{E}\left[\mu_{i^*} - \mu_{i'} | i^* \notin S' \cup \{ i_1 \}\right] \\
&= \mathbf{Pr}\left[i_1 \neq i^*\right] \mathbf{Pr}\left[i^* \notin S' | i^* \neq i_1\right] \mathbf{E}\left[\mu_{i^*} - \mu_{i'} | i^* \notin S' \cup \{ i_1 \}\right] \\
&\leq \frac{m-2}{n-1} \mathbf{Pr}\left[i_1 \neq i^*\right] \mathbf{E}\left[\mu_{i^*} - \mu_{i_1} | i^* \notin S' \cup \{ i_1 \}\right] \\
&\overset{\heartsuit}{=} \frac{m-2}{n-1} \mathbf{Pr}\left[i_1 \neq i^*\right] \mathbf{E}\left[\mu_{i^*} - \mu_{i_1} | i_1 \neq i^*\right] \\
&= \frac{m-2}{n-1} \mathbf{E}\left[\mu_{i^*} - \mu_{i_1}\right] \overset{\clubsuit}{\leq} \frac{m-2}{n-1} \sqrt{\frac{2n}{L_1}},
\end{aligned}$$

where $\heartsuit$ follows that $\mu_{i^*} - \mu_{i_1}$ is independent of $\mathbb{1}_{i^* \notin S'}$ conditioned on $i^* \neq i_1$, and $\clubsuit$ is because of Lemma 5. Therefore the regret is

$$\begin{aligned}
\sqrt{2nL_1} + \sqrt{2(n-m+2)L_2)} + \frac{m-2}{n-1}\sqrt{\frac{2n}{L_1}} L_2 &\leq O\left(\frac{\sqrt{m(n-m)^3}}{nr} + \frac{(n-m)^2}{nr}\right) \\
&\leq O\left(\frac{(n-m)^2}{nr}\left(1 + \sqrt{\frac{m}{n-m}}\right)\right).
\end{aligned}$$

When $n - m = \Omega(n)$, our regret bound is $O\left(\frac{(n-m)^2}{nr}\right)$.

Proof of the Lower Bound of Theorem 3 For the scenario where the algorithm does not almost surely stop within finite time, achieving a large regret lower bound requires more effort. In such cases, we cannot deduce a large regret by infinite sample complexity because the algorithm may continually pull the best arm. To tackle this, we first establish a lower bound for algorithms with an almost-surely finite stopping time, and then reduce any algorithm to this case.

For the algorithm with almost-surely finite stopping time, similar to the proof of the lower bound in Section 5.1, an r-BAR algorithm also acts as a $(\frac{2nr}{n-m}, \frac{n-m}{2n})$-PAC algorithm for BAR. Consequently, it must play the suboptimal arms $\mathbf{E}_1\left[T - T_1\right] = \Omega\left(\frac{(n-m)^3}{(nr)^2}\right)$ times on $\mathcal{H}_1$ with $\varepsilon = \frac{2nr}{n-m}$. Therefore, the regret by Wald's equation (see e.g. [28]) is $\Omega\left(\frac{2nr}{n-m}\mathbf{E}_1\left[T - T_1\right]\right) = \Omega\left(\frac{(n-m)^2}{nr}\right)$.

Now assume there exists a $\frac{r}{2}$-BAR algorithm $\mathcal{A}$ with regret $o\left(\frac{(n-m)^2}{nr}\right)$, and let $T' = \omega\left(\frac{(n-m)^2}{nr^2}\right)$ be a fixed number. We can construct an algorithm $\mathcal{A}'$ with finite stopping time as follows: If $\mathcal{A}$ stops with $T < T'$ and outputs S_T, then $\mathcal{A}'$ simulates it. Otherwise, $\mathcal{A}'$ stops in the T'-th round, chooses an arm i' proportional to the pull times of each arm in T' rounds, similar to the procedure FindBest, and outputs it with $m - 1$ randomly chosen arms as $S_{T'}$.

We use $\mathbf{E}_{\mathcal{A}}[\cdot]$ and $\mathbf{E}_{\mathcal{A}'}[\cdot]$ to denote the expectation of the corresponding algorithms running on some MAB instance and let $\hat{i}$ denote the optimal arm among the retained subset arms, similarly for $\mathbf{Pr}_{\mathcal{A}}[\cdot]$ and $\mathbf{Pr}_{\mathcal{A}'}[\cdot]$. We use T_i to denote the number of times that i is pulled and $\mathcal{R} = \sum_{i=1}^{n} \Delta_i T_i$.

It is evident that the regret of $\mathcal{A}'$ is less than that of $\mathcal{A}$ because it may stop earlier. Now we claim that $\mathcal{A}'$ is an r-BAR algorithm with regret $o\left(\frac{(n-m)^2}{nr}\right)$:

$$\begin{aligned}
&\mathbf{E}_{\mathcal{A}'}\left[\mu_{i^*} - \mu_{\hat{i}}\right] \\
&= \mathbf{Pr}_{\mathcal{A}'}\left[T \geq T'\right] \mathbf{E}_{\mathcal{A}'}\left[\mu_{i^*} - \mu_{\hat{i}} | T \geq T'\right] + \mathbf{Pr}_{\mathcal{A}'}\left[T < T'\right] \mathbf{E}_{\mathcal{A}'}\left[\mu_{i^*} - \mu_{\hat{i}} | T < T'\right] \\
&\leq \mathbf{Pr}_{\mathcal{A}}\left[T \geq T'\right] \mathbf{E}_{\mathcal{A}'}\left[\mu_{i^*} - \mu_{i'} | T \geq T'\right] + \mathbf{Pr}_{\mathcal{A}}\left[T < T'\right] \mathbf{E}_{\mathcal{A}}\left[\mu_{i^*} - \mu_{\hat{i}} | T < T'\right] \\
&\leq \mathbf{Pr}_{\mathcal{A}}\left[T \geq T'\right] \mathbf{E}_{\mathcal{A}'}\left[\frac{\mathcal{R}}{T'} | T \geq T'\right] + \mathbf{E}_{\mathcal{A}}\left[\mu_{i^*} - \mu_{\hat{i}}\right] \\
&\leq \frac{1}{T'} \mathbf{Pr}_{\mathcal{A}}\left[T \geq T'\right] \mathbf{E}_{\mathcal{A}}\left[\mathcal{R} | T \geq T'\right] + \frac{r}{2} \leq \frac{1}{T'} \mathbf{E}_{\mathcal{A}}\left[\mathcal{R}\right] + \frac{r}{2} \leq r,
\end{aligned}$$

which leads to a contradiction. Hence, the regret of any r-BAR algorithm is $\Omega\left(\frac{(n-m)^2}{nr}\right)$.

5.3 Difference between Sample Complexity and Regret Minimization

The proof of the lower bound in Section 5.1 reveals that the challenging scenario for r-BAR with optimal regret occurs in $\mathscr{H}_1$ with $\varepsilon = \Theta\left(\frac{nr}{n-m}\right)$. Our analysis in Section 4.2 shows that, on this instance, the requisite number of rounds T is $\Theta\left(\frac{(n-m)^3}{(nr)^2}\right)$ in expectation.

If we consider an MAB game with fixed rounds $T = \Theta\left(\frac{(n-m)^3}{(nr)^2}\right)$, it is well known that the optimal regret is $\Theta\left(\sqrt{nT}\right) = \mathrm{O}\left(\frac{(n-m)^2}{nr}\left(1 + \sqrt{\frac{m}{n-m}}\right)\right)$, which matches our upper bound in Theorem 3. Previous works have shown that this regret lower bound for MAB problem is achieved on the hard instance $\mathscr{H}_1$ but with mean gap parameter $\varepsilon' {=} \Theta\left(\sqrt{\frac{n}{T}}\right) = \Theta\left(\sqrt{\frac{n}{n-m}} \cdot \varepsilon\right)$ (see [22]). This indicates a regret lower bound for Algorithm 4: if an r-BAR algorithm runs for $T = \Theta\left(\frac{(n-m)^3}{(nr)^2}\right)$ rounds on any instances (as our Algorithm 4 does), then it has to suffer a regret of $\Omega\left(\frac{(n-m)^2}{nr}\left(1 + \sqrt{\frac{m}{n-m}}\right)\right)$ on some instances.

This discrepancy between our regret upper and lower bounds indicates a natural idea to improve the algorithm. Note that $\mathscr{H}_1$ with ε' is not the hardest instance for r-BAR. An optimal algorithm need not play $\Theta\left(\frac{(n-m)^3}{(nr)^2}\right)$ rounds on this instance. It should be more adaptive, sampling for different number of times on different instances, rather than treating all instances equally by handling them as the hardest one.

In a nutshell, we conjecture that our lower bound is tight, and a more sophisticated algorithm is required to obtain optimal regret upper bounds.

Conjecture 1. The regret complexity of the r-BAR is $\Theta\left(\frac{(n-m)^2}{nr}\right)$.

References

1. Assadi, S., Wang, C.: Single-pass streaming lower bounds for multi-armed bandits exploration with instance-sensitive sample complexity. In: Proceedings of the 35th Annual Conference on Neural Information Processing Systems (NeurIPS), 2022. 2, 4
2. Audibert, J., Bubeck, S., Munos, R.: Best arm identification in multi-armed bandits. In: Proceedings of the 23th Conference on Learning Theory (COLT), 2010. 4
3. Bubeck, S., Munos, R., Stoltz, G.: Pure exploration in multi-armed bandits problems. In: Proceedings of the 20th International Conference on Algorithmic Learning Theory (ALT), 2009. 3, 4, 9
4. Chen, C.H., He, D., Fu, M., Lee, L.H.: Efficient simulation budget allocation for selecting an optimal subset. INFORMS Journal on Computing **20**(4), 579–595 (2008). https://doi.org/10.1287/IJOC.1080.0268 4
5. Chen, H., He, Y., Zhang, C.: On interpolating experts and multi-armed bandits. arXiv preprint arXiv:2307.07264 (2023). https://doi.org/10.48550/ARXIV.2307.07264 9
6. Chen, L., Li, J., Qiao, M.: Towards instance optimal bounds for best arm identification. In: Proceedings of the 30th Conference on Learning Theory (COLT), 2017. 2, 4
7. Chen, S., Lin, T., King, I., Lyu, M.R., Chen, W.: Combinatorial pure exploration of multi-armed bandits. In: Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NeurIPS), 2014. 4
8. Degenne, R., Ménard, P., Shang, X., Valko, M.: Gamification of pure exploration for linear bandits. In: Proceedings of the 37th International Conference on Machine Learning (ICML), 2020. 4
9. Even-Dar, E., Mannor, S., Mansour, Y.: Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. Journal of Machine Learning Research **7**, 1079–1105 (2006) 2, 4, 5, 19
10. Fiez, T., Jain, L., Jamieson, K.G., Ratliff, L.: Sequential experimental design for transductive linear bandits. In: Proceedings of the 32th Annual Conference on Neural Information Processing Systems (NeurIPS), 2019. 4
11. Garivier, A., Kaufmann, E.: Optimal best arm identification with fixed confidence. In: Proceedings of the 29th Conference on Learning Theory (COLT), 2016. 2, 4
12. He, Y., Ye, Z., Zhang, C.: Understanding memory-regret trade-off for streaming stochastic multi-armed bandits. arXiv preprint arXiv:2405.19752 (2024) 2, 3, 4, 9, 11
13. Howard, S.R., Ramdas, A.: Sequential estimation of quantiles with applications to a/b testing and best-arm identification. Bernoulli **28**(3), 1704–1728 (2022) 4
14. Jamieson, K., Malloy, M., Nowak, R., Bubeck, S.: lil'ucb: An optimal exploration algorithm for multi-armed bandits. In: Proceedings of the 27th Conference on Learning Theory (COLT), 2014. 4
15. Jourdan, M., Degenne, R., Kaufmann, E.: An ε-best-arm identification algorithm for fixed-confidence and beyond. In: Proceedings of the 36th Annual Conference on Neural Information Processing Systems (NeurIPS), 2023. 4
16. Kalyanakrishnan, S., Stone, P.: Efficient selection of multiple bandit arms: Theory and practice. In: Proceedings of the 27th International Conference on Machine Learning (ICML), 2010. 2, 4
17. Kalyanakrishnan, S., Tewari, A., Auer, P., Stone, P.: Pac subset selection in stochastic multi-armed bandits. In: Proceedings of the 29th International Conference on Machine Learning (ICML), 2012. 2, 4
18. Karnin, Z., Koren, T., Somekh, O.: Almost optimal exploration in multi-armed bandits. In: Proceedings of the 30th International Conference on Machine Learning (ICML), 2013. 4

19. Kaufmann, E., Cappé, O., Garivier, A.: On the complexity of best arm identification in multi-armed bandit models. Journal of Machine Learning Research **17**, 1–42 (2016) 7
20. Kone, C., Kaufmann, E., Richert, L.: Bandit pareto set identification: the fixed budget setting. arXiv preprint arXiv:2311.03992 (2023). https://doi.org/10.48550/ARXIV.2311.03992 4
21. Lattimore, T., Gyorgy, A.: Mirror descent and the information ratio. In: Proceedings of the 34th Conference on Learning Theory (COLT), 2021. 5, 18, 19
22. Lattimore, T., Szepesvári, C.: Bandit algorithms. Cambridge University Press (2020) 4, 13
23. Locatelli, A., Gutzeit, M., Carpentier, A.: An optimal algorithm for the thresholding bandit problem. In: Proceedings of the 33th International Conference on Machine Learning (ICML), 2016. 4
24. Mannor, S., Tsitsiklis, J.N.: The sample complexity of exploration in the multi-armed bandit problem. Journal of Machine Learning Research **5**, 623–648 (2004) 2, 4, 7
25. Mason, B., Jain, L., Tripathy, A., Nowak, R.: Finding all ϵ-good arms in stochastic bandits. In: Proceedings of the 33th Annual Conference on Neural Information Processing Systems (NeurIPS), 2020. 4
26. Robbins, H.: Some aspects of the sequential design of experiments. Bulletin of the American Mathematical Society **58**(5), 527–535 (1952) 2
27. Russo, D.: Simple bayesian algorithms for best arm identification. In: Proceedings of the 29th Conference on Learning Theory (COLT), 2016. 4
28. Siegmund, D.: Sequential analysis: tests and confidence intervals. Springer Science & Business Media (2013) 12, 16
29. Simchi-Levi, D., Wang, C., Xu, J.: On experimentation with heterogeneous subgroups: An asymptotic optimal δ-weighted-pac design. Available at SSRN 4721755 (2024) 4
30. Simchowitz, M., Jamieson, K., Recht, B.: The simulator: Understanding adaptive sampling in the moderate-confidence regime. In: Proceedings of the 30th Conference on Learning Theory (COLT), 2017. 4
31. TOPSØE[1], F.: Some bounds for the logarithmic function. Inequality Theory and Applications **4**, 137 (2007) 16
32. Wang, P.A., Tzeng, R.C., Proutiere, A.: Fast pure exploration via frank-wolfe. In: Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS), 2021. 4
33. You, W., Qin, C., Wang, Z., Yang, S.: Information-directed selection for top-two algorithms. In: Proceedings of the 36th Conference on Learning Theory (COLT), 2023. 4
34. Zhao, Y., Stephens, C., Szepesvári, C., Jun, K.S.: Revisiting simple regret: Fast rates for returning a good arm. In: Proceedings of the 40th International Conference on Machine Learning (ICML), 2023. 4

A Proof of Lemma 1

Let $T_i(s)$ denote the index of the s-th pull of arm i for $s \le T_i$. Define the log-likelihood $L_T(a_1, r_1, a_2, r_2, \ldots, a_T, r_T) = \log \frac{\mathbf{Pr}_\mu[a_1,r_1,a_2,r_2,\ldots,a_T,r_T]}{\mathbf{Pr}_{\mu'}[a_1,r_1,a_2,r_2,\ldots,a_T,r_T]}$, abbreviated as L_T when the context is clear. By applying the chain rule to L_T, we have

$$\begin{aligned} L_T &= \log \frac{\prod_{t=1}^T \mathbf{Pr}_\mu\left[a_t|\mathcal{F}_{t-1}\right] \cdot \mathbf{Pr}_\mu\left[r_t|\mathcal{F}_{t-1}, a_t\right]}{\prod_{t=1}^T \mathbf{Pr}_{\mu'}\left[a_t|\mathcal{F}_{t-1}\right] \cdot \mathbf{Pr}_{\mu'}\left[r_t|\mathcal{F}_{t-1}, a_t\right]} \\ &= \sum_{t=1}^T \log \frac{\mathbf{Pr}_\mu\left[r_t|a_t\right]}{\mathbf{Pr}_{\mu'}\left[r_t|a_t\right]} = \sum_{i=1}^n \sum_{s=1}^{T_i} \log \frac{\mathbf{Pr}_\mu\left[r_{T_i(s)}|a_{T_i(s)}\right]}{\mathbf{Pr}_{\mu'}\left[r_{T_i(s)}|a_{T_i(s)}\right]}, \end{aligned}$$

where the second equality follows from $\mathbf{Pr}_{\mu}\left[a_t|\mathcal{F}_{t-1}\right] = \mathbf{Pr}_{\mu'}\left[a_t|\mathcal{F}_{t-1}\right]$ and that r_t is independent of $\mathcal{F}_{t-1}$ conditioned on a_t. With $\mathbf{E}_{\mu}\left[\log \frac{\mathbf{Pr}_{\mu}\left[r_{T_i(s)}|a_{T_i(s)}\right]}{\mathbf{Pr}_{\mu'}\left[r_{T_i(s)}|a_{T_i(s)}\right]}\right] = \mathfrak{d}(\mu_i, \mu_i')$, we apply Wald's Lemma (see e.g. [28]) to $\sum_{i=1}^{n}\sum_{s=1}^{T_i} \log \frac{\mathbf{Pr}_{\mu}\left[r_{T_i(s)}|a_{T_i(s)}\right]}{\mathbf{Pr}_{\mu'}\left[r_{T_i(s)}|a_{T_i(s)}\right]}$ to obtain:

$$\mathbf{E}_{\mu}\left[L_T\right] = \sum_{i=1}^{n} \mathbf{E}_{\mu}\left[T_i\right] \mathfrak{d}\left(\mu_i, \mu_i'\right). \tag{1}$$

The remaining task is to prove $\mathbf{E}_{\mu}\left[L_T\right] \geq \mathfrak{d}(\mathbf{Pr}_{\mu}\left[\mathcal{E}\right], \mathbf{Pr}_{\mu'}\left[\mathcal{E}\right])$ for any event $\mathcal{E} \in \mathcal{F}_T$, we reformulate the definition of L_T as

$$\mathbf{Pr}_{\mu'}\left[a_1, r_1, a_2, r_2, \ldots, a_T, r_T\right] = \exp\{-L\} \cdot \mathbf{Pr}_{\mu}\left[a_1, r_1, a_2, r_2, \ldots, a_T, r_T\right]$$

Summing over all $\omega \in \mathcal{E}$, we obtain

$$\mathbf{Pr}_{\mu'}\left[\mathcal{E}\right] = \mathbf{E}_{\mu}\left[\mathbb{1}_{\mathcal{E}} \cdot \exp\{-L_T\}\right]. \tag{2}$$

Continuing to lower bound Equation (2), we have

$$\begin{aligned} \mathbf{Pr}_{\mu'}\left[\mathcal{E}\right] &= \mathbf{E}_{\mu}\left[\mathbf{E}_{\mu}\left[\mathbb{1}_{\mathcal{E}} \cdot \exp\{-L\}|\mathbb{1}_{\mathcal{E}}\right]\right] \\ &\geq \mathbf{E}_{\mu}\left[\mathbb{1}_{\mathcal{E}} \cdot \exp\{-\mathbf{E}_{\mu}\left[L|\mathbb{1}_{\mathcal{E}}\right]\}\right] \\ &= \mathbf{Pr}_{\mu}\left[\mathcal{E}\right] \mathbf{E}_{\mu}\left[\mathbb{1}_{\mathcal{E}} \cdot \exp\{-\mathbf{E}_{\mu}\left[L|\mathbb{1}_{\mathcal{E}}\right]\}|\mathcal{E}\right] + \mathbf{Pr}_{\mu}\left[\bar{\mathcal{E}}\right] \cdot 0 \\ &= \mathbf{Pr}_{\mu}\left[\mathcal{E}\right] \exp\{-\mathbf{E}_{\mu}\left[L_T|\mathcal{E}\right]\}, \end{aligned}$$

where the inequality follows from the Jensen inequality. Rearranging, we get $\mathbf{E}_{\mu}\left[L_T|\mathcal{E}\right] \geq \log \frac{\mathbf{Pr}_{\mu}[\mathcal{E}]}{\mathbf{Pr}_{\mu'}[\mathcal{E}]}$. Similarly, $\mathbf{E}_{\mu}\left[L_T|\bar{\mathcal{E}}\right] \geq \log \frac{\mathbf{Pr}_{\mu}[\bar{\mathcal{E}}]}{\mathbf{Pr}_{\mu'}[\bar{\mathcal{E}}]}$. Hence, we conclude

$$\begin{aligned} \mathbf{E}_{\mu}\left[L_T\right] &= \mathbf{Pr}_{\mu}\left[\mathcal{E}\right] \mathbf{E}_{\mu}\left[L_T|\mathcal{E}\right] + \mathbf{Pr}_{\mu}\left[\bar{\mathcal{E}}\right] \mathbf{E}_{\mu}\left[L_T|\bar{\mathcal{E}}\right] \\ &\geq \mathbf{Pr}_{\mu}\left[\mathcal{E}\right] \log \frac{\mathbf{Pr}_{\mu}\left[\mathcal{E}\right]}{\mathbf{Pr}_{\mu'}\left[\mathcal{E}\right]} + \mathbf{Pr}_{\mu}\left[\bar{\mathcal{E}}\right] \log \frac{\mathbf{Pr}_{\mu}\left[\bar{\mathcal{E}}\right]}{\mathbf{Pr}_{\mu'}\left[\bar{\mathcal{E}}\right]} \\ &= \mathfrak{d}(\mathbf{Pr}_{\mu}\left[\mathcal{E}\right], \mathbf{Pr}_{\mu'}\left[\mathcal{E}\right]), \end{aligned}$$

which completes our proof in conjunction with Equation (1).

B Bounds of KL Divergence

We will utilize the following inequalities from [31] to bound the KL divergence.

Fact 2 *The following inequalities hold.*

(a) $\log(1+x) \geq \frac{x}{1+x}, \forall x > -1$*;*
(b) $\log(1+x) \geq \frac{x}{1+x}(1+\frac{x}{2+x}) = \frac{2x}{2+x}, \forall x > 0$*;*
(c) $\log(1+x) \geq \frac{x}{1+x}\frac{2+x}{2}$, *if* $-1 < x \leq 0$.

Lemma 6 (Restate Lemma 2). $\mathfrak{d}(\frac{1-\delta}{2}+\frac{1}{2n},1-\delta) = \Omega\left(\frac{1-\delta}{2}-\frac{1}{2n}\right)$ *if* $1-\delta = \frac{1+\Omega(1)}{n}$.

Proof. By definition,

$$
\begin{aligned}
&\mathfrak{d}(\frac{1-\delta}{2}+\frac{1}{2n},1-\delta)\\
&=\left(\frac{1-\delta}{2}+\frac{1}{2n}\right)\log\frac{\frac{1-\delta}{2}+\frac{1}{2n}}{1-\delta}+\left(1-\frac{1-\delta}{2}-\frac{1}{2n}\right)\log\frac{1-\frac{1-\delta}{2}-\frac{1}{2n}}{\delta}\\
&=\log\left(1+\frac{\frac{1-\delta}{2}-\frac{1}{2n}}{\delta}\right)+\left(\frac{1-\delta}{2}+\frac{1}{2n}\right)\left(\log\left(1-\frac{\frac{1-\delta}{2}-\frac{1}{2n}}{1-\delta}\right)+\log\left(1-\frac{\frac{1-\delta}{2}-\frac{1}{2n}}{\frac{1+\delta}{2}-\frac{1}{2n}}\right)\right)\\
&\geq\left(\frac{1-\delta}{2}-\frac{1}{2n}\right)\left(\frac{1}{\frac{1+\delta}{2}-\frac{1}{2n}}-\left(\frac{1-\delta}{2}+\frac{1}{2n}\right)\left(\frac{1}{\frac{1-\delta}{2}+\frac{1}{2n}}\cdot\left(1-\frac{\frac{1-\delta}{2}-\frac{1}{2n}}{2(1-\delta)}\right)+\frac{1}{\delta}\right)\right)\\
&=\left(\frac{1-\delta}{2}-\frac{1}{2n}\right)\left(\frac{1}{\frac{1+\delta}{2}-\frac{1}{2n}}-\frac{3}{4}-\frac{1}{4(1-\delta)n}-\frac{\frac{1-\delta}{2}+\frac{1}{2n}}{\delta}\right)\\
&=\Omega\left(\frac{1-\delta}{2}-\frac{1}{2n}\right),
\end{aligned}
$$

where the inequality follows from (a) & (b) of Fact 2.

Lemma 7 (Restate Lemma 3). *For any* $x_1,x_2\ldots,x_n\in[0,1]$ *with average* $a:=\frac{\sum_i x_i}{n}<b\in[0,1]$, *then* $\sum_{i:x_i<b}\mathfrak{d}(x_i,b)\geq n\cdot\mathfrak{d}(a,b)$.

Proof. Recall that $\mathfrak{d}(\cdot,y)$ is convex for any fixed y in Fact 1. Let $S=\{\,i:x_i<b\,\}$ and $k=|S|$. By the convexity of $\mathfrak{d}(\cdot,b)$, we have $\frac{1}{k}\sum_{i\in S}\mathfrak{d}(x_i,b)\geq\mathfrak{d}\left(\frac{\sum_{i\in S}x_i}{k},b\right)$. Since $\mathfrak{d}(x,b)>\mathfrak{d}(y,b)$ if $x<y<b$ in Fact 1,

$$\sum_{i\in S}\mathfrak{d}(x_i,b)\geq k\cdot\mathfrak{d}\left(\frac{\sum_{i\in S}x_i}{k},b\right)\geq k\cdot\mathfrak{d}\left(\frac{an-(n-k)b}{k},b\right).$$

Using the convexity of $\mathfrak{d}(\cdot,b)$ again, we get

$$\frac{k}{n}\cdot\mathfrak{d}\left(\frac{an-(n-k)b}{k},b\right)+\frac{n-k}{n}\cdot\mathfrak{d}(b,b)\geq\mathfrak{d}(a,b),$$

which implies $k\cdot\mathfrak{d}\left(\frac{an-(n-k)b}{k},b\right)\geq n\cdot\mathfrak{d}(a,b)$ since $\mathfrak{d}(b,b)=0$.

Lemma 8 (Restate Lemma 4). *For any* $0<a<b<1$, *if* $\frac{b-a}{a}=\Omega(1)$, *then* $\mathfrak{d}(b,a)=\Omega\left(b\cdot\log\frac{b}{a}\right)$.

Proof. By definition of the KL divergence, $\mathfrak{d}(b,a)=b\log\frac{b}{a}+(1-b)\log\frac{1-b}{1-a}$.
By Fact 2 (b) & (c),

$$b\log\frac{b}{a}=b\log\left(1+\frac{b-a}{a}\right)\geq(b-a)\left(1+\frac{(b-a)/a}{2+(b-a)/a}\right)$$

and

$$(1-b)\log\frac{1-b}{1-a} = (1-b)\log\left(1+\frac{a-b}{1-a}\right) \geq -(b-a)\left(1-\frac{b-a}{2(1-a)}\right).$$

Therefore if $r := \frac{b-a}{a} = \Omega(1)$,

$$\begin{aligned}\mathfrak{d}(b,a) &= \left(1-\frac{1}{1+r/(2+r)}\right) b\log\frac{b}{a} + \frac{1}{1+r/(2+r)} b\log\frac{b}{a} + (1-b)\log\frac{1-b}{1-a} \\ &\geq \left(1-\frac{1}{1+r/(2+r)}\right) b\log\frac{b}{a} + (b-a) - (b-a)\left(1-\frac{b-a}{2(1-a)}\right) \\ &\geq \left(1-\frac{1}{1+r/(2+r)}\right) b\log\frac{b}{a}.\end{aligned}$$

C Details of the OSMD Algorithm Corresponding to Proposition 1

For completeness, we provide a description of the OSMD algorithm used in Algorithm 2. For more detailed information, please refer to the work of [21].

Let $\Delta_{(n-1)}$ denote the probability simplex with $n-1$ dimensions, defined as $\Delta_{(n-1)} = \{\mathbf{q} \in \mathbb{R}_{\geq 0} : \sum_{i=1}^{n} \mathbf{q}(i) = 1\}$. Here, $\mathbf{q}(i)$ represents the value at the i-th position of vector $\mathbf{q}$. Consider a function $F : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$. The Bregman divergence with respect to F is defined as $B_F(\mathbf{q}, \mathbf{p}) = F(\mathbf{q}) - F(\mathbf{p}) - \langle \nabla F(\mathbf{p}), \mathbf{q} - \mathbf{p}\rangle$ for any $\mathbf{q}, \mathbf{p} \in \mathbb{R}^n$.

The algorithm proposed in [21] is designed for loss cases, where each pull results in a loss associated with the corresponding arm instead of a reward. To adapt their algorithm to our setting, we can perform a simple reduction by constructing the loss of each arm $\ell_t(i)$ as $1-r_t(i)$, where $r_t(i)$ is the reward of arm $\texttt{arm}_i$. It is straightforward to verify that the results in [21] also hold for the reward setting. Let η be the learning rate and $F : \mathbb{R}^{|S|} \to \mathbb{R} \cup \{\infty\}$ be the potential function, where S is the arm set. Without loss of generality, we index the arms in S by $[|S|]$.

Algorithm 5 Online Stochastic Mirror Descent([21])

```
Input: a set of arms $S$ and the number of rounds $L$
procedure MirrorDescent($S, L$)
    $Q_1 \leftarrow \arg\min_{\mathbf{q}\in\Delta_{(|\mathcal{S}|-1)}} F(\mathbf{q})$
    for $t = 1, 2 \ldots, L$ do
        Sample arm $a_t \sim Q_t$, observe reward $r_t(a_t)$ and let $\ell_t(a_t) = 1 - r_t(A_t)$
        Compute reward estimator $\hat{\ell}_t$ as
            $\hat{\ell}_t(i) = \mathbb{1}[A_t = i]\left(\ell_t(i) - \frac{1}{2} + \frac{\eta}{8}\left(1 + \frac{1}{Q_t(i)+\sqrt{Q_t(i)}}\right)\right) - \frac{\eta Q_t(A_t)}{8\left(Q_t(i)+\sqrt{Q_t(i)}\right)}$
        Set $Q_{t+1} = \arg\min_{\mathbf{q}\in\Delta_{(|\mathcal{S}|-1)}} \langle \mathbf{q}, \hat{\ell}_t\rangle + \frac{1}{\eta} \cdot B_F(\mathbf{q}, Q_t)$
```

By choosing $\eta = \sqrt{\frac{8}{L}}$ and $F(\mathbf{q}) = -2\sum_{i=1}^{|S|}\sqrt{\mathbf{q}(i)}$, the conclusion in Proposition 1 can be directly derived from Theorem 11 in [21].

D Details of the MedianElimination Algorithm Corresponding to Proposition 2

For completeness, we present the description of the MedianElimination algorithm we used in Algorithm 1. For more detailed information, please refer to Theorem 10 of [9].

Algorithm 6 Median Elimination([9])

```
Input: a set of arms $S$ of size $n$ and $(\varepsilon, \delta)$
Output: an arm
procedure MedianElimination($\varepsilon, \delta, S$)
    Set $S_1 = S, \varepsilon_1 = \varepsilon/4, \delta_1 = \delta/2, \ell = 1$
    while $|S| > 1$ do
        Sample every arm $a \in S$ for $\frac{4}{\varepsilon_\ell^2} \log \frac{3}{\delta_\ell}$ times, and let $\hat{p}_a^\ell$ denote its empirical mean
        Find the median of $\hat{p}_a^\ell$, denoted by $m_\ell$
        Update: $S_{\ell+1} = S_\ell \backslash \left\{ a : \hat{p}_a^\ell < m_\ell \right\}$
        Update: $\varepsilon_{\ell+1} = \frac{3}{4}\varepsilon_\ell, \delta_{\ell+1} = \frac{\delta_\ell}{2}, \ell = \ell + 1$
```