Optimal Estimation of the Best Mean in Multi-Armed Bandits

Takayuki Osogami
IBM Research – Tokyo
osogami@jp.ibm.com

Junya Honda
Kyoto University, RIKEN AIP
honda@i.kyoto-u.ac.jp

Junpei Komiyama
New York University, RIKEN AIP
junpei@komiyama.info

Abstract

We study the problem of estimating the mean reward of the best arm in a multi-armed bandit (MAB) setting. Specifically, given a target precision ε and confidence level $1 - \delta$, the goal is to return an ε-accurate estimate of the largest mean reward with probability at least $1 - \delta$, while minimizing the number of samples. We first establish an instance-dependent lower bound on the sample complexity, which requires handling the infinitely many possible candidates of the estimated best mean. This lower bound is expressed in a non-convex optimization problem, which becomes the main difficulty of this problem, preventing the direct application of standard techniques such as Track-and-Stop to provably achieve optimality. To overcome this difficulty, we introduce several new algorithmic and analytical techniques and propose an algorithm that achieves the asymptotic lower bound with matching constants in the leading term. Our method combines a confidence ellipsoid-based stopping condition with a two-phase sampling strategy tailored to manage non-convexity proposed algorithm is simple, nearly free of hyperparameters, and achieves the instance-dependent, asymptotically optimal sample complexity. Experimental results support our theoretical guarantees and demonstrate the practical effectiveness of our method.

1 Introduction

AI agents and foundation models are developed to be adapted for, or applied to, a wide range of tasks, users, and deployment contexts. Given this diversity of applications, it is essential to ensure that these systems perform reliably even under challenging or unfavorable conditions. For example, a foundation model should provide a minimal level of utility across various downstream tasks, and an AI agent should consistently satisfy all required safety criteria [14]. In other words, before deployment, we must ensure that the expected performance is acceptable even in worst-case scenarios.

This problem of evaluating expected performance in the worst-case scenario can be naturally formulated within the multi-armed bandit (MAB) framework as the task of estimating the mean of the best arm (i.e., best-mean estimation). This paper adopts the standard MAB convention in which rewards are to be maximized and focuses on designing an algorithm that accurately estimates the best mean.

Our primary focus in algorithm design is minimizing sample complexity, motivated by the substantial cost of evaluating AI agents and foundation models. This concern is particularly relevant given the growing trend toward inference-time reasoning [16, 7, 23], which often demands substantial test-time computation. An AI agent may rely on Best-of-N sampling with millions or billions of candidates [16, 7] or Monte Carlo tree search to select its next action [24], making each evaluation costly.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

More specifically, we develop a probably approximately correct (PAC) algorithm for best-mean estimation, achieving asymptotically optimal sample complexity. The algorithm guarantees an estimation error of at most ε with probability at least $1 - \delta$, and the expected number of samples it uses matches the theoretical lower bound that we establish in this paper. This ensures that, as $\delta \to 0$, the sample complexity achieved by our algorithm is asymptotically optimal.

Our lower bound proof leverages the approach of [2], which addresses the problem of selecting an arm when multiple (a finite number of) correct arms may exist. However, our setting differs in a crucial way: we must select a real-valued estimate of the best mean from *infinitely many possibilities*. This fundamental distinction requires modifications to the technique in [2]. We show that, for Gaussian rewards, any algorithm must use at least $2R^2 f(\boldsymbol{\mu}) \log(1/\delta) - o(\log(1/\delta))$ samples in expectation, where $f(\boldsymbol{\mu})$ is the optimal value of an optimization problem that characterizes the optimal allocation of samples among arms. While $f(\boldsymbol{\mu}) = O(K/\varepsilon^2)$ in the worst case, it is instance-dependent and can be significantly smaller. The key challenge, therefore, is to design an algorithm that adapts to the structure of each instance and achieves the corresponding sample complexity.

Our algorithm builds on the martingale-based anytime confidence bound introduced by [1], a technique that has been widely adopted in the linear bandit literature. We adapt this technique in a novel way to the MAB setting, using it to jointly estimate the expected rewards of arms as a confidence ellipsoid. A key insight is that this ellipsoidal representation enables efficient testing of global hypotheses—such as whether at least one arm has an expected reward above a given threshold μ'—which would be difficult to verify using conventional, per-arm confidence intervals. Moreover, to achieve asymptotically optimal sample complexity, we conduct a detailed analysis showing that the original $(K+1)$-dimensional non-convex characterizing optimization problem can be reduced to a one-dimensional non-convex optimization over a narrow interval. This enables efficient grid search within the narrow interval to optimize the allocation of samples used by our algorithm.

The primary contribution of this paper is a novel PAC algorithm for best-mean estimation, with sample complexity that is asymptotically optimal. A key challenge in our analysis lies in handling the non-convexity of the characterizing optimization problem, which makes it difficult to ensure that the sample allocation closely approximates the optimal one. To validate our theoretical analysis and highlight the practical relevance of our approach, we also present numerical experiments. These results illustrate not only the empirical advantages of the algorithm but also certain limitations that are not captured by the asymptotic theory.

2 Related work

Our problem can be viewed as an instance of pure exploration problems, which requires identifying a quantity that depends on unknown parameters. The most relevant literature in this context is fixed-confidence ε-best arm identification [3, 10, 4, 11, 8], where the goal is to identify one of the ε-best arms with confidence $1 - \delta$. However, our problem and ε-best arm identification are different. Identifying an ε-best arm does not guarantee that the best mean is estimated within ε error, since all estimated means can have greater than ε errors. Conversely, our problem of fixed-confidence ε-best-mean estimation does not guarantee that an ε-best arm is identified. However, our algorithm can be used for 2ε-best arm identification, since it finds U such that all means are below U as well as an arm with mean at least $U - 2\varepsilon$. We empirically compare our algorithm with a representative ε-best arm identification algorithm, UGapEc [4], adapted for best-mean estimation in Section 7.

The utility of the Track-and-Stop algorithm [5] for pure exploration problems is shown in [2]. While the idea of Track-and-Stop could potentially be applied to best-mean estimation, the non-convexity of the characterizing optimization problem prohibits us from establishing the continuity of its optimal solution with respect to the true means, which is utilized by Track-and-Stop. On the other hand, our two-phase algorithm leverages the continuity of the optimal *value* to guarantee its asymptotic optimality. Deriving the optimal sample complexity of gradient-based methods, such as the one in [21], is challenging due to the non-convex nature of the underlying optimization problem, which can result in convergence to suboptimal solutions.

Although non-convexity also appears in the characterizing optimization problems of classical best arm identification, those optimization problems typically allow convex reformulation (e.g., [12, 21]). Also, Russo and Pacchiano deal with the non-convexity by convex relaxation, which is shown to lose optimality at most by a factor of 4 [15]. On the other hand, we solve our nonconvex optimization problem by decomposing it into convex subproblems and reducing it to a one-dimensional optimization problem within a narrow interval specified later, which we can solve via grid search with an arbitrary accuracy.

Best-mean estimation has been studied by [13], where it arises as a subproblem in mechanism design—specifically, in ensuring that desired properties hold across all agent types. [13] proposes a simple two-step approach: first identifying an $(2/3)\varepsilon$-best arm with confidence level $1-\delta/2$, then collecting enough additional samples from it to meet the required guarantee. This method achieves a sample complexity of $O((K/\varepsilon^2)\log(1/\delta))$, and a matching lower bound of $\Omega((K/\varepsilon^2)\log(1/\delta))$ is established. In contrast, our results provide sharp, instance-dependent upper and lower bounds, with matching constants in the leading term, establishing that, for each instance, our algorithm has asymptotically optimal sample complexity.

The problem of best-mean estimation also arises frequently in both machine learning and reinforcement learning. In machine learning, it corresponds to estimating the expected performance of the optimal predictor [9], while in reinforcement learning, it corresponds to estimating the value function [19]—that is, the maximum expected return from a state, achieved by selecting the optimal action at each state. While prior work in these areas focuses on how to best estimate the best mean given a fixed set of samples [20], our work addresses a complementary question: how to efficiently sample to estimate the best mean with high confidence.

3 Settings

We consider a multi-armed bandit (MAB) setting with K arms, where each arm $i \in [K] := \{1, 2, \dots, K\}$ is associated with a reward distribution having mean μ_i. We assume that the reward distributions are R-sub-Gaussian (i.e., the reward X_i from each arm i satisfies $\Pr(|X_i| \geq x) \leq 2\exp(-x^2/R^2)$ for a known constant R), and that the means are bounded in magnitude by a known constant S (i.e., $\max_{i\in[K]} |\mu_i| \leq S$). Let $\mu^\star := \max_{i\in[K]} \mu_i$ denote the mean of the best arm (best mean) and $i^\star := \operatorname{argmax}_i \mu_i$ be the index of a best arm. In the case of multiple optimal arms, we break ties arbitrarily and fix one such arm as $i^\star$. We denote the full mean vector by $\boldsymbol{\mu} := (\mu_1, \mu_2, \dots, \mu_K)$ and define the suboptimality gap for arm i as $\Delta_i := \mu^\star - \mu_i$.

Our objective is to design an algorithm that adaptively samples from the K distributions and stops after a random number of samples, denoted by τ. Upon termination, the algorithm outputs an estimate $\hat{\mu}^\star$ of the best mean such that $|\mu^\star - \hat{\mu}^\star| \leq \varepsilon$ holds with probability at least $1-\delta$, for given parameters (ε, δ). In Section 7.3, we discuss an equivalent formulation of returning an interval of length 2ε that contains $\mu^\star$ with probability at least $1-\delta$. The sample complexity of the algorithm is defined as $\mathbb{E}[\tau]$, where the expectation is taken with respect to the underlying reward distributions. Throughout, we use the following notations: $(x)_+ := \max\{0, x\}$, $x \vee y = \max\{x, y\}$, and $x \wedge y = \min\{x, y\}$.

4 Sample Complexity Lower Bound

We start by deriving the following lower bound on the sample complexity $\mathbb{E}[\tau]$ of our best-mean estimation (BME) problem:

Theorem 1. *Consider any algorithm that returns an ε-accurate estimate of the best mean with probability at least $1-\delta$. Then, when rewards follow Gaussian distribution with variance R^2, the sample complexity is lower bounded as follows:*

$$\liminf_{\delta \downarrow 0} \frac{\mathbb{E}_{\boldsymbol{\mu}}[\tau]}{\log(1/\delta)} \geq 2R^2 f(\boldsymbol{\mu}), \tag{1}$$

where $f(\boldsymbol{\mu})$ is the optimal objective value of the following optimization problem:

$$\text{(P1)} \quad \min_{\mathbf{r}\in[0,\infty)^K, U\in(\mu^\star, \mu^\star+2\varepsilon)} \sum_{i\in[K]} r_i \tag{2}$$

$$\text{s.t.} \quad r_i (U-\mu_i)_+^2 \geq 1 \quad \forall i \in [K] \tag{3}$$

$$\sum_{i\in[K]} r_i (\mu_i - U + 2\varepsilon)_+^2 \geq 1. \tag{4}$$

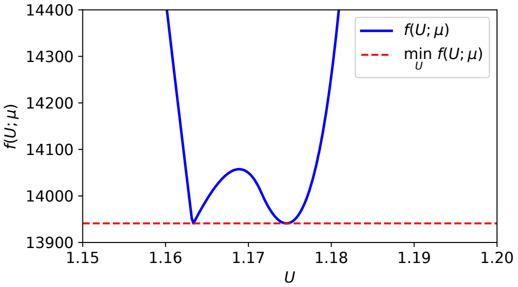


Figure 1: Non-uniqueness of the optimal solution for (P1).

Here, we outline the full proof of the theorem provided in Appendix A.1.

In the proof, we introduce a multiple correct answer problem (MCP) that can be reduced to BME in the sense that an algorithm for BME can also solve this MCB. We then obtain a lower bound for BME from a lower bound for MCP, which is established in [2] (see Lemma 7 in Appendix A.1). We primarily consider a lower bound for Gaussian rewards, similar to [8], but an analogous, albeit less explicit, lower bound may be obtained for more general distributions from that of the M-bin MCP.

Specifically, consider the following M-bin MCP problem with parameter $M \in \mathbb{N}$, which defines $\varepsilon' := \varepsilon/M$. The problem asks to return a bin from a finite set, $\mathcal{B} := \{(-\infty, \mu^\star - \varepsilon), [\mu^\star - \varepsilon, \mu^\star - \varepsilon + \varepsilon'), [\mu^\star - \varepsilon + \varepsilon', \mu^\star - \varepsilon + 2\varepsilon'), \ldots, [\mu^\star + \varepsilon, \mu^\star + \varepsilon + \varepsilon'), [\mu^\star + \varepsilon + \varepsilon', \infty)\}$, where the answer is considered correct if it is in $b^\star(\boldsymbol{\mu}) := \{b \in \mathcal{B} : b \cap [\mu^\star - \varepsilon, \mu^\star + \varepsilon] \neq \emptyset\}$. Notice that any δ-correct algorithm for BME can be converted into a δ-correct algorithm for the M-bin MCP by simply mapping the output of the BME algorithm to the bin that contains the output.

In Lemma 8 (Appendix A.1), we show that, for any $\eta > 0$, there exists a large M such that the sample complexity of the M-bin MCP is bounded from below by $2(1-\eta)R^2 f(\boldsymbol{\mu}) \log(1/\delta) - o(\log(1/\delta))$, where $f(\boldsymbol{\mu})$ is the optimal objective value of (P1). Letting $\eta \to 0$, we establish that any δ-correct algorithm for BME—one that returns an ε-accurate estimate of $\mu^\star$ with probability at least $1-\delta$ for any $\boldsymbol{\mu}$—must incur a sample complexity of at least $2R^2 f(\boldsymbol{\mu}) \log(1/\delta) - o(\log(1/\delta))$.

The optimization problem (P1) plays a central role in our analysis: besides it provides a lower bound on sample complexity, it characterizes the minimal number of samples required from each arm $i \in [K]$ to estimate the best mean to match this bound asymptotically. Specifically, $2R^2 r_i \log(1/\delta)$ represents the number of samples that should be taken from arm $i \in [K]$.

Note that Eq. (2) is a non-convex optimization over $(\mathbf{r}, U)$. An important observation is that, when we fix the value $U \in (\mu^\star, \mu^\star + 2\varepsilon)$, the corresponding subproblem, given by

$$\text{(P2)} \;\; \min_{\mathbf{r}} \sum_{i \in [K]} r_i \quad \text{s.t. the same constraints as (P1) for a given } U \tag{5}$$

is a convex optimization over $\mathbf{r}$. We denote the optimal solution and value of (P2) by $\{r_i(U;\boldsymbol{\mu})\}_{i\in[K]}$ and $f(U;\boldsymbol{\mu})$, respectively. Although $f(U;\boldsymbol{\mu})$ depends on ε, we regard ε as a fixed constant and often omit its dependence throughout the paper for brevity. The optimal solution to (P1) is then given by $U(\boldsymbol{\mu}) = \mathrm{argmin}_{U\in(\mu^\star,\mu^\star+2\varepsilon)} f(U;\boldsymbol{\mu})$ and $r_i(\boldsymbol{\mu}) = r_i(U(\boldsymbol{\mu});\boldsymbol{\mu})$, and we define the corresponding optimal value as $f(\boldsymbol{\mu}) = \min_{U\in(\mu^\star,\mu^\star+2\varepsilon)} f(U;\boldsymbol{\mu})$.

Figure 1 illustrates a numerical example highlighting the nonconvex nature of the optimization problem (P1). Specifically, it shows the objective value $f(U;\boldsymbol{\mu})$ of (P2) as a function of U, revealing the existence of multiple disconnected local minima, which leads to discontinuities in the optimal allocation and complicates the analysis compared to settings where convex reformulations are possible. See Appendix B.8 for further details.

Since (P1) is nonconvex optimization, giving upper and lower bounds of the optimal solution $U(\boldsymbol{\mu})$ is beneficial for implementation and analysis. To this end, we now introduce additional notations. Let

$\underline{U}(\boldsymbol{\mu})$ and $\overline{U}(\boldsymbol{\mu})$ be the solutions to $\underline{g}(U;\boldsymbol{\mu}) = 0$ and $\overline{g}(U;\boldsymbol{\mu}) = 0$, respectively, where we define

$$\underline{g}(U;\boldsymbol{\mu}) = 1 - \sum_{i \in [K]} \frac{(\mu_i - U + 2\varepsilon)_+^2}{(U - \mu_i)^2}$$

$$\overline{g}(U;\boldsymbol{\mu}) = \frac{1}{(\mu^\star - U + 2\varepsilon)^3} - \sum_{i \neq i^\star} \frac{1}{(U - \mu_i)^3} \tag{6}$$

for any $U \in (\mu^\star, \mu^\star + 2\varepsilon)$. As we will show below, $\underline{U}(\boldsymbol{\mu})$ and $\overline{U}(\boldsymbol{\mu})$ can be used to define a narrow interval in which $U(\boldsymbol{\mu})$ is contained. Here, $\underline{g}$ and $\overline{g}$ are monotonically increasing in U, and their roots can be efficiently computed using standard methods such as bisection over the interval $[\mu^\star, \mu^\star + 2\varepsilon]$. For convenience, we extend their domains by setting

$$\begin{aligned} \underline{g}(U;\boldsymbol{\mu}) = \overline{g}(U;\boldsymbol{\mu}) &= -\infty && \text{for } U \leq \mu^\star \\ \underline{g}(U;\boldsymbol{\mu}) = \overline{g}(U;\boldsymbol{\mu}) &= \infty && \text{for } U \geq \mu^\star + 2\varepsilon. \end{aligned} \tag{7}$$

The solution to the optimization problem (P2) can then be characterized as follows:

Lemma 2. *The optimal solution and objective value of (P2) satisfy the following properties:*

(i) $\underline{U}(\boldsymbol{\mu}) \in [\mu^\star + \varepsilon, \mu^\star + 2\varepsilon)$ *and* $\overline{U}(\boldsymbol{\mu}) < \mu^\star + 2\varepsilon$.

(ii) *When* $U \geq \underline{U}(\boldsymbol{\mu})$*, there is an optimal solution*

$$r_i(U;\boldsymbol{\mu}) = \begin{cases} \frac{1}{(U-\mu_i)^2}, & i \neq i^\star \\ \frac{1}{(\mu^\star - U + 2\varepsilon)^2}\left(1 - \sum_{i \neq i^\star} \frac{(\mu_i - U + 2\varepsilon)_+^2}{(U-\mu_i)^2}\right) & i = i^\star, \end{cases} \tag{8}$$

which satisfies $r_{i^\star}(U;\boldsymbol{\mu}) \geq \frac{1}{(U-\mu^\star)^2}$. *In addition, the optimal value satisfies*

$$f(U;\boldsymbol{\mu}) = \sum_{i \neq i^\star} \frac{1}{(U - \mu_i)^2} + \frac{1}{(\mu^\star - U + 2\varepsilon)^2}\left(1 - \sum_{i \neq i^\star} \frac{(\mu_i - U + 2\varepsilon)_+^2}{(U - \mu_i)^2}\right). \tag{9}$$

(iii) *The minimizer of* $f(U;\boldsymbol{\mu})$ *satisfies* $U(\boldsymbol{\mu}) \in [\underline{U}(\boldsymbol{\mu}), \underline{U}(\boldsymbol{\mu}) \vee \overline{U}(\boldsymbol{\mu})]$.

(iv) $r_{i^\star}(U(\boldsymbol{\mu});\boldsymbol{\mu}) \leq \frac{\left((K-1)^{1/3}+1\right)^2}{4\varepsilon^2}$.

(v) $r_{i^\star}(U(\boldsymbol{\mu});\boldsymbol{\mu}) \geq \frac{1}{4\varepsilon^2}$.

The proof of the lemma is postponed to Appendix A.2. Although the function (9) is generally nonconvex, we can still confine its minimizer $U(\boldsymbol{\mu}) = \operatorname{argmin}_U f(U;\boldsymbol{\mu})$ within a reasonably narrow interval using the bounds, $\underline{U}(\boldsymbol{\mu})$ and $\underline{U}(\boldsymbol{\mu}) \vee \overline{U}(\boldsymbol{\mu})$. This allows for efficient approximation of the minimizer via grid search over the interval $[\underline{U}(\boldsymbol{\mu}), \underline{U}(\boldsymbol{\mu}) \vee \overline{U}(\boldsymbol{\mu})] \subset [\mu^\star + \varepsilon, \mu^\star + 2\varepsilon)$. When $\underline{U}(\boldsymbol{\mu}) \geq \overline{U}(\boldsymbol{\mu})$, the minimizer is $\underline{U}(\boldsymbol{\mu})$, and there is no need for solving the non-convex optimization.

5 Ellipsoid-based Estimation Algorithm

Motivated by the structure of the characterizing optimization, we propose the Ellipsoid-based Estimation algorithm (EllipsoidEst) shown in Algorithm 1. In addition to the problem parameters $(K, R, S, \varepsilon, \delta)$ introduced in Section 3, EllipsoidEst uses a regularization parameter λ.

EllipsoidEst is characterized by its novel stopping condition based on a tight confidence *ellipsoid* formed by regularized estimators. It also employs a two-phase sampling strategy to minimize the need for solving non-convex optimization problems. These components are critical to derive an optimal and computationally efficient algorithm for a probably approximately correct (PAC) estimate of the best mean. In what follows, we describe each of these components in detail.

EllipsoidEst computes upper confidence bounds (UCBs; $\hat{\mu}_i(t) + \sqrt{\beta(t,\delta)/N_i(t)}$) in Step 9, where $\beta(t,\delta)$ is defined as

$$\beta(t,\delta) \coloneqq 2R^2 \log\frac{1}{\delta} + \sum_{i \in [K]} \left(R^2 \log\frac{\lambda + N_i(t)}{\lambda} + \lambda S^2 \frac{N_i(t)}{N_i(t) + \lambda}\right), \tag{10}$$

Algorithm 1 EllipsoidEst

```
Require: $K, R, S, \varepsilon, \delta$: problem dependent parameters; $\lambda$: regularization parameter
 1: $N_i(0) \leftarrow 0, \forall i \in [K]$
 2: $\hat{\mu}_i(t) \leftarrow 0, \forall i \in [K]$
 3: Phase $\leftarrow 1$                                        ▷ There are two phases.
 4: for $t = 1, 2, \ldots$ do
 5:     $i(t) \leftarrow \begin{cases} \hat{i}^\star, & \text{Phase} = 2 \text{ and } N_{\hat{i}^\star}(t-1) \le r_{\hat{i}^\star}\beta(t-1,\delta), \\ \operatorname{argmax}_i \left\{ \hat{\mu}_i(t-1) + \sqrt{\frac{\beta(t-1,\delta)}{N_i(t-1)}} \right\} & \text{otherwise.} \end{cases}$
 6:     Update $N_{i(t)}(t) \leftarrow N_{i(t)}(t-1) + 1$
 7:     Pull arm $i(t)$ and obtain reward $x_{i(t),N_{i(t)}(t)}$
 8:     Update $\hat{\mu}_{i(t)}(t) \leftarrow \frac{1}{N_{i(t)}(t)+\lambda} \sum_{m=1}^{N_{i(t)}(t)} x_{i(t),m}$
 9:     $U(t) \leftarrow \max_i \left\{ \hat{\mu}_i(t) + \sqrt{\frac{\beta(t,\delta)}{N_i(t)}} \right\}$
10:     if $\sum_{i\in[K]} N_i(t)(\hat{\mu}_i(t) - U(t) + 2\,\varepsilon)_+^2 \ge \beta(t,\delta)$ then
11:         Output $\hat{\mu}^\star \leftarrow U(t) - \varepsilon$ and terminate.
12:     if Phase $= 1$ and $\underline{g}(U(t); \hat{\boldsymbol{\mu}}(t)) \vee \overline{g}(U(t); \hat{\boldsymbol{\mu}}(t)) \le 0$ then
13:         Phase $\leftarrow 2$                                ▷ Determine the target confidence interval.
14:         $\hat{i}^\star \leftarrow \operatorname{argmax}_i \hat{\mu}_i(t)$
15:         $\hat{U} \leftarrow U(\hat{\boldsymbol{\mu}}(t))$        ▷ Minimize (9) over $U \in [\underline{U}(\hat{\boldsymbol{\mu}}(t)), \overline{U}(\hat{\boldsymbol{\mu}}(t)) \vee \underline{U}(\hat{\boldsymbol{\mu}}(t))]$.
16:         $r_{\hat{i}^\star} \leftarrow r_{\hat{i}^\star}(\hat{U}; \hat{\boldsymbol{\mu}}(t))$
```

and

$$\hat{\mu}_i(t) \coloneqq \frac{1}{N_i(t) + \lambda} \sum_{m=1}^{N_i(t)} x_{i,m} \tag{11}$$

denotes a regularized estimate of the mean reward based on the $N_i(t)$ samples $\{x_{i,1}, \ldots, x_{i,N_i(t)}\}$ collected up to round t.

A natural, yet naive, stopping condition would be to terminate when $U(t) - L_i(t) \le 2\varepsilon$ holds for some arm i, where $U(t)$ is the highest UCB and $L_i(t) \coloneqq \hat{\mu}_k(t) - \sqrt{\beta(t,\delta)/N_i(t)}$ is the lower confidence bound (LCB) for i in round t. This condition guarantees that $U(t) - \varepsilon \in [\mu^\star - \varepsilon, \mu^\star + \varepsilon]$ when the true means are simultaneously contained in the respective confidence bounds.

When multiple arms have estimated means $\hat{\mu}_i(t)$ close to the highest UCB $U(t)$, we can improve upon the naive stopping rule. Namely, EllipsoidEst employs a more refined stopping condition, as specified in Step 10. Figure 2 provides an intuitive illustration of this condition in the case of two arms. It can be shown that, with probability at least $1 - \delta$, the true mean vector lies within the confidence ellipsoid defined by the individual upper and lower confidence bounds. Therefore, if the region $\{(\mu_1, \mu_2) : \mu_1 \vee \mu_2 \le U(t) - 2\varepsilon\}$ lies outside this ellipsoid, we can conclude that the true best-mean must lie within the interval $[U(t) - 2\varepsilon, U(t)]$, even if the confidence interval for each individual arm is not contained in $[U(t) - 2\varepsilon, U(t)]$.

Formally, the stopping condition of EllipsoidEst is supported by the following lemma, which we prove in Appendix A.3 using the martingale-based anytime confidence bound from [1]:

Lemma 3. *Let S be such that $|\mu_i| \le S, \forall i \in [K]$. Then, with probability at least $1 - \delta$, the regularized estimator $\{\hat{\mu}_i(t)\}$ satisfies for all $t \ge 2$ that*

$$\sum_{i\in[K]} N_i(t) \left(\hat{\mu}_i(t) - \mu_i\right)^2 \le \beta(t,\delta). \tag{12}$$

Moreover, when $t \ge 2$ and $\lambda \ge 2/K$, the following simpler bound holds:

$$\beta(t,\delta) \le 2R^2 \log(1/\delta) + KR^2 \log t + K\lambda S^2. \tag{13}$$

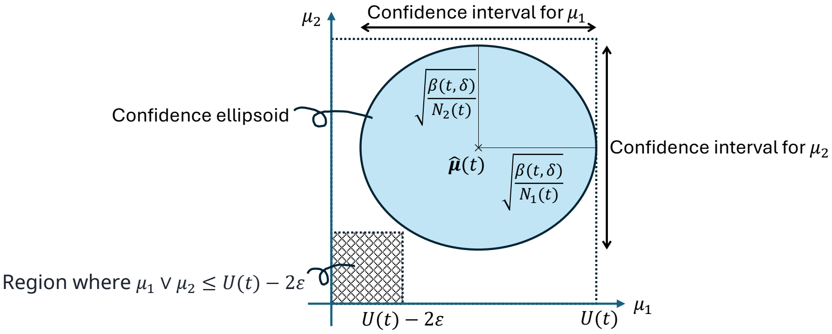

μ_2
Confidence interval for μ_1
Confidence ellipsoid
$\sqrt{\frac{\beta(t,\delta)}{N_2(t)}}$
$\hat{\boldsymbol{\mu}}(t)$
$\sqrt{\frac{\beta(t,\delta)}{N_1(t)}}$
Confidence interval for μ_2
Region where $\mu_1 \vee \mu_2 \le U(t) - 2\varepsilon$
$U(t) - 2\varepsilon$
$U(t)$
μ_1

Figure 2: Confidence ellipsoid for two arms: $\boldsymbol{\mu}$ is in the ellipsoid with probability at least $1 - \delta$.

Remark 1. *Instead, one may construct a confidence bound based on a deviation inequality for exponential families explored in [12]. Since such a bound can be constructed with $\beta(t, \delta) = O(\log\log(N_i(t)))$ instead of $\beta(t, \delta) = O(\log(N_i(t)))$ as in* (10)*, it can lead to a tighter bound for large $N_i(t)$'s. This deviation inequality has an additional advantage that it does not require the knowledge of the constant S (such that $\max_{i\in[K]} |\mu_i| \le S$). However, we confirm in Appendix B.9 that the bound based on this deviation inequality can have improvement only for quite large $N_i(t)$.*

The leading term $2R^2 \log(1/\delta)$ in Eq. (13) is asymptotically optimal as it matches the lower bound in Theorem 1. The following theorem then follows from Lemma 3:

Theorem 4 (Correctness)**.** *The output $\hat{\mu}^\star$ of EllipsoidEst satisfies $|\hat{\mu}^\star - \mu^\star| \le \varepsilon$ with probability at least $1 - \delta$.*

Proof. For the stopping time τ, let $\mathcal{K} := \{i \in [K] : \hat{\mu}_i(\tau) > U(\tau) - 2\varepsilon\}$. Then we have

$$\begin{aligned}
\sum_{i\in\mathcal{K}} N_i(\tau)\left(\hat{\mu}_i(\tau) - (U(\tau) - 2\varepsilon)\right)^2 &= \sum_{i\in[K]} N_i(\tau)\left(\hat{\mu}_i(\tau) - (U(\tau) - 2\varepsilon)\right)_+^2 \\
&\ge \beta(\tau, \delta) \quad \text{(by the stopping condition of Algorithm 1)} \\
&\ge \sum_{i\in[K]} N_i(t)(\hat{\mu}_i(\tau) - \mu_i)^2 \quad \text{(by Lemma 3)} \\
&\ge \sum_{i\in\mathcal{K}} N_i(t)(\hat{\mu}_i(\tau) - \mu_i)^2. \qquad (14)
\end{aligned}$$

This means that $(U(\tau) - 2\varepsilon)_{i\in\mathcal{K}}$ is outside the confidence ellipsoid for the arms in $\mathcal{K}$, which implies that $\mu^\star \in [U(\tau) - 2\varepsilon, U(\tau)]$ with probability at least $1 - \delta$, which in turn implies that $\hat{\mu}^\star = U(\tau) - \varepsilon \in [\mu^\star - \varepsilon, \mu^\star + \varepsilon]$ with probability at least $1 - \delta$. □

While the stopping condition guarantees that the output $\hat{\mu}^\star = U(\tau) - \varepsilon$ is a PAC estimator of the true best-mean—regardless of how the samples are collected—EllipsoidEst is carefully designed to collect samples in a way that achieves asymptotically optimal sample complexity. Our lower bound analysis reveals that all non-best arms should be sampled so that they maintain the same UCB, while the (presumed) best arm sometimes needs to be sampled more frequently to tighten its LCB and accurately estimate the best mean.

Specifically, EllipsoidEst operates in two phases. In Phase 1, it repeatedly selects the arm with the highest UCB, aiming to obtain rough estimates of the arm means without oversampling any arm. Phase 2 is designed to match the lower bound when the estimated means are close to the true values, while still ensuring termination within a reasonable number of rounds even if the estimates at the time of the phase shift are inaccurate. Specifically, the algorithm transitions to Phase 2 when the condition in Step 12 is satisfied. At this point, it tentatively identifies the best arm $\hat{i}^\star$ and estimates its optimal sample allocation $r_{\hat{i}^\star}$. More precisely, it determines that arm $\hat{i}^\star$ should be sampled at least $r_{\hat{i}^\star}\beta(t, \delta)$ times. In Phase 2, the algorithm continues to pull the arm with the highest UCB, except when $\hat{i}^\star$ has not yet received its allocated number of samples (i.e., $N_{\hat{i}^\star}(t) \le r_{\hat{i}^\star}\beta(t, \delta)$ holds); in that case, $\hat{i}^\star$ is pulled instead.

Note that EllipsoidEst returns the midpoint $\hat{\mu}^\star = U(\tau) - \varepsilon$, but what it actually finds is the interval $[U(\tau) - 2\varepsilon, U(\tau)]$ that contains $\mu^\star$ with high probability. We will discuss this in detail in Section 7.3.

6 Sample complexity analysis

Here, we establish the sample complexity of EllipsoidEst that matches the lower bound, and thus proves its asymptotic optimality. We first show that the stopping time τ of EllipsoidEst is almost surely bounded by a certain quantity $T_{\max}$ defined as follows (see Lemma 11 in Appendix A.4):

$$T_{\max} \coloneqq 2\xi R^2 \log(1/\delta) + \gamma = O(KR^2\varepsilon^{-2}\log(1/\delta)), \tag{15}$$

where

$$\xi \coloneqq \frac{4(K-1) + \left((K-1)^{1/3} + 1\right)^2}{4\varepsilon^2} = O(K\varepsilon^{-2}) \tag{16}$$

$$\gamma \coloneqq K\left(2\xi R^2 \log\left(\frac{2\xi R^2}{\lambda K}\log(1/\delta)\right) + \lambda + 2\lambda\xi S^2 + 2\right) = O(\log\log(1/\delta)). \tag{17}$$

This is then used to establish the following sample complexity of EllipsoidEst:

Lemma 5 (Sample complexity)**.** *Consider any sufficiently small* $\delta > 0$ *such that* $\log(1/\delta) \geq K$*. Then the stopping time of EllipsoidEst satisfies* $\tau \leq T_{\max}$ *almost surely for the* $T_{\max}$ *defined in* (15)*. Furthermore, for any* $\varepsilon_c \leq \varepsilon/2$*, the sample complexity satisfies*

$$\mathbb{E}\left[\tau\right] \leq 2R^2 f(\boldsymbol{\mu}) \log(1/\delta) + \frac{\varepsilon_c}{\varepsilon}\zeta\beta(T_{\max}, \delta) + K + T_{\max} \min\left\{1, \delta^{c^2} T_{\max}^{K/2} \exp\left(\frac{K\lambda S^2}{2R^2}\right)\right\}, \tag{18}$$

where

$$\begin{aligned}
\beta(T_{\max}, \delta) &\leq 2R^2 \log(1/\delta) + \alpha(T_{\max}) \\
\alpha(T_{\max}) &\coloneqq KR^2 \log\left(1 + \frac{T_{\max}}{\lambda K}\right) + \lambda S^2 K = O(\log\log(1/\delta)) \\
c &\coloneqq \frac{\varepsilon_c}{\Delta_{\max} + \left(3 + \frac{\alpha(T_{\max})}{R^2 \log(1/\delta)}\right)\varepsilon} \\
\zeta &\coloneqq \frac{2+c}{3} f(\boldsymbol{\mu}) + \frac{28(1+c)^2 K}{\varepsilon} = O(K/\varepsilon^2).
\end{aligned} \tag{19}$$

By setting $\varepsilon_c = \varepsilon(\log(1/\delta))^{-1/3}$, all but the first term in the right-hand side of (18) are $o(\log(1/\delta))$ and we obtain the following theorem (see Appendix A.4):

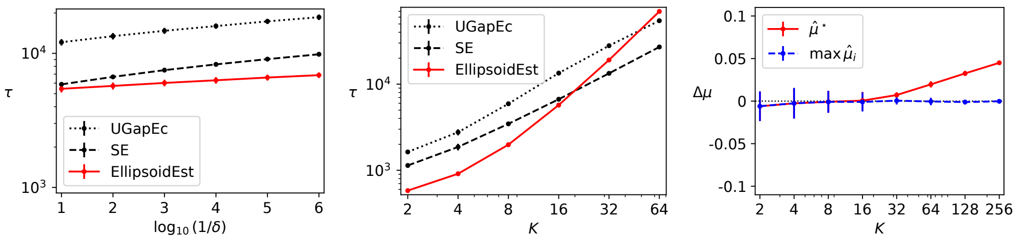
Theorem 6. *EllipsoidEst is asymptotically optimal, that is,*

$$\lim_{\delta \downarrow 0} \frac{\mathbb{E}[\tau]}{\log(1/\delta)} = 2R^2 f(\boldsymbol{\mu}). \tag{20}$$

Lemma 5 and Theorem 6 are proved in Appendix A.4. In the proof, we consider the event

$$\mathcal{A}(c) \coloneqq \left\{\sum_{i \in [K]} N_i(t)(\hat{\mu}_i(t) - \mu_i)^2 \leq c^2 \beta(t, \delta), \forall t \in \{2, 3, \ldots, T_{\max}\}\right\}, \tag{21}$$

whose probability approaches 1 as $\delta \to 0$ (see Lemma 14). Under $\mathcal{A}(c)$, the estimation error in $\hat{\mu}_i(t)$ after the phase shift (i.e., for $t \geq \tau_1$, where τ_1 denotes the first time the condition in Step 12 is met) can be made arbitrarily small by setting a sufficiently small c (see Lemma 12). This concentration result allows us to bound the stopping time τ under $\mathcal{A}(c)$ by a quantity whose leading term is $2R^2 f(\boldsymbol{\mu}) \log(1/\delta)$, by leveraging the structure of the characterizing optimization (see Lemma 13).



(a) $K = 16$

(b) $\delta = 0.01$

(c) $\delta = 0.01$ (bias)

Figure 3: (a)–(b) Sample complexity τ of EllipsoidEst and baselines (SE and UGapEc). (c) Bias in the output (midpoint) and the maximum empirical mean. The results are based on 30 random seeds.

7 Experiments and discussion

7.1 Experimental settings and implementation

We choose to set the regularization parameter as $\lambda = (R/S)^2$. As we discuss in Appendix B.1, this choice minimizes the size of the confidence ellipsoid in the limit of large sample size and is also empirically shown to approximately minimize the sample complexity in a wide range of settings. Owing to this nearly optimal choice, EllipsoidEst is essentially free of hyperparameters.

We design our experiments to evaluate the empirical properties of EllipsoidEst, focusing on three aspects. First, we evaluate how the performance of EllipsoidEst depends on the confidence parameter δ and the number of arms K, comparing it against baselines. Second, we evaluate the EllipsoidEst's sensitivity to the misspecification of the parameters R and S, which reflect assumptions about the reward distributions. Finally, we evaluate the bias in the midpoint of the interval returned by EllipsoidEst and discuss its implications. Due to space constraints, we present full results and details in Appendix B and summarize the main findings in this section.

As baselines, we consider Successive Elimination (SE) [6, 3] and UGapEc [4], both originally designed for best-arm identification and adapted here for best-mean estimation following the approach in [13]. They thus satisfy the same PAC guarantee: the best mean is estimated within ε error with probability at least $1 - \delta$. Figure 9 in Appendix B.4 confirms that these baselines and EllipsoidEst indeed estimate best means with the required accuracy. While SE has suboptimal asymptotic sample complexity, it is known to perform well for moderate values of δ, especially when the number of arms is large. UGapEc achieves an asymptotically optimal order of sample complexity for best-arm identification and also demonstrates strong empirical performance at practical confidence levels.

We assume that each arm's reward follows a Bernoulli distribution and that the means $\boldsymbol{\mu}$ are equally spaced within $[0, 1]$ (specifically, $\mu_i = i/(K+1), \forall i \in [K]$). We fix $\varepsilon = 0.1$ and study the impact of varying δ (the confidence level) and K (the number of arms). In Appendix B, we explore other settings of experiments. See Appendix B.7 for details of the computational environment.

7.2 Results

Figure 3a–3b compares the sample complexity of EllipsoidEst against SE and UGapEc. The results show that EllipsoidEst consistently outperforms the baselines for small and moderate K. Moreover, EllipsoidEst exhibits the slowest growth in sample complexity as $\delta \to 0$, which is consistent with the asymptotic optimality of EllipsoidEst. However, the sample complexity of EllipsoidEst increases more rapidly than the baselines as K grows. This motivates further research on best-mean estimation algorithms that scale more gracefully with K.

The experiments with other settings in Appendix B.4 suggest that UGapEc performs poorly when K is small or when the true means are tightly clustered. SE shows a faster increase in sample complexity (as $\delta \to 0$) across settings. Overall, although each method has its own strengths and limitations, EllipsoidEst demonstrates robust and competitive performance in configurations of practical interest.

In Appendix B.3, we examine the sensitivity of the sample complexity of EllipsoidEst to the values of R and S. As is consistent with our asymptotic analysis (i.e., $\mathbb{E}[\tau] \sim 2R^2 f(\boldsymbol{\mu}) \log(1/\delta)$), the sample complexity exhibits approximately quadratic growth with respect to R. While the influence of S is relatively weak, the sample complexity tends to increase with S, which motivates the sample-shifting technique that we discuss in Appendix B.1.

Figure 3c evaluates the output $\hat{\mu}^\star$ returned by EllipsoidEst (red solid curve) and the maximum of the empirical means (i.e., $\max_i \hat{\mu}_i(\tau)$), based on the samples collected by EllipsoidEst (blue dashed line), relative to the true best-mean $\mu^\star$. The output $\hat{\mu}^\star$ tends to overestimate the best mean, with the bias being particularly pronounced when K is large, while the maximum empirical mean $\max_i \hat{\mu}_i(\tau)$ is notably less biased. In the following, we will discuss this bias in detail.

7.3 Bias in the midpoint of the interval

In Section 3, we defined best-mean estimation as the problem of returning a point estimate $\hat{\mu}^\star$ that lies within ε of the true best-mean $\mu^\star$ with probability at least $1 - \delta$. This may raise questions about the potential bias of $\hat{\mu}^\star$. However, the problem can be equivalently framed as returning an interval of length (at most) 2ε that contains $\mu^\star$ with probability at least $1 - \delta$. Under this equivalent formulation, the point estimate in the original definition is merely the midpoint of such an interval of length 2ε.

The original formulation might be simpler and more familiar within the PAC learning framework, which is why we chose to adopt it initially. However, it has the drawback of placing unnecessary emphasis on the output $\hat{\mu}^\star$, requiring it to lie at the center of an interval solely due to how the problem is framed. It should be understood that *the essential output of EllipsoidEst is the interval* $[U(\tau) - 2\varepsilon, U(\tau)]$*, which contains* $\mu^\star$ *with probability at least* $1 - \delta$. The output $\hat{\mu}^\star = U(\tau) - \varepsilon$ should be viewed as merely the midpoint of this interval, returned to satisfy the requirements of the original problem formulation.

The output is typically higher than the maximum empirical mean (i.e., $\hat{\mu}^\star \geq \max_i \hat{\mu}_i(\tau)$), as is also demonstrated in Figure 3c. Notice that the maximum empirical mean is typically biased upward due to the "winner's curse" [18], i.e., $\mathbb{E}[\max_i \hat{\mu}_i(\tau)] > \max_i \mu_i$, although this phenomenon is not visible in Figure 3c due to the large sample size. In practice, the maximum empirical mean could be provided as supplementary information alongside the primary output—the confidence interval itself. Additional bias correction techniques such as [17, 22] could also be applied to reduce the bias in such supplementary information.

8 Conclusion

We have considered the problem of estimating the mean of the best arm, which can be viewed as a pure exploration problem with an infinite number of answers. Although the characterizing optimization is non-convex, it can be decomposed into a convex subproblem if we fix one parameter. By using this structure, we propose EllipsoidEst that only calculates a non-convex optimization at most once during the runtime. The sample complexity of our algorithm is asymptotically optimal, and its practical performance is verified through numerical experiments. In particular, our algorithm has outperformed the baselines for the number of arms K up to around 16. Improving the sample complexity for large K is interesting future work.

A limitation of EllipsoidEst is that it relies on a known upper bound S on the magnitude of the true means, although this assumption is standard in the linear bandit literature with sub-Gaussian noise since [1]. In some settings (e.g., Bernoulli rewards) S is trivially known, but this is not always the case. When S is unknown, the confidence bound from Theorem 2 in [1] (which we use) becomes inapplicable. However, Theorem 1 in [1] still provides a valid anytime confidence bound that does not require knowledge of S, although the resulting confidence region becomes more complicated.

Acknowledgements

We thank the reviewer who pointed out an error in the original proof of Theorem 1. It has since been corrected using an approach that is simpler than the one proposed during the discussion. JH was supported by JSPS KAKENHI Grant Number JP25K02232.

References

[1] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[2] R. Degenne and W. M. Koolen. Pure exploration with multiple correct answers. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[3] 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*, 7(39):1079–1105, 2006.

[4] V. Gabillon, M. Ghavamzadeh, and A. Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[5] A. Garivier and E. Kaufmann. Optimal best arm identification with fixed confidence. In V. Feldman, A. Rakhlin, and O. Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 998–1027, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR.

[6] A. Hassidim, R. Kupfer, and Y. Singer. An optimal elimination algorithm for learning a best arm. In *Advances in Neural Information Processing Systems*, volume 33, pages 10788–10798. Curran Associates, Inc., 2020.

[7] A. Huang, A. Block, Q. Liu, N. Jiang, A. Krishnamurthy, and D. J. Foster. Is best-of-n the best of them? coverage, scaling, and optimality in inference-time alignment, 2025.

[8] M. Jourdan, R. Degenne, and E. Kaufmann. An ε-best-arm identification algorithm for fixed-confidence and beyond. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[9] H. Kajino, K. Miyaguchi, and T. Osogami. Biases in evaluation of molecular optimization methods and bias reduction strategies. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 15567–15585, 23–29 Jul 2023.

[10] S. Kalyanakrishnan, A. Tewari, P. Auer, and P. Stone. PAC subset selection in stochastic multi-armed bandits. In *International Conference on Machine Learning*, ICML'12, page 227–234. Omnipress, 2012.

[11] Z. Karnin, T. Koren, and O. Somekh. Almost optimal exploration in multi-armed bandits. In *International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 1238–1246. PMLR, 17–19 Jun 2013.

[12] E. Kaufmann and W. M. Koolen. Mixture martingales revisited with applications to sequential tests and confidence intervals. *Journal of Machine Learning Research*, 22(246):1–44, 2021.

[13] T. Osogami, H. Kinoshita, and S. Wasserkrug. Mechanism design with multi-armed bandit, 2024. `https://arxiv.org/abs/2412.00345v1`.

[14] M. Rauh, N. Marchal, A. Manzini, L. A. Hendricks, R. Comanescu, C. Akbulut, T. Stepleton, J. Mateos-Garcia, S. Bergman, J. Kay, C. Griffin, B. Bariach, I. Gabriel, V. Rieser, W. Isaac, and L. Weidinger. Gaps in the safety evaluation of generative AI. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 7(1):1200–1217, Oct. 2024.

[15] A. Russo and A. Pacchiano. Adaptive exploration for multi-reward multi-policy evaluation. In *Forty-second International Conference on Machine Learning*, 2025.

[16] C. Snell, J. Lee, K. Xu, and A. Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters, 2024. `https://arxiv.org/abs/2408.03314`.

[17] L. Sun, A. Dimitromanolakis, L. L. Faye, A. D. Paterson, D. Waggott, T. D. R. Group, and S. B. Bull. BR-squared: A practical solution to the winner's curse in genome-wide scans. *Human Genetics*, 129, 2011.

[18] R. Thaler. Anomalies: The winner's curse. *Journal of Economic Perspectives*, 2(1):191–202, 1988.

[19] H. van Hasselt. Double Q-learning. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

[20] H. van Hasselt. Estimating the maximum expected value: An analysis of (nested) cross validation and the maximum sample average, 2013. `https://arxiv.org/abs/1302.7175`.

[21] P.-A. Wang, R.-C. Tzeng, and A. Proutiere. Fast pure exploration via Frank-Wolfe. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 5810–5821. Curran Associates, Inc., 2021.

[22] R. Xiao and M. Boehnke. Quantifying and correcting for the winner's curse in genetic association studies. *Genetic Epidemiology*, 33(5):453–462, 2009.

[23] F. Xu, Q. Hao, Z. Zong, J. Wang, Y. Zhang, J. Wang, X. Lan, J. Gong, T. Ouyang, F. Meng, C. Shao, Y. Yan, Q. Yang, Y. Song, S. Ren, X. Hu, Y. Li, J. Feng, C. Gao, and Y. Li. Towards large reasoning models: A survey on scaling LLM reasoning capabilities, 2025. `https://arxiv.org/abs/2501.09686`.

[24] X. Yu, B. Peng, V. Vajipey, H. Cheng, M. Galley, J. Gao, and Z. Yu. ExACT: Teaching AI agents to explore with reflective-MCTS and exploratory learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

A Proofs and technical lemmas

A.1 Analysis of the lower bound

Consider the following multiple correct answer problem (MCP) [2]. In addition to K arms with means $\boldsymbol{\mu}$, this problem involves a finite answer set $\mathcal{B}$. There is an answer function $b^\star : \mathbb{R}^K \to 2^{\mathcal{B}}$ that determines the set of correct answers for a given model $\boldsymbol{\mu}$. Consider an adaptive algorithm that samples rewards from K arms and stops. Let τ be the stopping time. When this algorithm stops, it outputs $\hat{b}^\star \in \mathcal{B}$. An algorithm is δ-correct if $\hat{b}^\star \in b^\star(\boldsymbol{\mu})$ with probability at least $1 - \delta$ under any model $\boldsymbol{\mu}$.

Lemma 7. (Lower bound in the multiple correct answer problem, Theorem 1 in [2]) *Any δ-correct MCP algorithm satisfies*

$$\liminf_{\delta \downarrow 0} \frac{\mathbf{E}_{\boldsymbol{\mu}}[\tau]}{\log(1/\delta)} \geq \frac{1}{\max\limits_{b \in b^\star(\boldsymbol{\mu})} \max\limits_{\mathbf{w} \in \Delta^K} \inf\limits_{\boldsymbol{\lambda}: b \notin b^\star(\boldsymbol{\lambda})} \sum\limits_{i \in [K]} w_i d(\mu_i, \lambda_i)}, \tag{22}$$

where $d(\mu_i, \lambda_i)$ denotes the KL divergence from the distribution with mean μ_i to that with mean λ_i; hence, $d(\mu_i, \lambda_i) = (\mu_i - \lambda_i)^2/(2R^2)$ for Gaussian distributions. Here, Δ^K denotes the $(K-1)$-dimensional simplex.

We then introduce an M-bin MCP problem with parameter $M \in \mathbb{N}$. Let $\varepsilon' = \varepsilon/M$. This problem involves bins $\mathcal{B} := \{(-\infty, \mu^\star - \varepsilon), [\mu^\star - \varepsilon, \mu^\star - \varepsilon + \varepsilon'), [\mu^\star - \varepsilon + \varepsilon', \mu^\star - \varepsilon + 2\varepsilon'), \dots, [\mu^\star + \varepsilon, \mu^\star + \varepsilon + \varepsilon'), [\mu^\star + \varepsilon + \varepsilon', \infty)\}$. The number of bins is finite ($2M + 2$). We define the answer function as $b^\star(\boldsymbol{\mu}) := \{b \in \mathcal{B} : b \cap [\mu^\star - \varepsilon, \mu^\star + \varepsilon] \neq \emptyset\}$.

The output $\hat{\mu}^\star$ of the original BME problem can be put into one of the bins. We can see that, if we convert a δ-correct BME algorithm into the M-bin MCP problem by assigning the estimated best mean into a bin including it, the resulting algorithm is a δ-correct M-bin MCP algorithm. By this fact, Lemma 7 applies as a lower bound of the original BME algorithm. In the following, we lower-bound the performance of the M-bin MCP algorithm.

Lemma 8. (Lower bound optimization for M-bin MCP algorithm) *Let $\eta > 0$ be arbitrary. Then there exists $M \in \mathbb{N}$ such that the RHS of Eq.* (22) *for M-bin MCP is larger than $2(1-\eta)R^2 f(\boldsymbol{\mu})$ for Gaussian rewards with variance R^2.*

Proof of Lemma 8. The denominator of RHS of Eq. (22) is

$$\max_{b \in b^\star(\boldsymbol{\mu})} \max_{\boldsymbol{w} \in \Delta^K} \inf_{\boldsymbol{\lambda}: b \notin b^\star(\boldsymbol{\lambda})} \sum_{i \in [K]} w_i d(\mu_i, \lambda_i). \tag{23}$$

We upper-bound Eq. (23) as follows. Take any bin $b \in b^\star(\boldsymbol{\mu}) = \{[\mu^\star - \varepsilon, \mu^\star - \varepsilon + \varepsilon'), \dots, [\mu^\star + \varepsilon, \mu^\star + \varepsilon + \varepsilon')\}$. Let $L(b)$ the leftmost point in b. Then, $L(b) \in [\mu^\star - \varepsilon, \mu^\star + \varepsilon]$. Let $b_A = (-\infty, L(b) - \varepsilon - \varepsilon']$, $b_B = [L(b) + \varepsilon + \varepsilon', \infty)$. Then

$$b \notin b^\star(\boldsymbol{\lambda}) \iff (\lambda^\star + \varepsilon + \varepsilon' \leq L(b)) \vee (L(b) + \varepsilon' \leq \lambda^\star - \varepsilon) \iff \lambda^\star \in b_A \cup b_B, \tag{24}$$

where we denote $\lambda^\star := \max_{i \in [K]} \lambda_i$. Hence, we have

$$\begin{aligned} \max_{\boldsymbol{w} \in \Delta^K} \inf_{\boldsymbol{\lambda}: b \notin b^\star(\boldsymbol{\lambda})} \sum_{i \in [K]} w_i d(\mu_i, \lambda_i) &= \max_{\boldsymbol{w} \in \Delta^K} \inf_{\boldsymbol{\lambda}: \lambda^\star \in b_A \cup b_B} \sum_{i \in [K]} w_i d(\mu_i, \lambda_i) & (25) \\ &= \frac{1}{2R^2} \max_{\boldsymbol{w} \in \Delta^K} \inf_{\boldsymbol{\lambda}: \lambda^\star \in b_A \cup b_B} \sum_{i \in [K]} w_i (\mu_i - \lambda_i)^2 & (26) \end{aligned}$$

for Gaussian rewards. Here,

$$\inf_{\boldsymbol{\lambda}: \lambda^\star \in b_A} \sum_{i \in [K]} w_i (\mu_i - \lambda_i)^2 = \sum_{i \in [K]} w_i (\mu_i - (L(b) - \varepsilon - \varepsilon'))_+^2 =: A(b, \boldsymbol{w}), \tag{27}$$

since $L(b) \leq \mu^\star + \varepsilon$ and hence $L(b) - \varepsilon - \varepsilon' < \mu^\star$ (the minimizer is $\lambda_i = L(b) - \varepsilon - \varepsilon'$ if $\mu_i \geq L(b) - \varepsilon - \varepsilon'$ and $\lambda_i = \mu_i$ otherwise). Also,

$$\inf_{\boldsymbol{\lambda}: \lambda^\star \in b_B} \sum_{i \in [K]} w_i (\mu_i - \lambda_i)^2 = \min_i w_i (L(b) + \varepsilon + \varepsilon' - \mu_i)_+^2 =: B(b, \boldsymbol{w}) \tag{28}$$

since $L(b) \geq \mu^\star - \varepsilon$ and hence $L(b) + \varepsilon + \varepsilon' > \mu^\star$ (the minimizer is $\lambda_i = L(b) + \varepsilon + \varepsilon'$ for the minimizer i of $w_i(\mu_i - (L(b) - \varepsilon - \varepsilon'))_+^2$ and $\lambda_i = \mu_i$ for other i). By using these, we obtain

$$\inf_{\boldsymbol{\lambda}:\lambda^\star \in b_A \cup b_B} \sum_{i\in[K]} w_i(\mu_i - \lambda_i)^2 = \min(A(b, \boldsymbol{w}), B(b, \boldsymbol{w})) =: C(b, \boldsymbol{w}),$$

and therefore

$$\frac{1}{\max_{b\in b^\star(\boldsymbol{\mu})} \max_{\boldsymbol{w}\in\Delta^K} \inf_{\boldsymbol{\lambda}:b\notin b^\star(\boldsymbol{\lambda})} \sum_{i\in[K]} w_i d(\mu_i, \lambda_i)} = \frac{2R^2}{C^\star} \tag{29}$$

for $C^\star = \max_{b\in b^\star(\boldsymbol{\mu})} \max_{\boldsymbol{w}\in\Delta^K} C(b, \boldsymbol{w})$. We write $(b^\star, \boldsymbol{w}^\star)$ to denote its maximizer, which always exists because this is a maximization of a continuous function $C(b, \boldsymbol{w})$ over a compact set. Here note that

$$\begin{aligned}
B(b^\star, \boldsymbol{w}^\star) \geq C(b^\star, \boldsymbol{w}^\star) \geq C([\mu^\star, \mu^\star + \varepsilon'), \mathbf{1}/K)& \\
&= \frac{1}{K} \min\left\{ \sum_{i\in[K]} (\varepsilon + \varepsilon' - \Delta_i)_+^2, \min_i(\Delta_i + \varepsilon + \varepsilon')_+^2 \right\} \\
&= \frac{(\varepsilon + \varepsilon')^2}{K} \geq \frac{\varepsilon^2}{K},
\end{aligned} \tag{30}$$

where $\mathbf{1}$ is the all-one vector, and $\Delta_i := \mu^\star - \mu_i$ for $i \in [K]$. Let $\alpha := \left(1 - \frac{2\sqrt{K}\varepsilon'}{\varepsilon}\right)^{-1} > 1$, $U' := L(b^\star) + \varepsilon - \varepsilon'$, and $r'_i := \alpha w_i^\star / C^\star$ for $i \in [K]$. Then we have

$$\sum_{i\in[K]} r'_i(\mu_i - U' + 2\varepsilon)_+^2 = \alpha A(b^\star, \boldsymbol{w}^\star)/C^\star \geq \alpha > 1,$$

$$\begin{aligned}
r'_i(U' - \mu_i)_+^2 &= \frac{w_i^\star(L(b^\star) + \varepsilon + \varepsilon' - \mu_i)_+^2}{C^\star} \cdot \alpha \left(\frac{\sqrt{w_i^\star}(L(b^\star) + \varepsilon - \varepsilon' - \mu_i)_+}{\sqrt{w_i^\star}(L(b^\star) + \varepsilon + \varepsilon' - \mu_i)_+} \right)^2 \\
&\geq \frac{B(b^\star, \boldsymbol{w}^\star)}{C^\star} \cdot \alpha \left(1 - \frac{2\varepsilon' \cdot \sqrt{w_i^\star}}{\sqrt{w_i^\star}(L(b^\star) + \varepsilon + \varepsilon' - \mu_i)_+} \right)^2 \\
&\geq 1 \cdot \alpha \left(1 - \frac{2\varepsilon' \cdot 1}{\sqrt{B(b^\star, \boldsymbol{w}^\star)}} \right)^2 \geq 1, \quad \forall i \in [K],
\end{aligned}$$

where the last line follows from (30), $w_i^\star \leq 1$, and the definition of α. From this result we see that U' and $\boldsymbol{r}' = (r'_1, \ldots, r'_K)$ are a feasible solution of the minimization problem in (P1), and therefore the optimal value $f(\boldsymbol{\mu})$ of (P1) satisfies

$$f(\boldsymbol{\mu}) \leq \sum_{i\in[K]} r'_i = \frac{\alpha}{C^\star} = \frac{1}{C^\star \left(1 - \frac{2\sqrt{K}\varepsilon'}{\varepsilon}\right)}.$$

Combining this fact with (29) we obtain

$$\frac{1}{\max_{b\in b^\star(\boldsymbol{\mu})} \max_{\boldsymbol{w}\in\Delta^K} \inf_{\boldsymbol{\lambda}:b\notin b^\star(\boldsymbol{\lambda})} \sum_{i\in[K]} w_i d(\mu_i, \lambda_i)} \geq 2R^2 f(\boldsymbol{\mu}) \left(1 - \frac{2\sqrt{K}\varepsilon'}{\varepsilon}\right),$$

the RHS of which approaches $2R^2 f(\boldsymbol{\mu})$ as $\varepsilon' \downarrow 0$. □

Proof of Theorem 1. Theorem 1 is an immediate consequence of Lemma 7, Lemma 8, and the fact that any δ-correct BME algorithm can be converted into an M-bin MCP, as well as the choice of $\eta \to 0$. □

A.2 Analysis of the characterizing optimization

Proof of Lemma 2. (i) The statement can be easily obtained from

$$\begin{aligned}\underline{g}(U;\boldsymbol{\mu}) &\in \left(1 - K\frac{(\mu^\star - U + 2\varepsilon)_+^2}{(U-\mu^\star)^2}, 1 - \frac{(\mu^\star - U + 2\varepsilon)_+^2}{(U-\mu^\star)^2}\right)\\ \overline{g}(U;\boldsymbol{\mu}) &\geq \frac{1}{(\mu^\star - U + 2\varepsilon)^3} - \sum_{i\neq i^\star}\frac{1}{(U-\mu^\star)^3}\end{aligned} \tag{31}$$

for $U \in (\mu^\star, \mu^\star + 2\varepsilon)$.

(ii) By the Lagrange multiplier method, we immediately obtain

$$r_i(U;\boldsymbol{\mu}) = \begin{cases}\frac{1}{(U-\mu_i)^2}, & i \neq i^\star\\ \frac{1}{(\mu^\star - U + 2\varepsilon)^2}\left(1 - \sum_{i\neq i^\star}\frac{(\mu_i - U + 2\varepsilon)_+^2}{(U-\mu_i)^2}\right) \vee \frac{1}{(U-\mu^\star)^2} & i = i^\star,\end{cases} \tag{32}$$

where recall that we set one optimal arm as $i^\star$, and therefore the case $i = i^\star$ happens only for one $i \in [K]$ even if there are multiple optimal arms.

Here, $r_{i^\star}(U;\boldsymbol{\mu}) = \frac{1}{(U-\mu^\star)^2}$ holds when

$$\frac{1}{(\mu^\star - U + 2\varepsilon)^2}\left(1 - \sum_{i\neq i^\star}\frac{(\mu_i - U + 2\varepsilon)_+^2}{(U-\mu_i)^2}\right) \leq \frac{1}{(U-\mu^\star)^2} \tag{33}$$

or equivalently

$$\underline{g}(U;\boldsymbol{\mu}) = 1 - \sum_{i\in[K]}\frac{(\mu_i - U + 2\varepsilon)_+^2}{(U-\mu_i)^2} \leq 0, \tag{34}$$

that is, $U \leq \underline{U}(\boldsymbol{\mu})$. In this case,

$$f(U;\boldsymbol{\mu}) = \sum_{i\in[K]}\frac{1}{(U-\mu_i)^2}, \tag{35}$$

which is decreasing in U. Therefore, $U < \underline{U}(\boldsymbol{\mu})$ cannot be the optimal solution $U(\boldsymbol{\mu})$ minimizing $f(U;\boldsymbol{\mu})$, and we obtain $U(\boldsymbol{\mu}) \geq \underline{U}(\boldsymbol{\mu})$, where

$$r_i(U;\boldsymbol{\mu}) = \begin{cases}\frac{1}{(U-\mu_i)^2}, & i \neq i^\star\\ \frac{1}{(\mu^\star - U + 2\varepsilon)^2}\left(1 - \sum_{i\neq i^\star}\frac{(\mu_i - \bar{\mu} - 2\varepsilon)_+^2}{(U-\mu_i)^2}\right) & i = i^\star.\end{cases} \tag{36}$$

and

$$\begin{aligned}f(U;\boldsymbol{\mu}) &= \sum_{i\neq i^\star}\frac{1}{(U-\mu_i)^2} + \frac{1}{(\mu^\star - U + 2\varepsilon)^2}\left(1 - \sum_{i\neq i^\star}\frac{(\mu_i - U + 2\varepsilon)_+^2}{(U-\mu_i)^2}\right)\\ &= \sum_{i\neq i^\star}\frac{1}{(U-\mu_i)^2}\left(1 - \frac{(\mu_i - U + 2\varepsilon)_+^2}{(\mu^\star - U + 2\varepsilon)^2}\right) + \frac{1}{(\mu^\star - U + 2\varepsilon)^2},\end{aligned} \tag{37}$$

which is not necessarily convex in U.

(iii) $U(\boldsymbol{\mu}) \geq \underline{U}(\boldsymbol{\mu})$ is already proved above. By letting $f'(U;\boldsymbol{\mu}) = \frac{\partial f'(U;\boldsymbol{\mu})}{\partial U}$, we obtain for $U \geq \underline{U}(\boldsymbol{\mu})$ that

$$\begin{aligned}
\frac{f'(U;\boldsymbol{\mu})}{2} &= -\sum_{i\in[K]} \frac{1}{(U-\mu_i)^3}\left(1-\frac{(\mu_i-U+2\varepsilon)_+^2}{(\mu^\star-U+2\varepsilon)^2}\right) \\
&\quad + \sum_{i\in[K]} \frac{1}{(U-\mu_i)^2} \frac{(\mu^\star-U+2\varepsilon)(\mu_i-U+2\varepsilon)_+ - (\mu_i-U+2\varepsilon)_+^2}{(\mu^\star-U+2\varepsilon)^3} + \frac{1}{(\mu^\star-U+2\varepsilon)^3} \\
&\geq -\sum_{i\in[K]} \frac{1}{(U-\mu_i)^3}\left(1-\frac{(\mu_i-U+2\varepsilon)_+^2}{(\mu^\star-U+2\varepsilon)^2}\right) + \frac{1}{(\mu^\star-U+2\varepsilon)^3} \\
&= -\sum_{i\neq i^\star} \frac{1}{(U-\mu_i)^3}\left(1-\frac{(\mu_i-U+2\varepsilon)_+^2}{(\mu^\star-U+2\varepsilon)^2}\right) + \frac{1}{(\mu^\star-U+2\varepsilon)^3} \\
&\geq -\sum_{i\neq i^\star} \frac{1}{(U-\mu_i)^3} + \frac{1}{(\mu^\star-U+2\varepsilon)^3} \\
&= \overline{g}(U;\boldsymbol{\mu}). \qquad (38)
\end{aligned}$$

Therefore, when $\overline{g}(U;\boldsymbol{\mu}) > 0$, that is, when $U > \overline{U}(\boldsymbol{\mu})$, we see that $f(U;\boldsymbol{\mu})$ is increasing. Therefore, $U > \overline{U}(\boldsymbol{\mu})$ satisfying $U \geq \underline{U}(\boldsymbol{\mu})$ cannot minimize $f(U;\boldsymbol{\mu})$. Therefore we obtain $U(\boldsymbol{\mu}) \leq \underline{U}(\boldsymbol{\mu}) \vee \overline{U}(\boldsymbol{\mu})$.

(iv) Recall that $\overline{U}(\boldsymbol{\mu})$ is the solution to $\overline{g}(U;\boldsymbol{\mu})$, and $\overline{g}(U;\boldsymbol{\mu})$ is increasing with U in $(\mu^\star, \mu^\star+2\varepsilon)$. Now, observe that

$$\begin{aligned}
\overline{g}(U;\boldsymbol{\mu}) &= \frac{1}{(\mu^\star-U+2\varepsilon)^3} - \sum_{i\neq\hat{i}^\star} \frac{1}{(U-\mu_i)^3} \\
&\geq \frac{1}{(\mu^\star-U+2\varepsilon)^3} - \sum_{i\neq\hat{i}^\star} \frac{1}{(U-\mu^\star)^3} \\
&= \frac{1}{(\mu^\star-U+2\varepsilon)^3} - \frac{K-1}{(U-\mu^\star)^3}, \qquad (39)
\end{aligned}$$

which attains 0 at

$$U = \mu^\star + \frac{2}{1+\frac{1}{(K-1)^{1/3}}}\varepsilon < \mu^\star + 2\varepsilon. \qquad (40)$$

Hence,

$$\overline{U}(\boldsymbol{\mu}) \leq \mu^\star + \frac{2}{1+\frac{1}{(K-1)^{1/3}}}\varepsilon < \mu^\star + 2\varepsilon \qquad (41)$$

and we get

$$\begin{aligned}
r_i(U(\boldsymbol{\mu});\boldsymbol{\mu}) &\leq \frac{1}{(\mu_{i^\star}-U(\boldsymbol{\mu})+2\varepsilon)^2} \\
&\leq \frac{\left((K-1)^{1/3}+1\right)^2}{4\varepsilon^2}. \qquad (42)
\end{aligned}$$

(v) By (32), we have

$$r_{i^\star}(U(\boldsymbol{\mu});\boldsymbol{\mu}) \geq \frac{1}{(U(\boldsymbol{\mu})-\mu_{i^\star})^2}. \qquad (43)$$

Then the statement is immediate from $U(\boldsymbol{\mu}) \leq \overline{U}(\boldsymbol{\mu}) < \mu^\star + 2\varepsilon$ shown in (i) and (iii). □

A.3 Analysis of the confidence ellipsoid

Proof of Lemma 3. Let us define

$$M_t = \sqrt{\frac{\det(V)}{\det(V+V_t)}} \exp\left(\frac{1}{2}\|S_t\|^2_{(V+V_t)^{-1}}\right) \tag{44}$$

where

$$\begin{aligned} S_t &= \sum_{s=1}^{t} \eta_s X_s \\ &= \sum_{s=1}^{t} (Y_t - X_s^\top \theta) X_s. \end{aligned} \tag{45}$$

for observation $Y_t = X_t^\top \theta + \eta_t$ with R-sub-Gaussian η_t.

Then by Theorem 1 of [1], with probability at least $1-\delta$ it holds for all t that

$$\|S_t\|^2_{(V+V_t)^{-1}} \le 2R^2 \log \frac{\det(V+V_t)^{1/2}}{\delta \det(V)^{1/2}}. \tag{46}$$

Here, let us consider our setting with $V = \lambda I$ and regularized estimator $\hat{\mu}_i(t) = \frac{1}{N_i(t)+\lambda} \sum_{m=1}^{N_i(t)} Y_{i,m}$, where $Y_{i,m}$ is the m-th observation from arm i. Then

$$\begin{aligned} \|S_t\|^2_{(V+V_t)^{-1}} &= \sum_{i\in[K]} \frac{1}{N_i(t)+\lambda} ((N_i(t)+\lambda)\hat{\mu}_i(t) - N_i(t)\mu_i)^2 \\ &= \sum_{i\in[K]} \left((N_i(t)+\lambda)\hat{\mu}_i(t)^2 - 2\hat{\mu}_i(t)\mu_i N_i(t) + \frac{N_i(t)^2\mu_i^2}{N_i(t)+\lambda} \right) \\ &\ge \sum_{i\in[K]} \left(N_i(t)\hat{\mu}_i(t)^2 - 2\hat{\mu}_i(t)\mu_i N_i(t) + \frac{N_i(t)^2\mu_i^2}{N_i(t)+\lambda} \right) \\ &= \sum_{i\in[K]} \left(N_i(t)\left(\hat{\mu}_i(t) - \mu_i\right)^2 + \frac{N_i(t)^2\mu_i^2}{N_i(t)+\lambda} - N_i(t)\mu_i^2 \right) \\ &= \sum_{i\in[K]} \left(N_i(t)\left(\hat{\mu}_i(t) - \mu_i\right)^2 - \frac{\lambda N_i(t)\mu_i^2}{N_i(t)+\lambda} \right) \\ &\ge \sum_{i\in[K]} \left(N_i(t)\left(\hat{\mu}_i(t) - \mu_i\right)^2 - \frac{\lambda N_i(t) S^2}{N_i(t)+\lambda} \right) \\ &= \sum_{i\in[K]} N_i(t)\left(\hat{\mu}_i(t) - \mu_i\right)^2 - \lambda S^2 \sum_{i\in[K]} \frac{N_i(t)}{N_i(t)+\lambda} \end{aligned} \tag{47}$$

where[1] $S \ge \max_i |\mu_i|$. From these results, with probability at least $1-\delta$ it holds for all t that

$$\begin{aligned} \sum_{i\in[K]} N_i(t)\left(\hat{\mu}_i(t) - \mu_i\right)^2 &\le 2R^2 \log \frac{\det(V+V_t)^{1/2}}{\delta \det(V)^{1/2}} + \lambda S^2 \sum_{i\in[K]} \frac{N_i(t)}{N_i(t)+\lambda} \\ &= 2R^2 \log \frac{\prod_{i=1}^{K} \sqrt{\lambda + N_i(t)}}{\delta \lambda^{K/2}} + \lambda S^2 \sum_{i\in[K]} \frac{N_i(t)}{N_i(t)+\lambda} \\ &= 2R^2 \log \frac{1}{\delta} + \sum_{i=1}^{K} \left(R^2 \log \frac{\lambda + N_i(t)}{\lambda} + \lambda S^2 \frac{N_i(t)}{N_i(t)+\lambda} \right). \end{aligned} \tag{48}$$

[1] In our settings, the max norm is more natural than the L_2 norm in [1].

While the right-hand side of this expression is computable and sufficient for implementation, for the theoretical analysis it becomes convenient to bound the reminder term by

$$\begin{aligned}
&\sum_{i\in[K]}\left(R^2\log\frac{\lambda+N_i(t)}{\lambda}+\lambda S^2\frac{N_i(t)}{N_i(t)+\lambda}\right)\\
&\leq K\left(R^2\log\frac{\lambda+t/K}{\lambda}+\lambda S^2\frac{t/K}{t/K+\lambda}\right)\\
&\leq K\left(R^2\log t+R^2\log\frac{\lambda/t+1/K}{\lambda}+\lambda S^2\right)\\
&\leq K\left(R^2\log t+\lambda S^2\right) \qquad (49)
\end{aligned}$$

when $t \geq 2$ and $\lambda \geq 2/K$. □

A.4 Analysis of the sample complexity of Algorithm 1

A.4.1 Preliminary

Lemma 9. *For any $t \geq 0$ and $\delta > 0$, we have*

$$\beta(t,\delta) \leq 2R^2\log(1/\delta)+\alpha(t), \qquad (50)$$

where

$$\alpha(t) \coloneqq KR^2\log\left(1+\frac{t}{\lambda K}\right)+\lambda S^2 K. \qquad (51)$$

Moreover, at $t = T_{\max}$, we have

$$\alpha(T_{\max}) \leq KR^2\log\left(\frac{2\xi R^2}{\lambda K}\log(1/\delta)\right)+K\frac{\gamma+\lambda K}{2\xi\log(1/\delta)}+\lambda S^2 K. \qquad (52)$$

When δ is sufficiently small and satisfies $\log(1/\delta) \geq K$, we have

$$\alpha(T_{\max}) \leq KR^2\log\left(\frac{2\xi R^2}{\lambda K}\log(1/\delta)\right)+\frac{\gamma+\lambda K}{2\xi}+\lambda S^2 K. \qquad (53)$$

Proof. Since $\log(1+x/\lambda)$ and $x/(x+\lambda)$ are concave in x for $\lambda > 0$, $\beta(t,\delta)$ defined in (10) is maximized when $N_i(t) = t/K, \forall i \in [K]$. Hence,

$$\begin{aligned}
\beta(t,\delta) &\leq 2R^2\log(1/\delta)+KR^2\log\left(1+\frac{t}{\lambda K}\right)+\lambda S^2 K\frac{t/K}{t/K+\lambda}\\
&\leq 2R^2\log(1/\delta)+KR^2\log\left(1+\frac{t}{\lambda K}\right)+\lambda S^2 K. \qquad (54)
\end{aligned}$$

By the definition of $T_{\max}$ in (15), we have

$$\begin{aligned}
\alpha(T_{\max}) &= KR^2\log\left(1+\frac{T_{\max}}{\lambda K}\right)+\lambda S^2 K\\
&= KR^2\log\left(\frac{2\xi R^2}{\lambda K}\log(1/\delta)+\frac{\gamma}{\lambda K}+1\right)+\lambda S^2 K\\
&= KR^2\log\left(\frac{2\xi R^2}{\lambda K}\log(1/\delta)\right)+KR^2\log\left(1+\frac{\frac{\gamma}{\lambda K}+1}{\frac{2\xi R^2}{\lambda K}\log(1/\delta)}\right)+\lambda S^2 K\\
&\leq KR^2\log\left(\frac{2\xi R^2}{\lambda K}\log(1/\delta)\right)+KR^2\frac{\frac{\gamma}{\lambda K}+1}{\frac{2\xi R^2}{\lambda K}\log(1/\delta)}+\lambda S^2 K \quad \text{(by } \log(1+x)\leq x\text{)}
\end{aligned} \qquad (55)$$

which gives (52). When $\log(1/\delta) \geq K$ holds, the second term in the previous expression is bounded as follows:

$$KR^2\frac{\frac{\gamma}{\lambda K}+1}{\frac{2\xi R^2}{\lambda K}\log(1/\delta)} \leq \frac{\gamma+\lambda K}{2\xi} \qquad (56)$$

which gives (53). □

In the following, we derive a couple of lemmas on the properties of $f(U;\boldsymbol{\mu})$ and $\{r_i(U;\boldsymbol{\mu})\}_i$ that are needed for the sample complexity analysis. Define

$$f_k(r,U;\boldsymbol{\mu}) = \sum_{i\neq k} \frac{1}{(U-\mu_i)^2} + \left(r \vee \frac{1}{(U-\mu_k)^2}\right). \tag{57}$$

An intuition is that $f_k(r,U;\boldsymbol{\mu})$ corresponds to the sample complexity rate to ensure that the best mean is at most U given that arm k is pulled at least r times.

Lemma 10. *Let $\varepsilon_1 \in (0,\varepsilon/2)$, $\varepsilon_2 \in (0,\varepsilon)$ and take $\boldsymbol{\mu}$ arbitrary. Let $\boldsymbol{\mu}'$ be such that $|\mu'_i - \mu_i| \le \varepsilon_1$ for all i. For $k^\star = \operatorname{argmax}_i \mu'_i$, define $r^\star = r_{k^\star}(\boldsymbol{\mu}')$. Then,*

$$f_{k^\star}(r^\star, U(\boldsymbol{\mu}')-\varepsilon_2;\boldsymbol{\mu}') \le f(\boldsymbol{\mu}) + \frac{4K\varepsilon_1}{\varepsilon(\varepsilon-2\varepsilon_1)^2} + \frac{2K\varepsilon_2}{\varepsilon(\varepsilon-\varepsilon_2)^2}. \tag{58}$$

In this lemma, $\boldsymbol{\mu}'$ intuitively corresponds to the estimator of $\boldsymbol{\mu}$ when the proportion of the optimal arm is determined. We can use this lemma to show that the sample complexity is close to the optimal one, $f(\boldsymbol{\mu})$, as far as the estimator $\boldsymbol{\mu}'$ is not far from $\boldsymbol{\mu}$.

The technical difficulty of this lemma comes from the non-convexity of the characterizing optimization (P1). While we can easily show $f_{k^\star}(r^\star,U;\boldsymbol{\mu}) = f(\boldsymbol{\mu})$ if $r^\star = r_{i^\star}(\boldsymbol{\mu})$ and $U = U_{i^\star}(r;\boldsymbol{\mu})$, we cannot guarantee that the optimal allocation $r_{k^\star}(\boldsymbol{\mu}')$ for $\boldsymbol{\mu}'$ becomes close to the optimal allocation $r_{k^\star}(\boldsymbol{\mu})$ for $\boldsymbol{\mu}$ due to the non-convexity. This lemma can be used to guarantee that if r and U are determined based on an estimator close to $\boldsymbol{\mu}$ then the sample complexity becomes close to the optimal one despite the non-convexity of (P1).

Proof of Lemma 10. Before evaluating $f_{k^\star}(r^\star, U(\boldsymbol{\mu}')-\varepsilon_2;\boldsymbol{\mu}')$ we prepare several elementary relations. First we have

$$\begin{aligned}
&\frac{1}{(U(\boldsymbol{\mu}')-\mu'_i-\varepsilon_2)^2}\\
&= \frac{1}{(U(\boldsymbol{\mu}')-\mu'_i)^2} + \left(\frac{1}{(U(\boldsymbol{\mu}')-\mu'_i-\varepsilon_2)^2} - \frac{1}{(U(\boldsymbol{\mu}')-\mu'_i)^2}\right)\\
&= \frac{1}{(U(\boldsymbol{\mu}')-\mu'_i)^2} + \frac{\varepsilon_2(2(U(\boldsymbol{\mu}')-\mu'_i)-\varepsilon_2)}{(U(\boldsymbol{\mu}')-\mu'_i-\varepsilon_2)^2(U(\boldsymbol{\mu}')-\mu'_i)^2}\\
&\le \frac{1}{(U(\boldsymbol{\mu}')-\mu'_i)^2} + \frac{\varepsilon_2(2(U(\boldsymbol{\mu}')-\mu'_i)-\varepsilon_2)}{(U(\boldsymbol{\mu}')-\mu'_i-\varepsilon_2)^2\varepsilon^2} \quad \text{(by } U(\boldsymbol{\mu}') \ge \max_i \mu'_i + \varepsilon \text{ from Lemma 2 (i)(iii))}\\
&\le \frac{1}{(U(\boldsymbol{\mu}')-\mu'_i)^2} + \frac{\varepsilon_2(2\varepsilon-\varepsilon_2)}{(\varepsilon-\varepsilon_2)^2\varepsilon^2} \quad \text{(by } U(\boldsymbol{\mu}') \ge \max_i \mu'_i + \varepsilon \text{ from Lemma 2 (i)(iii))}\\
&\le \frac{1}{(U(\boldsymbol{\mu}')-\mu'_i)^2} + \frac{2\varepsilon_2}{(\varepsilon-\varepsilon_2)^2\varepsilon}.
\end{aligned} \tag{59}$$

and similarly,

$$\frac{1}{(U(\boldsymbol{\mu})-\mu_i-2\varepsilon_1)^2} \le \frac{1}{(U(\boldsymbol{\mu})-\mu_i)^2} + \frac{4\varepsilon_1}{(\varepsilon-2\varepsilon_1)^2\varepsilon}. \tag{60}$$

For any $k \in [K]$, we also have

$$
\begin{aligned}
&\sum_{i \neq k} \frac{(\mu'_i - U(\boldsymbol{\mu}) + 2\varepsilon + \varepsilon_1)_+^2}{(U(\boldsymbol{\mu}) - \mu'_i - \varepsilon_1)^2} \\
&= \sum_{i \neq k, i^\star} \frac{(\mu'_i - U(\boldsymbol{\mu}) + 2\varepsilon + \varepsilon_1)_+^2}{(U(\boldsymbol{\mu}) - \mu'_i - \varepsilon_1)^2} + \mathbb{1}[k \neq i^\star] \frac{(\mu'_{i^\star} - U(\boldsymbol{\mu}) + 2\varepsilon + \varepsilon_1)_+^2}{(U(\boldsymbol{\mu}) - \mu'_{i^\star} - \varepsilon_1)^2} \\
&\geq \sum_{i \neq k, i^\star} \frac{(\mu_i - U(\boldsymbol{\mu}) + 2\varepsilon)_+^2}{(U(\boldsymbol{\mu}) - \mu_i)^2} + \mathbb{1}[k \neq i^\star] \frac{(\mu_{i^\star} - U(\boldsymbol{\mu}) + 2\varepsilon)_+^2}{(U(\boldsymbol{\mu}) - \mu_{i^\star})^2} \\
&= \sum_{i \neq k, i^\star} \frac{(\mu_i - U(\boldsymbol{\mu}) + 2\varepsilon)_+^2}{(U(\boldsymbol{\mu}) - \mu_i)^2} + \mathbb{1}[k \neq i^\star] \frac{(\max_i \mu_i - U(\boldsymbol{\mu}) + 2\varepsilon)_+^2}{(U(\boldsymbol{\mu}) - \max_i \mu_i)^2} \\
&\geq \sum_{i \neq k, i^\star} \frac{(\mu_i - U(\boldsymbol{\mu}) + 2\varepsilon)_+^2}{(U(\boldsymbol{\mu}) - \mu_i)^2} + \mathbb{1}[k \neq i^\star] \frac{(\mu_k - U(\boldsymbol{\mu}) + 2\varepsilon)_+^2}{(U(\boldsymbol{\mu}) - \mu_k)^2} \\
&= \sum_{i \neq i^\star} \frac{(\mu_i - U(\boldsymbol{\mu}) + 2\varepsilon)_+^2}{(U(\boldsymbol{\mu}) - \mu_i)^2}. \qquad (61)
\end{aligned}
$$

We also obtain the following from $|\mu'_i - \mu_i| \leq \varepsilon_1$:

$$
\sum_{i \neq k} \frac{1}{(U(\boldsymbol{\mu}) - \mu'_i - \varepsilon_1)^2} \leq \sum_{i \neq i^\star} \frac{1}{(U(\boldsymbol{\mu}) - \mu_i - 2\varepsilon_1)^2}. \tag{62}
$$

Since $r^\star = r_{k^\star}(\boldsymbol{\mu}')$ for $k^\star = \mathrm{argmax}_i \, \mu'_i$,

$$
r^\star = \frac{1}{(\mu'_{k^\star} - U(\boldsymbol{\mu}') + 2\varepsilon)^2} \left(1 - \sum_{i \neq k^\star} \frac{(\mu'_i - U(\boldsymbol{\mu}') + 2\varepsilon)_+^2}{(U(\boldsymbol{\mu}') - \mu'_i)^2} \right) \geq \frac{1}{(U(\boldsymbol{\mu}') - \mu'_{k^\star})^2} \tag{63}
$$

by Lemma 2 (ii).

Now we evaluate $f_{k^\star}(r^\star, U(\boldsymbol{\mu}') - \varepsilon_2; \boldsymbol{\mu}')$ as follows.

$$
\begin{aligned}
&f_{k^\star}(r^\star, U(\boldsymbol{\mu}') - \varepsilon_2; \boldsymbol{\mu}') \\
&= \sum_{i \neq k^\star} \frac{1}{(U(\boldsymbol{\mu}') - \mu'_i - \varepsilon_2)^2} + \left(r^\star \vee \frac{1}{(U(\boldsymbol{\mu}') - \mu'_{k^\star} - \varepsilon_2)^2} \right). \\
&\leq \sum_{i \neq k^\star} \frac{1}{(U(\boldsymbol{\mu}') - \mu'_i)^2} + \left(r^\star \vee \frac{1}{(U(\boldsymbol{\mu}') - \mu'_{k^\star})^2} \right) + \frac{2K\varepsilon_2}{\varepsilon(\varepsilon - \varepsilon_2)^2} \quad \text{(by (59))} \\
&= \sum_{i \neq k^\star} \frac{1}{(U(\boldsymbol{\mu}') - \mu'_i)^2} + \frac{1}{(\mu'_{k^\star} - U(\boldsymbol{\mu}') + 2\varepsilon)^2} \left(1 - \sum_{i \neq k^\star} \frac{(\mu'_i - U(\boldsymbol{\mu}') + 2\varepsilon)_+^2}{(U(\boldsymbol{\mu}') - \mu'_i)^2} \right) + \frac{2K\varepsilon_2}{\varepsilon(\varepsilon - \varepsilon_2)^2} \\
&\qquad\qquad\qquad \text{(by (63))} \\
&= f(U(\boldsymbol{\mu}'); \boldsymbol{\mu}') + \frac{2K\varepsilon_2}{\varepsilon(\varepsilon - \varepsilon_2)^2} \\
&= \inf_U f(U; \boldsymbol{\mu}') + \frac{2K\varepsilon_2}{\varepsilon(\varepsilon - \varepsilon_2)^2} \\
&= \inf_U f(U; \boldsymbol{\mu}' + \varepsilon_1 \mathbf{1}) + \frac{2K\varepsilon_2}{\varepsilon(\varepsilon - \varepsilon_2)^2} \qquad (64) \\
&\leq f(U(\boldsymbol{\mu}); \boldsymbol{\mu}' + \varepsilon_1 \mathbf{1}) + \frac{2K\varepsilon_2}{\varepsilon(\varepsilon - \varepsilon_2)^2} \\
&= \sum_{i \neq k^\star} \frac{1}{(U(\boldsymbol{\mu}) - \mu'_i - \varepsilon_1)^2} + \frac{1}{(\mu'_{k^\star} - U(\boldsymbol{\mu}) + 2\varepsilon + \varepsilon_1)^2} \left(1 - \sum_{i \neq k^\star} \frac{(\mu'_i - U(\boldsymbol{\mu}) + 2\varepsilon + \varepsilon_1)_+^2}{(U(\boldsymbol{\mu}) - \mu'_i - \varepsilon_1)^2} \right) + \frac{2K\varepsilon_2}{\varepsilon(\varepsilon - \varepsilon_2)^2}
\end{aligned}
$$

$$\leq \sum_{i \neq i^\star} \frac{1}{(U(\boldsymbol{\mu}) - \mu_i - 2\varepsilon_1)^2} + \frac{1}{(\mu'_{k^\star} - U(\boldsymbol{\mu}) + 2\varepsilon + \varepsilon_1)^2} \underbrace{\left(1 - \sum_{i \neq i^\star} \frac{(\mu_i - U(\boldsymbol{\mu}) + 2\varepsilon)_+^2}{(U(\boldsymbol{\mu}) - \mu_i)^2}\right)}_{\geq 0 \text{ by } \underline{g}(U(\boldsymbol{\mu})) \geq 0} + \frac{2K\varepsilon_2}{\varepsilon(\varepsilon - \varepsilon_2)^2}$$

(by (61) and (62))

$$\leq \sum_{i \neq i^\star} \frac{1}{(U(\boldsymbol{\mu}) - \mu_i - 2\varepsilon_1)^2} + \frac{1}{(\mu^\star - U(\boldsymbol{\mu}) + 2\varepsilon)^2} \left(1 - \sum_{i \neq i^\star} \frac{(\mu_i - U(\boldsymbol{\mu}) + 2\varepsilon)_+^2}{(U(\boldsymbol{\mu}) - \mu_i)^2}\right) + \frac{2K\varepsilon_2}{\varepsilon(\varepsilon - \varepsilon_2)^2}$$

$$\leq \sum_{i \neq i^\star} \frac{1}{(U(\boldsymbol{\mu}) - \mu_i)^2} + \frac{1}{(\mu^\star - U(\boldsymbol{\mu}) + 2\varepsilon)^2} \left(1 - \sum_{i \neq i^\star} \frac{(\mu_i - U(\boldsymbol{\mu}) + 2\varepsilon)_+^2}{(U(\boldsymbol{\mu}) - \mu_i)^2}\right) + \frac{4K\varepsilon_1}{\varepsilon(\varepsilon - 2\varepsilon_1)^2} + \frac{2K\varepsilon_2}{\varepsilon(\varepsilon - \varepsilon_2)^2}$$

(by (60))

$$= f(\boldsymbol{\mu}) + \frac{4K\varepsilon_1}{\varepsilon(\varepsilon - 2\varepsilon_1)^2} + \frac{2K\varepsilon_2}{\varepsilon(\varepsilon - \varepsilon_2)^2} \quad \text{(by the definition of } f(\boldsymbol{\mu})) .$$

Here, in (64), $\mathbf{1}$ is the all-one vector and the equality comes from the shift-invariance in $\boldsymbol{\mu}$ of the original problem (3). □

A.4.2 Analysis of $T_{\max}$

Lemma 11. *When $K \leq \log(1/\delta)$ holds, the algorithm stops by $T_{\max}$ almost surely.*

Proof. We begin by bounding the number of times each arm is pulled before the algorithm terminates. Recall that at Step 5, the algorithm selects an arm, and that the tentative best arm $\hat{i}^\star$, identified at Step 14, is treated differently from the others. Therefore, we analyze two separate cases depending on whether arm k is equal to $\hat{i}^\star$ or not.

Case 1: $k \neq \hat{i}^\star$. For the arm $k \neq \hat{i}^\star$ to be pulled in round t, it must be the maximizer $k^\star$ at Step 9 in round $t-1$. Since the algorithm has not stopped in round $t-1$, we must have

$$\begin{aligned}
\beta(t-1,\delta) &> \sum_{i \in [K]} N_i(t-1)(\hat{\mu}_i(t-1) - U(t-1) + 2\varepsilon)_+^2 \\
&\geq N_k(t-1)(\hat{\mu}_k(t-1) - U(t-1) + 2\varepsilon)_+^2 \\
&= N_k(t-1)\left(\hat{\mu}_k(t-1) - \hat{\mu}_k(t-1) - \sqrt{\frac{\beta(t-1,\delta)}{N_k(t-1)}} + 2\varepsilon\right)_+^2 \\
&= \left(-\sqrt{\beta(t-1,\delta)} + 2\varepsilon\sqrt{N_k(t-1)}\right)_+^2 ,
\end{aligned} \tag{65}$$

implying that

$$N_k(t-1) < \frac{\beta(t-1,\delta)}{\varepsilon^2} \leq \frac{\beta(T_{\max},\delta)}{\varepsilon^2}. \tag{66}$$

Since this relation holds whenever arm k is pulled in round $t \leq \tau$, we must have

$$N_k(\tau) < \frac{\beta(T_{\max},\delta)}{\varepsilon^2} + 1. \tag{67}$$

Case 2: $k = \hat{i}^\star$. For the arm $\hat{i}^\star$ to be pulled in round t, it must be the maximizer $k^\star$ at Step 9 in round $t-1$ or it satisfies $N_{\hat{i}^\star}(t-1) < r_{\hat{i}^\star}\beta(t-1,\delta)$ at Step 5 in round t. However, by the same argument as above, the first situation implies

$$N_{\hat{i}^\star}(t-1) < \frac{\beta(T_{\max},\delta)}{\varepsilon^2}. \tag{68}$$

The second situation can happen only when

$$N_{\hat{i}^\star}(t-1) < \frac{\left((K-1)^{1/3}+1\right)^2}{4\varepsilon^2}\beta(T_{\max},\delta), \tag{69}$$

by Lemma 2(v).

From the above argument on two cases, we obtain

$$N_k(\tau) \le \begin{cases} \frac{\beta(T_{\max},\delta)}{\varepsilon^2} + 1 & k \neq \hat{i}^\star \\ \frac{\left((K-1)^{1/3}+1\right)^2}{4\varepsilon^2}\beta(T_{\max},\delta) + 1 & k = \hat{i}^\star, \end{cases} \tag{70}$$

which means

$$\begin{aligned} \tau &= \sum_{k\in[K]} N_k(\tau) \\ &\le \left(\frac{K-1}{\varepsilon^2} + \frac{\left((K-1)^{1/3}+1\right)^2}{4\varepsilon^2}\right)\beta(T_{\max},\delta) + K \\ &= \xi\beta(T_{\max},\delta) + K, \end{aligned} \tag{71}$$

where ξ is as defined in (16).

By Lemma 9, we have under $\log(1/\delta) \ge K$ that

$$\begin{aligned} \beta(T_{\max},\delta) &\le 2R^2 \log\frac{1}{\delta} + KR^2 \log\left(\frac{2\xi R^2}{\lambda K}\log\frac{1}{\delta}\right) + \frac{\lambda K + \gamma}{2\xi} + \lambda S^2 K \\ &= \frac{T_{\max} - K}{\xi}, \end{aligned} \tag{72}$$

where the equality follows from the definition of γ in (17). This together with (71) implies

$$\tau \le \xi\beta(T_{\max},\delta) + K \le T_{\max}. \tag{73}$$

Hence, the algorithm must stop within $T_{\max}$ rounds almost surely when $\log(1/\delta) \ge K$ holds. □

A.4.3 Stopping time under $\mathcal{A}(c)$

We will first show, in Lemma 12, that the estimation error in $\hat{\mu}_i(t)$ after the phase shift (i.e., for $t \ge \tau_1$) can be made arbitrarily small under the event $\mathcal{A}(c)$ with a sufficiently small c. This will then be used in Lemma 13 to bound the stopping time τ under $\mathcal{A}(c)$.

Lemma 12. *Let $\Delta_{\max} := \max_i(\mu^\star - \mu_i)$ and $\varepsilon'' := 2\varepsilon\left(1 + \frac{\alpha(T_{\max})}{2R^2\log\frac{1}{\delta}}\right)$ for the α defined in Lemma 9. Then, under $\mathcal{A}(c)$ with $c \in \left[0, \frac{\varepsilon_c}{\Delta_{\max}+\varepsilon''+2\varepsilon_c}\right]$ for any $\varepsilon_c > 0$, we have*

$$|\hat{\mu}_i(t) - \mu_i| \le \varepsilon_c, \forall t \ge \tau_1, \forall i \in [K]. \tag{74}$$

Proof. Let τ_1 be the time of the phase shift, that is,

$$\tau_1 = \min\{t : \underline{g}(U(t); \hat{\boldsymbol{\mu}}(t)) \vee \overline{g}(U(t); \hat{\boldsymbol{\mu}}(t)) \le 0\}. \tag{75}$$

Observe that $\underline{g}(U(t); \hat{\boldsymbol{\mu}}(t)) \vee \overline{g}(U(t); \hat{\boldsymbol{\mu}}(t)) \le 0$ is equivalent to $U(t) \le \underline{U}(\hat{\boldsymbol{\mu}}(t)) \vee \overline{U}(\hat{\boldsymbol{\mu}}(t))$ and that $\underline{U}(\hat{\boldsymbol{\mu}}(t)) \vee \overline{U}(\hat{\boldsymbol{\mu}}(t)) < \hat{\mu}^\star(t) + 2\varepsilon$ by Lemma 2(i). These imply that

$$\begin{aligned} \tau_1 &= \min\{t : \underline{g}(U(t); \hat{\boldsymbol{\mu}}(t)) \vee \overline{g}(U(t); \hat{\boldsymbol{\mu}}(t)) \le 0\} \\ &\ge \min\left\{t : U(t) < \hat{\mu}_{k^\star(t)}(t) + 2\varepsilon\right\} \\ &= \min\left\{t : N_{\hat{k}^\star(t)}(t) > \frac{\beta(t,\delta)}{(2\varepsilon)^2}\right\}, \end{aligned} \tag{76}$$

where $\hat{k}^\star(t) := \operatorname{argmax}_i \hat{\mu}_i(t)$.

Consider the event

$$\mathcal{B} = \left\{U(t) \le \mu^\star + 2(1+c)\varepsilon\sqrt{\frac{\beta(t,\delta)}{A(\tau_1)}}, \forall t \in \{\tau_1, \tau_1+1, \ldots, T_{\max}\}\right\}. \tag{77}$$

Then $\mathcal{B}$ always holds under $\mathcal{A}(c)$ because

$$\begin{aligned}
U(t) &= \max_{k\in[K]}\left\{\hat{\mu}_k(t)+\sqrt{\frac{\beta(t,\delta)}{N_k(t)}}\right\} \\
&= \hat{\mu}_{\hat{k}^\star}(t)+\sqrt{\frac{\beta(t,\delta)}{N_{\hat{k}^\star}(t)}} \\
&\le \mu_{\hat{k}^\star}+(1+c)\sqrt{\frac{\beta(t,\delta)}{N_{\hat{k}^\star}(t)}} \quad (\text{under } \mathcal{A}(c)) \\
&\le \mu^\star+(1+c)\sqrt{\frac{\beta(t,\delta)}{N_{\hat{k}^\star}(t)}} \\
&\le \mu^\star+(1+c)\sqrt{\frac{\beta(t,\delta)}{N_{\hat{k}^\star}(\tau_1)}} \quad (\text{by } t\ge\tau_1) \\
&\le \mu^\star+2(1+c)\varepsilon\sqrt{\frac{\beta(t,\delta)}{A(\tau_1)}} \quad (\text{by (76)}) \qquad (78)
\end{aligned}$$

Hence, under $\mathcal{B}\supseteq\mathcal{A}(c)$, we have

$$\begin{aligned}
\mu_i+(1-c)\sqrt{\frac{\beta(t,\delta)}{N_i(t)}} &\le \hat{\mu}_i(t)+\sqrt{\frac{\beta(t,\delta)}{N_i(t)}} \quad (\text{under } \mathcal{A}(c)) \\
&\le U(t) \\
&\le \mu^\star+2(1+c)\varepsilon\sqrt{\frac{\beta(t,\delta)}{A(\tau_1)}} \quad (\text{by (78)}) \qquad (79)
\end{aligned}$$

for any $t\ge\tau_1, i\in[K]$. Therefore, under $\mathcal{A}(c)$, we have

$$\begin{aligned}
|\hat{\mu}_i(t)-\mu_i| &\le c\sqrt{\frac{\beta(t,\delta)}{N_i(t)}} \\
&\le \frac{c}{1-c}(\mu^\star-\mu_i)+\frac{2c(1+c)}{1-c}\varepsilon\sqrt{\frac{\beta(t,\delta)}{A(\tau_1)}} \quad (\text{by (79)}) \\
&\le \frac{c}{1-c}\Delta_{\max}+\frac{2c(1+c)}{1-c}\varepsilon\sqrt{\frac{\beta(T_{\max},\delta)}{A(0)}} \quad (\text{by the definition of } \Delta_{\max}\text{; } A \text{ is increasing}) \\
&\le \frac{c}{1-c}\Delta_{\max}+\frac{2c(1+c)}{1-c}\varepsilon\sqrt{1+\frac{\alpha(T_{\max})}{2R^2\log\frac{1}{\delta}}} \quad (\text{by (10) with } t=0 \text{ and Lemma 9}) \\
&= \frac{c}{1-c}\left(\Delta_{\max}+(1+c)\varepsilon''\right). \qquad (80)
\end{aligned}$$

To prove (74), it suffices to show that

$$g(c) \coloneqq c\Delta_{\max}+c(1+c)\varepsilon''-(1-c)\varepsilon_c\le 0 \qquad (81)$$

for any $\varepsilon_c>0$ and $c\in\left[0,\frac{\varepsilon_c}{\Delta_{\max}+\varepsilon''+2\varepsilon_c}\right]$. Since $g(c)$ is a quadratic function with $g(0)<0$, it is straightforward to verify that $g(c)\le 0$ for

$$c\in\left[0,\frac{D}{2\varepsilon''}\left(\sqrt{1+\frac{4\varepsilon''\varepsilon_c}{D^2}}-1\right)\right] \qquad (82)$$

where

$$D\coloneqq\Delta_{\max}+\varepsilon''+\varepsilon_c \qquad (83)$$

Since $\sqrt{1+x} \geq 1 + \frac{x}{2(1+x/4)}$ for $x > 0$, the upper bound on c can be simplified to

$$\begin{aligned}\frac{D}{2\varepsilon''}\left(\sqrt{1+\frac{4\varepsilon''\varepsilon_c}{D^2}}-1\right) &\geq \frac{D}{2\varepsilon''}\frac{\frac{4\varepsilon''\varepsilon_c}{D^2}}{2\left(1+\frac{\varepsilon''\varepsilon_c}{D^2}\right)} \\ &= \frac{\varepsilon_c}{D+\frac{\varepsilon''\varepsilon_c}{D}}, \\ &\geq \frac{\varepsilon_c}{D+\varepsilon_c} \quad (\text{since } D > \varepsilon''), \end{aligned} \tag{84}$$

which completes the proof of the lemma. □

Lemma 13. *Under $\mathcal{A}(c)$ with c in the range specified in Lemma 12 for a sufficiently small ε_c such that $\varepsilon_c < \varepsilon/2$ and $2c\varepsilon + \varepsilon_c < \varepsilon$, the stopping time τ of Algorithm 1 satisfies*

$$\tau \leq (1+c)^2\left(f(\boldsymbol{\mu}) + \frac{4K\varepsilon_c}{\varepsilon(\varepsilon-\varepsilon_c)} + \frac{2K(2c\varepsilon+\varepsilon_c)}{\varepsilon(\varepsilon-2c\varepsilon-\varepsilon_c)}\right)\beta(T_{\max},\delta) + K. \tag{85}$$

Proof. We prove the lemma by showing that sufficiently large $N_i(t)$'s make the stopping condition in Step 10 of the algorithm satisfied. To derive a lower bound on a term involved in the stopping condition, consider

$$\begin{aligned} \text{minimize: } & \frac{(x-U+2\varepsilon)_+}{y} \\ \text{subject to: } & x + cy \geq \mu, \\ & x + y \leq U, \\ & y > 0, \end{aligned} \tag{86}$$

where $c \in (0,1)$.

This optimization problem is feasible only when $U \geq \mu$, and in this case, we can see that the optimal value is

$$\left(\frac{2(1-c)\varepsilon}{U-\mu}-1\right)_+ = \left(\frac{\mu-(U+2c\varepsilon)+2\varepsilon}{U-\mu}\right)_+. \tag{87}$$

Indeed, when $U \geq \mu + 2(1-c)\varepsilon$, the objective value can achieve 0 with $x = U - 2\varepsilon$ and $y = 2\varepsilon$. Consider the case where $U < \mu + 2(1-c)\varepsilon$. Observe that the minimum is achieved with $x = \mu - cy$ for any y. With this x, we must have $y \leq \frac{U-\mu}{1-c}$. Hence,

$$\begin{aligned}\frac{x-U+2\varepsilon}{y} &\geq \frac{\mu-cy-U+2\varepsilon}{y} \\ &= \frac{\mu-U+2\varepsilon}{y} - c \\ &\geq \frac{2(1-c)\varepsilon}{U-\mu} - 1, \end{aligned} \tag{88}$$

which is nonnegative under $U < \mu + 2(1-c)\varepsilon$. This lower bound is achieved with $y = \frac{U-\mu}{1-c}$ and $x = \mu - cy$.

Therefore, under $\mathcal{A}(c)$, we have

$$\frac{1}{\beta(t,\delta)} \cdot N_k(t)(\hat{\mu}_k(t) - U(t) + 2\,\varepsilon)_+^2 \geq \left(\frac{\mu_k-(U(t)+2c\varepsilon)+2\varepsilon}{U(t)-\mu_k}\right)_+^2. \tag{89}$$

Similarly, if we additionally have $N_{\hat{i}^\star}(t) \geq r_{\hat{i}^\star}\beta(t,\delta)$, then we can show

$$\begin{aligned}&\frac{1}{\beta(t,\delta)} \cdot N_{\hat{i}^\star}(t)(\hat{\mu}_{\hat{i}^\star}(t) - U(t) + 2\,\varepsilon)_+^2 \\ &\geq r_{\hat{i}^\star}\left(\mu_{\hat{i}^\star} - \left(U(t) + \frac{c}{\sqrt{r_{\hat{i}^\star}}}\right) + 2\varepsilon\right)_+^2 \vee \left(\frac{\mu_{\hat{i}^\star}-(U(t)+2c\varepsilon)+2\varepsilon}{U(t)-\mu_{\hat{i}^\star}}\right)_+^2. \end{aligned} \tag{90}$$

To see this, consider the optimization problem with the additional constraint of

$$y \leq \sqrt{\frac{1}{r}} \tag{91}$$

Similar to the previous argument, the minimum is achieved with $x = \mu - cy$ for any y. With this x, we must have $y \leq \frac{U-\mu}{1-c} \wedge \sqrt{\frac{1}{r}}$. Hence,

$$\begin{aligned} \frac{x - U + 2\varepsilon}{y} &\geq \frac{\mu - U + 2\varepsilon}{y} - c \\ &\geq \left(\frac{2(1-c)\varepsilon}{U - \mu} - 1\right) \vee \sqrt{r}\left(\mu - \left(U + \frac{c}{\sqrt{r}}\right) + 2\varepsilon\right), \end{aligned} \tag{92}$$

which is achieved with $y = \frac{U-\mu}{1-c} \wedge \sqrt{\frac{1}{r}}$ and $x = \mu - cy$.

From these results, under $\mathcal{A}(c)$, if $N_{\hat{i}^\star}(t) \geq r_{\hat{i}^\star}\beta(t,\delta)$ holds, then

$$\begin{aligned} &\frac{1}{\beta(t,\delta)} \sum_{k \in [K]} N_k(t)(\hat{\mu}_k(t) - U(t) + 2\,\varepsilon)_+^2 \\ &\geq \sum_{k \neq \hat{i}^\star} \left(\frac{\mu_k - (U(t) + 2c\varepsilon) + 2\varepsilon}{U(t) - \mu_k}\right)_+^2 \\ &\quad + r_{\hat{i}^\star}\left(\mu_{\hat{i}^\star} - \left(U(t) + \frac{c}{\sqrt{r_{\hat{i}^\star}}}\right) + 2\varepsilon\right)_+^2 \vee \left(\frac{\mu_{\hat{i}^\star} - (U(t) + 2c\varepsilon) + 2\varepsilon}{U(t) - \mu_{\hat{i}^\star}}\right)_+^2. \end{aligned} \tag{93}$$

Therefore, if $|\hat{\mu}_k(\tau_1) - \mu_k| \leq \varepsilon_c$ additionally holds, then

$$\begin{aligned} &\frac{1}{\beta(t,\delta)} \sum_{k \in [K]} N_k(t)(\hat{\mu}_k(t) - U(t) + 2\,\varepsilon)_+^2 \\ &\geq \sum_{k \neq \hat{i}^\star} \left(\frac{\hat{\mu}_k(\tau_1) - (U(t) + 2c\varepsilon + \varepsilon_c) + 2\varepsilon}{(U(t) + \varepsilon_c) - \hat{\mu}_k(\tau_1)}\right)_+^2 \\ &\quad + r_{\hat{i}^\star}\left(\hat{\mu}_{\hat{i}^\star}(\tau_1) - \left(U(t) + \frac{c}{\sqrt{r_{\hat{i}^\star}}} + \varepsilon_c\right) + 2\varepsilon\right)_+^2 \vee \left(\frac{\hat{\mu}_{\hat{i}^\star}(\tau_1) - (U(t) + 2c\varepsilon + \varepsilon_c) + 2\varepsilon}{(U(t) + \varepsilon_c) - \hat{\mu}_{\hat{i}^\star}(\tau_1)}\right)_+^2. \end{aligned} \tag{94}$$

From this discussion with the definition of r, if

$$U(t) \leq U(\hat{\boldsymbol{\mu}}(\tau_1)) - \max\left\{2c\varepsilon, \frac{c}{\sqrt{r_{\hat{i}^\star}}}\right\} - \varepsilon_c \tag{95}$$

holds, then we can show that

$$\frac{1}{\beta(t,\delta)} \sum_{k \in [K]} N_k(t)(\hat{\mu}_k(t) - U(t) + 2\,\varepsilon)_+^2 \geq 1, \tag{96}$$

that is, the algorithm terminates. To show this, let $\hat{U} \coloneqq U(\hat{\boldsymbol{\mu}}(\tau_1))$. Then, since (94) is decreasing with $U(t)$, we have under (95) that

$$\begin{aligned}
&\frac{1}{\beta(t,\delta)} \sum_{k\in[K]} N_k(t)(\hat{\mu}_k(t) - U(t) + 2\,\varepsilon)_+^2 \\
&\geq \sum_{k\neq \hat{i}^\star} \left(\frac{\hat{\mu}_k(\tau_1) - (U(t) + 2c\varepsilon + \varepsilon_c) + 2\varepsilon}{(U(t)+\varepsilon_c) - \hat{\mu}_k(\tau_1)} \right)_+^2 + r_{\hat{i}^\star} \left(\hat{\mu}_{\hat{i}^\star}(\tau_1) - \left(U(t) + \frac{c}{\sqrt{r_{\hat{i}^\star}}} + \varepsilon_c \right) + 2\varepsilon \right)_+^2 \\
&\geq \sum_{k\neq \hat{i}^\star} \left(\frac{\hat{\mu}_k(\tau_1) - \hat{U} + 2\varepsilon}{\hat{U} - \hat{\mu}_k(\tau_1) - 2c\varepsilon} \right)_+^2 + r_{\hat{i}^\star} \left(\hat{\mu}_{\hat{i}^\star}(\tau_1) - \hat{U} + 2\varepsilon \right)_+^2 \\
&\geq 1 + \sum_{k\neq \hat{i}^\star} \left(\frac{\hat{\mu}_k(\tau_1) - \hat{U} + 2\varepsilon}{\hat{U} - \hat{\mu}_k(\tau_1) - 2c\varepsilon} \right)_+^2 - \sum_{k\neq \hat{i}^\star} \left(\frac{\hat{\mu}_k(\tau_1) - \hat{U} + 2\varepsilon}{\hat{U} - \hat{\mu}_k(\tau_1)} \right)_+^2 \quad \text{(by the definition of } r_{\hat{i}^\star}) \\
&\geq 1. \qquad (97)
\end{aligned}$$

Therefore, it remains to evaluate the number of samples until $U(t)$ decreases to the point where (95) is satisfied.

If arm $i \neq \hat{i}^\star$ is pulled in round t, we must have

$$\begin{aligned}
&\hat{\mu}_i(t-1) + \sqrt{\frac{\beta(t-1,\delta)}{N_i(t-1)}} \\
&= U(t-1) \\
&> U(\hat{\boldsymbol{\mu}}(\tau_1)) - \max\left\{ 2c\varepsilon, \frac{c}{\sqrt{r_{\hat{i}^\star}}} \right\} - \varepsilon_c \quad \text{(since not terminated in round } t-1\text{)}\,. \qquad (98)
\end{aligned}$$

Hence, we must have

$$\begin{aligned}
N_i(t-1) &< \frac{\beta(t-1,\delta)}{\left(U(\hat{\boldsymbol{\mu}}(\tau_1)) - \hat{\mu}_i(t-1) - \max\left\{ 2c\varepsilon, \frac{c}{\sqrt{r_{\hat{i}^\star}}} \right\} - \varepsilon_c \right)^2} \\
&\leq \frac{\beta(t-1,\delta)}{\left(U(\hat{\boldsymbol{\mu}}(\tau_1)) - \hat{\mu}_i(t-1) - 2c\varepsilon - \varepsilon_c \right)^2} \quad \text{(by Lemma 2(v))} \\
&\leq \frac{\beta(t-1,\delta)}{\left(U(\hat{\boldsymbol{\mu}}(\tau_1)) - \mu_i - c\sqrt{\frac{\beta(t-1,\delta)}{N_i(t-1)}} - 2c\varepsilon - \varepsilon_c \right)^2} \quad \text{(under } \mathcal{A}(c))\,. \qquad (99)
\end{aligned}$$

By solving the previous inequality for $N_i(t-1)$, we obtain

$$\begin{aligned}
N_i(t-1) &\leq \frac{(1+c)^2}{\left(U(\hat{\boldsymbol{\mu}}(\tau_1)) - \mu_i - 2c\varepsilon - \varepsilon_c \right)^2} \beta(t-1,\delta) \\
&\leq \frac{(1+c)^2}{\left(U(\hat{\boldsymbol{\mu}}(\tau_1)) - \mu_i - 2c\varepsilon - \varepsilon_c \right)^2} \beta(T_{\max},\delta). \qquad (100)
\end{aligned}$$

Likewise, if arm $\hat{i}^\star$ is pulled in round t, we must have

$$\begin{aligned}
N_{\hat{i}^\star}(t-1) &\leq \frac{(1+c)^2}{\left(U(\hat{\boldsymbol{\mu}}(\tau_1)) - \mu_{\hat{i}^\star} - 2c\varepsilon - \varepsilon_c \right)^2} \beta(t-1,\delta) \vee r_{\hat{i}^\star} \beta(t-1,\delta) \\
&\leq \left(\frac{(1+c)^2}{\left(U(\hat{\boldsymbol{\mu}}(\tau_1)) - \mu_{\hat{i}^\star} - 2c\varepsilon - \varepsilon_c \right)^2} \vee r_{\hat{i}^\star} \right) \beta(T_{\max},\delta). \qquad (101)
\end{aligned}$$

Hence, for any $t \leq \tau$, we have

$$N_i(t) \leq \begin{cases} \frac{(1+c)^2}{(U(\hat{\boldsymbol{\mu}}(\tau_1)) - \mu_i - 2c\varepsilon - \varepsilon_c)^2} \beta(T_{\max},\delta) + 1 & (i \neq \hat{i}^\star) \\ \left(\frac{(1+c)^2}{\left(U(\hat{\boldsymbol{\mu}}(\tau_1)) - \mu_{\hat{i}^\star} - 2c\varepsilon - \varepsilon_c \right)^2} \vee r_{\hat{i}^\star} \right) \beta(T_{\max},\delta) + 1 & (i = \hat{i}^\star) \end{cases} \qquad (102)$$

Hence, for $\varepsilon_c < \varepsilon/2$ and $2c\varepsilon + \varepsilon_c < \varepsilon$, we can use Lemma 10 to establish

$$\begin{aligned}\tau &= \sum_{i\in[K]} N_i(\tau)\\ &\le (1+c)^2\left(f(\boldsymbol{\mu}) + \frac{4K\varepsilon_c}{\varepsilon(\varepsilon-\varepsilon_c)} + \frac{2K(2c\varepsilon+\varepsilon_c)}{\varepsilon(\varepsilon-2c\varepsilon-\varepsilon_c)}\right)\beta(T_{\max},\delta) + K. \end{aligned} \tag{103}$$

□

A.4.4 Probability of $\mathcal{A}(c)$

Lemma 14. *For any $c \in (0,1)$, we have*

$$\Pr(\neg\mathcal{A}(c)) \le \min\left\{1, \delta^{c^2} T_{\max}^{K/2} \exp\left(\frac{K\lambda S^2}{2R^2}\right)\right\}. \tag{104}$$

Proof. Lemma 3 implies

$$\Pr\left(\sum_{i\in[K]} N_i(t)(\hat{\mu}_i(t)-\mu_i)^2 \le \beta(t,\delta), \forall t \in \{2,3,T_{\max}\}\right) \ge 1-\delta \tag{105}$$

Hence, $\mathcal{A}(c)$ holds with probability at least $1-\delta'$ if δ' satisfies

$$c^2\beta(t,\delta) \ge 2R^2\log\frac{1}{\delta'} + \alpha(t), \forall t \in \{2,3,\ldots,T_{\max}\} \tag{106}$$

By solving this for δ', we obtain

$$\delta' \ge \delta^{c^2}\exp\left(\frac{1-c^2}{2R^2}\alpha(t)\right), \forall t \in \{2,3,\ldots,T_{\max}\}, \tag{107}$$

which is satisfied by choosing

$$\delta' = \delta^{c^2}\exp\left(\frac{1-c^2}{2R^2}\alpha(T_{\max})\right). \tag{108}$$

We thus obtain

$$\begin{aligned}\Pr(\neg\mathcal{A}(c)) &\le \delta'\\ &= \delta^{c^2}\exp\left(\frac{1-c^2}{2R^2}(KR^2\log T_{\max} + K\lambda S^2)\right)\\ &\le \delta^{c^2}\exp\left(\frac{1}{2R^2}(KR^2\log T_{\max} + K\lambda S^2)\right)\\ &= \delta^{c^2} T_{\max}^{K/2}\exp\left(\frac{K\lambda S^2}{2R^2}\right). \end{aligned}\tag{109}$$

□

A.4.5 Proof of the upper bound theorem

Proof of Lemma 5. Note that the upper bound on $T_{\max}$ is from Lemma 11, the upper bound on $\beta(T_{\max},\delta)$ is from Lemma 9, $\alpha(T_{\max})$ is given in the proof of Lemma 9, and c is in the range specified in Lemma 12 and satisfies the condition, $2c\varepsilon + \varepsilon_c < \varepsilon$, in Lemma 13:

$$\begin{aligned} c &= \frac{\varepsilon_c}{\Delta_{\max} + \varepsilon'' + \varepsilon}\\ &\le \frac{\varepsilon_c}{\Delta_{\max} + \varepsilon'' + \varepsilon_c}\\ 2c\varepsilon + \varepsilon_c &= \frac{\Delta_{\max} + \left(5 + \frac{\alpha(T_{\max})}{R^2\log(1/\delta)}\right)\varepsilon}{\Delta_{\max} + \left(3 + \frac{\alpha(T_{\max})}{R^2\log(1/\delta)}\right)\varepsilon}\varepsilon_c\\ &< \frac{5}{3}\varepsilon_c && (110)\\ &< \varepsilon && (111)\end{aligned}$$

since $\varepsilon_c \le \frac{1}{2}\varepsilon$. Notice also that (110) gives $c < \frac{\varepsilon_c}{3\varepsilon}$.

Lemma 11 shows that $\tau \le T_{\max}$ almost surely under $\log(1/\delta) \ge K$, and Lemma 13 shows that, under $\mathcal{A}(c)$ with the given conditions on c and ε_c, we almost surely have

$$
\begin{aligned}
\tau &< (1+c)^2 \left(f(\boldsymbol{\mu}) + \frac{4K\varepsilon_c}{\varepsilon(\varepsilon - \varepsilon_c)} + \frac{2K(2c\varepsilon + \varepsilon_c)}{\varepsilon(\varepsilon - 2c\varepsilon - \varepsilon_c)} \right) \beta(T_{\max}, \delta) + K \\
&= (1 + c(2+c)) f(\boldsymbol{\mu}) \beta(T_{\max}, \delta) + (1+c)^2 \left(\frac{4K\varepsilon_c}{\varepsilon(\varepsilon - \varepsilon_c)} + \frac{2K(2c\varepsilon + \varepsilon_c)}{\varepsilon(\varepsilon - 2c\varepsilon - \varepsilon_c)} \right) \beta(T_{\max}, \delta) + K \\
&< f(\boldsymbol{\mu}) \beta(T_{\max}, \delta) + \frac{\varepsilon_c}{\varepsilon} \left(\frac{2+c}{3} f(\boldsymbol{\mu}) + \frac{4(1+c)^2 K}{\varepsilon - \varepsilon_c} + \frac{2(1+c)^2 K (2c\varepsilon + \varepsilon_c)}{\varepsilon_c(\varepsilon - 2c\varepsilon - \varepsilon_c)} \right) \beta(T_{\max}, \delta) + K \\
&\quad \left(\text{by } c < \tfrac{\varepsilon_c}{3\varepsilon}\right) \\
&\le f(\boldsymbol{\mu}) \beta(T_{\max}, \delta) + \frac{\varepsilon_c}{\varepsilon} \left(\frac{2+c}{3} f(\boldsymbol{\mu}) + \frac{4(1+c)^2 K}{\varepsilon - \varepsilon_c} + \frac{\frac{10}{3}(1+c)^2 K \varepsilon_c}{\varepsilon_c(\varepsilon - \frac{5}{3}\varepsilon_c)} \right) \beta(T_{\max}, \delta) + K \\
&\quad \left(\text{by } c < \tfrac{\varepsilon_c}{3\varepsilon}\right) \\
&\le f(\boldsymbol{\mu}) \beta(T_{\max}, \delta) + \frac{\varepsilon_c}{\varepsilon} \left(\frac{2+c}{3} f(\boldsymbol{\mu}) + \frac{8(1+c)^2 K}{\varepsilon} + \frac{20(1+c)^2 K}{\varepsilon} \right) \beta(T_{\max}, \delta) + K \\
&\quad \left(\text{by } \varepsilon_c \le \tfrac{1}{2}\varepsilon\right) \\
&= f(\boldsymbol{\mu}) \beta(T_{\max}, \delta) + \frac{\varepsilon_c}{\varepsilon} \zeta \beta(T_{\max}, \delta) + K. \qquad (112)
\end{aligned}
$$

Hence, we have

$$
\begin{aligned}
\mathbb{E}[\tau] &= \Pr(\mathcal{A}(c)) \mathbb{E}[\tau \mid \mathcal{A}(c)] + \Pr(\neg\mathcal{A}(c)) \mathbb{E}[\tau \mid \neg\mathcal{A}(c)] \\
&\le \mathbb{E}[\tau \mid \mathcal{A}(c)] + \Pr(\neg\mathcal{A}(c)) T_{\max} \\
&\le f(\boldsymbol{\mu}) \beta(T_{\max}, \delta) + \frac{\varepsilon_c}{\varepsilon} \zeta \beta(T_{\max}, \delta) + K + \min\left\{ \delta^{c^2} T_{\max}^{K/2} \exp\left(\frac{K\lambda S^2}{2R^2} \right) \right\} T_{\max} \qquad (113)
\end{aligned}
$$

which implies (18) by the bound on $\Pr(\neg\mathcal{A}(c))$ from Lemma 14.

Since the sample complexity has a lower bound of $2R^2 f(\boldsymbol{\mu}) \log(1/\delta) + o(\log(1/\delta))$ by Theorem 1 and an upper bound of $T_{\max} = O(KR^2 \log(1/\delta)/\varepsilon^2)$ by Lemma 11, we have $f(\boldsymbol{\mu}) = O(K/\varepsilon^2)$. Hence, we must have $\zeta = O(K/\varepsilon^2)$. □

Proof of Theorem 6. Since $\Delta_{\max} \ge 0$ and $\alpha(T_{\max}) = O(\log\log(1/\delta))$, we have

$$
c \le \frac{1}{3 + \frac{\alpha(T_{\max})}{R^2 \log(1/\delta)}} \frac{\varepsilon_c}{\varepsilon} = O\left((\log(1/\delta))^{-1/3} \right). \qquad (114)
$$

Now, notice that

$$
\begin{aligned}
\log \delta^{c^2} &= c^2 \log \delta \\
&= -\left(\log(1/\delta)\right)^{1/3}. \qquad (115)
\end{aligned}
$$

Hence, we have

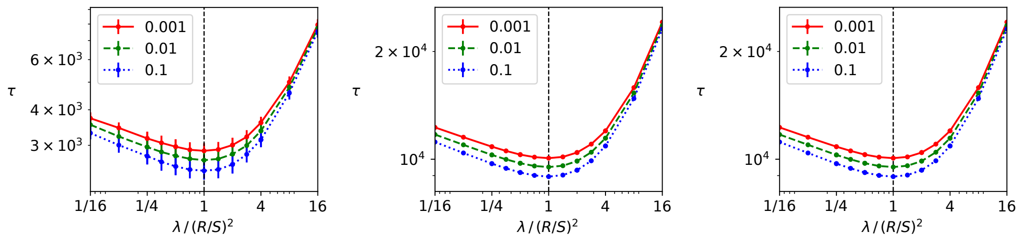
$$
\begin{aligned}
\delta^{c^2} T_{\max}^{1+K/2} &= O\left(e^{-(\log(1/\delta))^{1/3}} (\log(1/\delta))^{1+K/2} \right) \\
&= o(1). \qquad (116)
\end{aligned}
$$

These together with Lemma 5 imply (20). □

B Details of experiments

B.1 Implementation

In this section, we discuss two implementation choices that can enhance the efficiency of EllipsoidEst. These techniques are broadly applicable and are recommended when appropriate.



(a) Uniform $\boldsymbol{\mu}$

(b) Clustered $\boldsymbol{\mu}$

(c) Gaussian $\boldsymbol{\mu}$

Figure 4: Sample complexity τ against the regularization parameter λ, where we set $K = 10, \varepsilon = 0.1$; δ is as shown in the legend. The results are over 10 random seeds.

First, while there may be room for further refinement, we recommend setting $\lambda = (R/S)^2$. With this setting, EllipsoidEst becomes effectively free of tunable hyperparameters. This choice can be supported both theoretically and empirically.

Theoretically, note that our confidence ellipsoid (specifically, $\beta(t, \delta)$ in (10)) involves terms of the form

$$\omega(x) \coloneqq R^2 \log(1 + n/x) + xS^2 \frac{n}{n + x}. \tag{117}$$

The derivative of $\omega(x)$ is

$$\frac{\partial \omega(x)}{\partial x}\omega(x) = -\frac{nR^2}{x^2 + nx} + \frac{n^2 S^2}{(n + x)^2} \tag{118}$$

and equals zero at $x = \frac{nR^2}{nS^2 - R^2}$, which tends to $(R/S)^2$ as $n \to \infty$. So, for large n, $x = (R/S)^2$ tends to minimize $\omega(x)$, meaning that the size of the confidence ellipsoid tends to be minimized with $\lambda = (R/S)^2$ as more samples are collected from the arms.

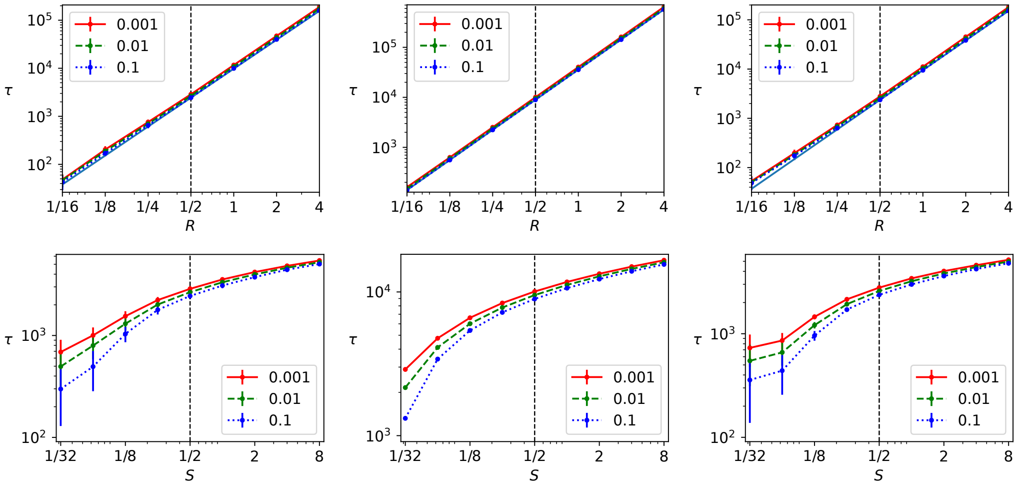
The choice of $\lambda = (R/S)^2$ is also supported empirically, as shown in Figure 4. The sample complexity τ is minimized around $\lambda = (R/S)^2$ across a wide range of settings. In the figure, $\boldsymbol{\mu}$ is configured in three different settings: Uniform, Clustered, and Gaussian. These configurations will be explained in detail in Appendix B.2.

The second implementation choice concerns the selection of S. Recall that S must be chosen such that $|\mu_i| \le S, \forall i \in [K]$ in order to guarantee the PAC property. If we know that $\mu_i \in [\ell, u], \forall i$ for constants ℓ and u, we can set $S = |\ell| \vee |u|$. As demonstrated in Appendix B.3, the sample complexity of EllipsoidEst tends to increase with S (even with the choice of $\lambda = (R/S)^2$). This motivates us to shift the sample by $-(\ell + u)/2$, which makes the true means lie within $[(\ell - u)/2, (u - \ell)/2]$. Consequently, we can set $S = (u - \ell)/2$, which is guaranteed to be no greater than $|\ell| \vee |u|$.

B.2 Experimental settings and baselines

We design our experiments to evaluate the empirical properties of EllipsoidEst and compare its performance against baseline methods, focusing on three aspects. First, EllipsoidEst requires the parameters R and S, which depend on assumptions about the reward distributions. In practice, these values may be misspecified, so it is important to assess the algorithm's sensitivity to such inaccuracies. Second, the midpoint of the interval returned by EllipsoidEst tends to overestimate the best mean (i.e., $\mathbb{E}[\hat{\mu}^\star] > \mu^\star$). It is useful and interesting to quantify the extent of this bias. Third, although EllipsoidEst achieves asymptotically optimal sample complexity as $\delta \to 0$, this does not guarantee strong performance for moderate values of δ. We therefore empirically evaluate its performance under practical confidence levels and compare it to standard baselines.

As baselines, we consider Successive Elimination (SE) [6, 3] and UGapEc [4], both originally designed for best-arm identification and adapted here for best-mean estimation following the approach in [13]. While SE has suboptimal asymptotic sample complexity, it is known to perform well for moderate values of δ, especially when the number of arms is large. UGapEc achieves an



(a) Uniform $\boldsymbol{\mu}$ (b) Clustered $\boldsymbol{\mu}$ (c) Gaussian $\boldsymbol{\mu}$

Figure 5: Sensitivity of the sample complexity τ to the values of R (top row) and S (bottom row), where we set $K = 10, \varepsilon = 0.1$; δ is as shown in the legend. The results are over 10 random seeds.

asymptotically optimal order of sample complexity for best-arm identification and also demonstrates strong empirical performance at practical confidence levels. Specifically, we use the SE variant proposed in [13], which is inspired by the original SE but tailored for best-mean estimation. For UGapEc, we adopt the two-step procedure from [13]: first, identify the best arm using UGapEc, and then sample that arm to estimate its mean with the desired confidence level[2].

In our experiments, we typically assume that each arm's reward follows a Bernoulli distribution. This choice aligns with the baselines (SE and UGapEc), which are originally designed for bounded reward distributions. To assess the robustness of our findings, we also examine settings with Gaussian reward distributions in Appendix B.6. We consider three configurations for the mean vector $\boldsymbol{\mu}$: Uniform, Clustered, and Gaussian. In the Uniform configuration, the means are evenly spaced in $[0, 1]$, with $\mu_i = i/(K + 1), \forall i \in [K]$. In the Clustered configuration, all arms share the same mean: $\mu_i = 1/2, \forall i \in [K]$. In the Gaussian configuration, the means follow a Gaussian-shaped distribution centered at 0.5. Specifically, we compute the percent point function (inverse CDF) of a standard normal distribution at the $100(i - 0.5)/K$ percentile, then linearly scale the values so that $\mu_1 = 0.1$ and $\mu_K = 0.9$.

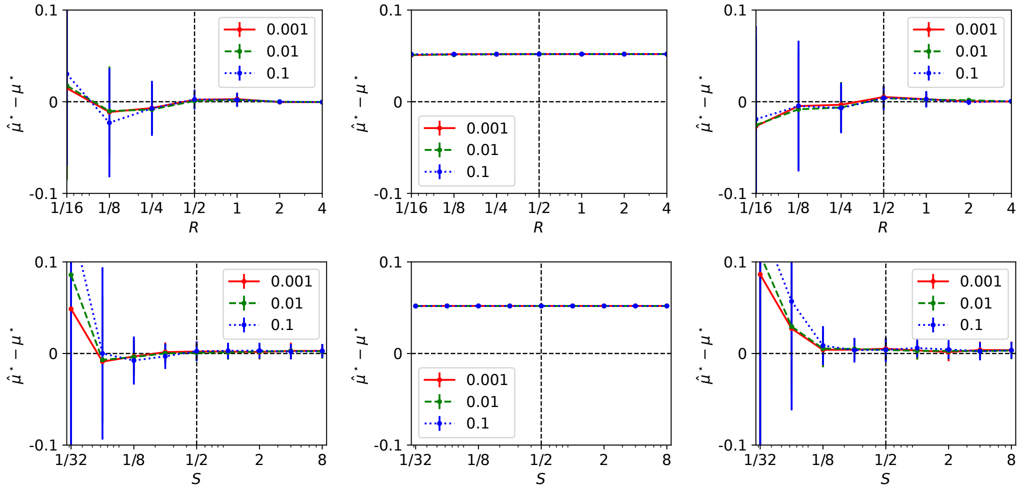
We assume that the user of EllipsoidEst knows that the rewards follow a Bernoulli distribution. This allows the user to set $R = S = 1/2$, while shifting the sample by -0.5 as we have discussed in Appendix B.1. In all experiments, we fix $\varepsilon = 0.1$ and vary δ and K to evaluate the performance of EllipsoidEst under different confidence and problem-size settings.

B.3 Sensitivity to R and S

Figure 5 examines the sensitivity of the sample complexity τ of EllipsoidEst to the values of R and S. In the top row, we fix $S = 1/2$ and vary R, while in the bottom row, we fix $R = 1/2$ and vary S.

The sensitivity of sample complexity to R is consistent with our asymptotic analysis, which shows that $\mathbb{E}[\tau]$ scales as $2R^2 f(\boldsymbol{\mu}) \log(1/\delta)$, implying approximately quadratic growth with respect to R. In contrast, the influence of S is less direct but still significant, as it affects the size of the confidence ellipsoid through the term $\beta(t, \delta)$ in (10). As shown in the figure, sample complexity increases with larger values of S, which supports the sample-shifting technique described in Appendix B.1. Overall, while the impact of S is weaker than that of R, it is still non-negligible, underscoring the importance of setting both parameters carefully.

[2] The proof of Lemma 4 in [13] does not hold for the case when the samples from best-arm identification is reused for best-mean estimation; hence, we set $M_{\hat{I}} = 0$ in Algorithm 2 in [13].



(a) Uniform $\boldsymbol{\mu}$ (b) Clustered $\boldsymbol{\mu}$ (c) Gaussian $\boldsymbol{\mu}$

Figure 6: Sensitivity of the estimated best-mean $\hat{\mu}^\star$ to the values of R (top row) and S (bottom row), where we set $K = 10, \varepsilon = 0.1$; δ is as shown in the legend. The results are over 10 random seeds.

When R and S are set too high, the sample complexity increases unnecessarily; when set too low, they can lead to significant errors in the estimated best-mean $\hat{\mu}^\star$. This trade-off is illustrated in Figure 6, particularly in the Uniform and Gaussian configurations, where $\hat{\mu}^\star$ exhibits significant deviation from the true best-mean $\mu^\star$ when R or S is set too low. For the Clustered configuration, the best mean is consistently overestimated, which we discuss in detail in Section 7.3.

B.4 Comparison against baselines

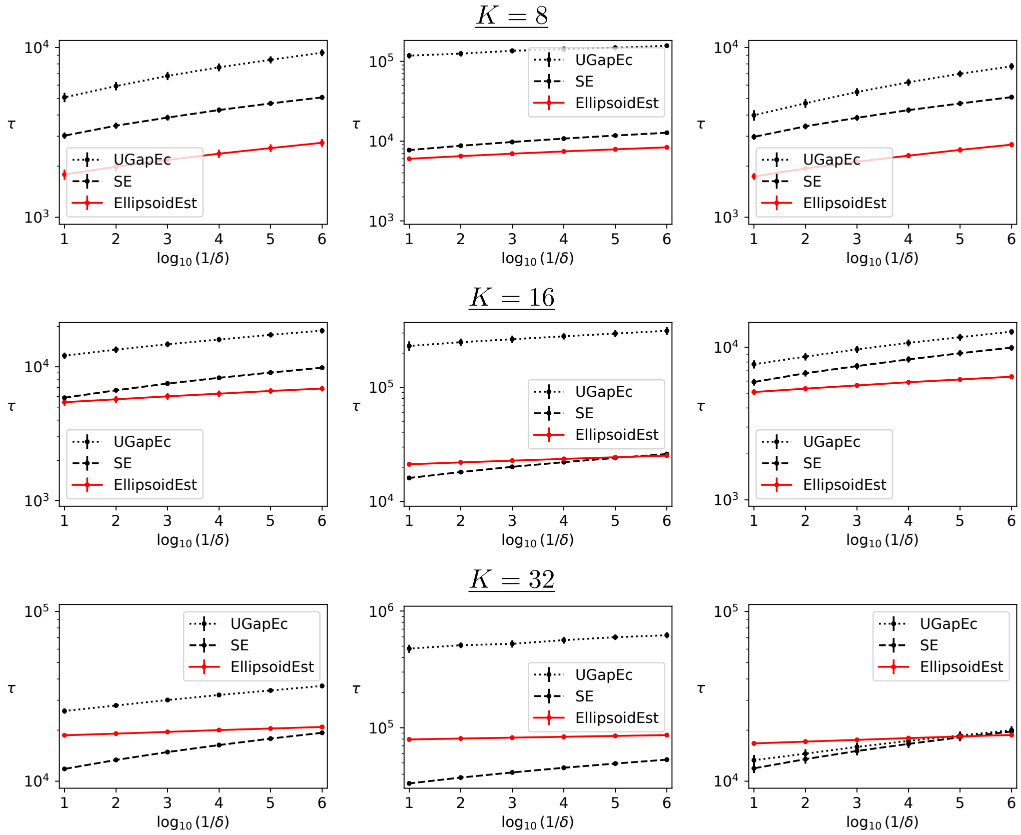
Figure 7 compares the sample complexity of EllipsoidEst against two baseline methods—Successive Elimination (SE) and UGapEc—both adapted for best-mean estimation. All three methods satisfy the same PAC guarantee: the best mean is estimated within ε error with probability at least $1 - \delta$. The results show that EllipsoidEst consistently outperforms the baselines when the number of arms is small to moderate (up to around $K = 16$). Moreover, as predicted by theory, EllipsoidEst exhibits the slowest growth in sample complexity as $\log(1/\delta)$ (i.e., as $\delta \to 0$). UGapEc performs poorly when K is small or when the true means are tightly clustered, while SE shows a faster increase in sample complexity (as $\delta \to 0$) across settings. Overall, although each method has its own strengths and limitations, EllipsoidEst demonstrates robust and competitive performance in configurations of practical interest.

Figure 8 highlights a limitation of EllipsoidEst by showing how its sample complexity τ scales with the number of arms K. As the figure illustrates, the sample complexity of EllipsoidEst increases more rapidly than that of the baseline methods as K grows. This observation motivates further research on best-mean estimation algorithms that scale more gracefully with the number of arms.

Figure 9 confirms the correctness of the algorithms in returning an estimate that is within ε of the true best-mean with probability at least $1 - \delta$. Here, we set $\varepsilon = 0.1$ and $\delta = 0.01$ in the settings of Figure 7. The box-and-whisker plots indicate that, *in every case*, the returned estimates lie within $\varepsilon = 0.1$ of the true best-mean across all 30 random seeds. In Appendix B.5, we discuss the bias in the estimated best-means.

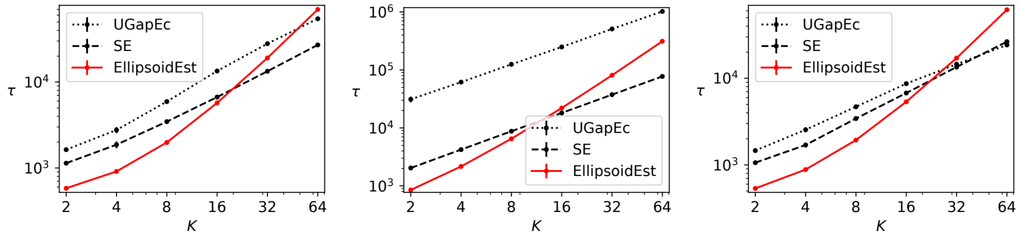
B.5 Bias in the midpoint of the interval

As discussed in Section 7.3, EllipsoidEst returns the output $\hat{\mu}^\star$ that typically exceeds the maximum of the empirical means, $\max_i \hat{\mu}_i(\tau)$, which itself tends to overestimate the true best-mean $\mu^\star$ due to the winner's curse. Figure 10 explores this bias in the output—the midpoint of the interval that contains the true best-mean with high probability— returned by EllipsoidEst, using the same settings as Figure 8. The red curve shows the bias in the midpoint (i.e., $\hat{\mu}^\star - \mu^\star$). As expected, the midpoint



(a) Uniform $\boldsymbol{\mu}$ (b) Clustered $\boldsymbol{\mu}$ (c) Gaussian $\boldsymbol{\mu}$

Figure 7: Sample complexity τ of EllipsoidEst compared against baselines, Successive Elimination (SE) and UGapEc, where we vary δ, setting $\varepsilon = 0.1$ and K as indicated in above the panels. The results are based on 30 random seeds.

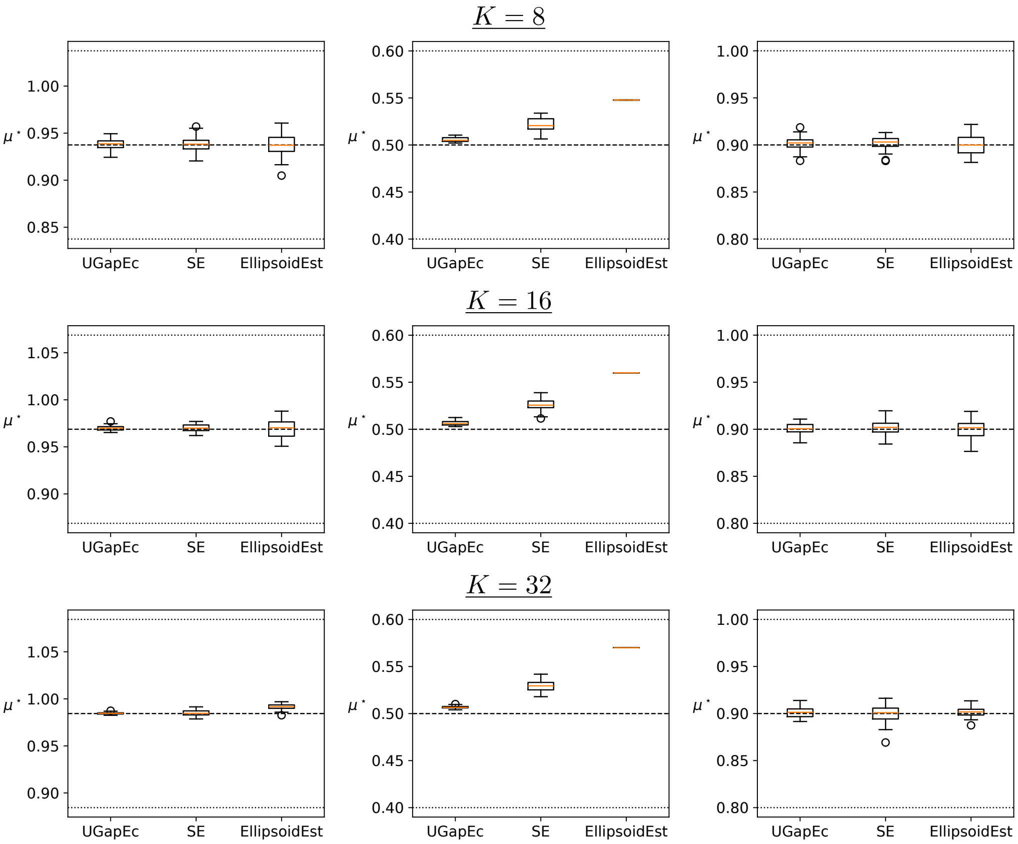


(a) Uniform $\boldsymbol{\mu}$ (b) Clustered $\boldsymbol{\mu}$ (c) Gaussian $\boldsymbol{\mu}$

Figure 8: Sample complexity τ of EllipsoidEst compared against baselines, Successive Elimination (SE) and UGapEc, where we vary K, while setting $\delta = 0.01, \varepsilon = 0.1$. The results are based on 30 random seeds.

returned by EllipsoidEst consistently overestimates the best mean, with the bias being particularly pronounced when the number of arms K is large or when the true means are tightly clustered—scenarios where many arms are close to optimal. Note that despite the bias, the midpoint (output) $\hat{\mu}^\star$ still satisfies the PAC guarantee with $\varepsilon = 0.1$ in this experiment: the bias remains below ε, in line with the theoretical guarantee.

The blue curve in Figure 10 illustrates the bias in the maximum of the empirical means (i.e., $\max_i \hat{\mu}_i(\tau) - \mu^\star$), based on the samples collected by EllipsoidEst. As expected, this value—already stored in the memory of EllipsoidEst—is notably less biased than the output $\hat{\mu}^\star$.

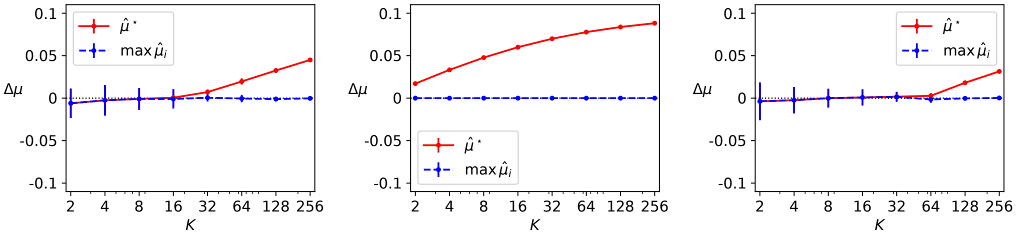


(a) Uniform $\boldsymbol{\mu}$

(b) Clustered $\boldsymbol{\mu}$

(c) Gaussian $\boldsymbol{\mu}$

Figure 9: The best means estimated by UGapEC, SE, and EllipsoidEst in the settings of Figure 7 with $\varepsilon = 0.1$ and $\delta = 0.01$.



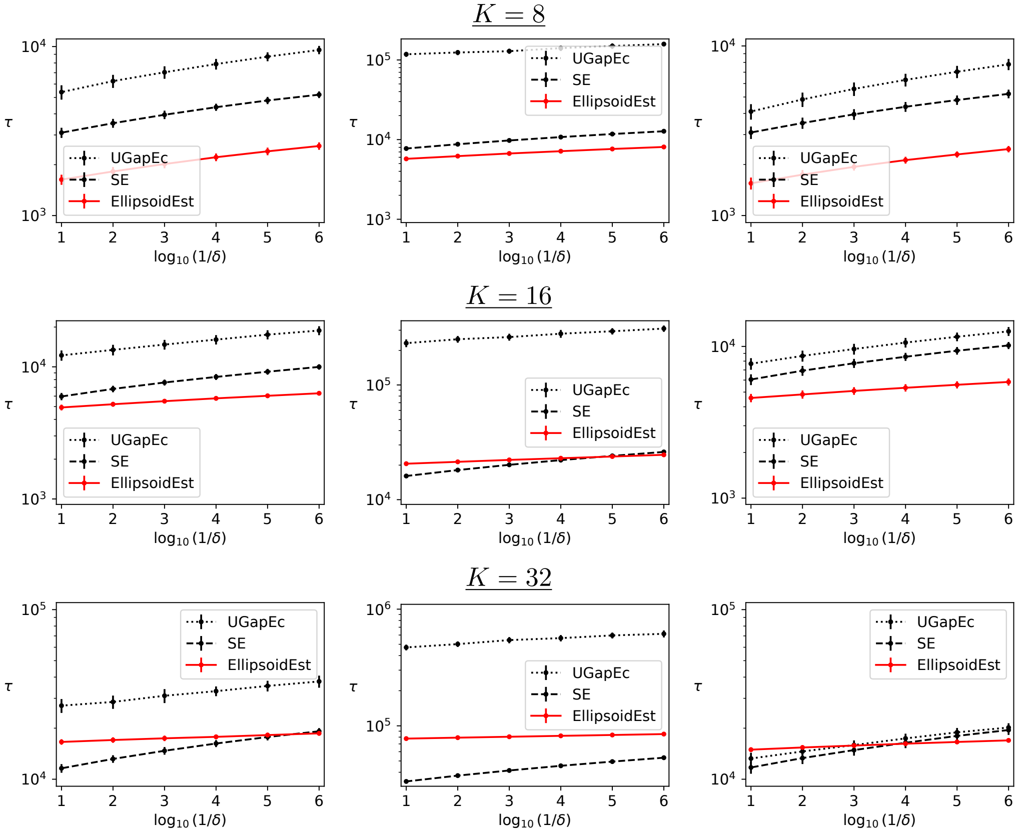
(a) Uniform $\boldsymbol{\mu}$

(b) Clustered $\boldsymbol{\mu}$

(c) Gaussian $\boldsymbol{\mu}$

Figure 10: Bias in the midpoint of the confidence interval returned by EllipsoidEst, where we vary K, while setting $\delta = 0.01, \varepsilon = 0.1$. The results are based on 30 random seeds.

Also, recall from Section 1 that, although we have framed the problem as best-mean estimation following the multi-armed bandit literature, our original motivation was to estimate the mean performance *in worst scenarios*. In such settings, the overestimation can be viewed as a form of conservatism, which is typically more appropriate than optimistic estimates in safety-critical applications. In fact, an unbiased estimate may unintentionally underestimate risks and lead to overly optimistic conclusions from the perspective of confidence guarantees, which can be undesirable from the standpoint of safety assurance.



(a) Uniform $\boldsymbol{\mu}$ (b) Clustered $\boldsymbol{\mu}$ (c) Gaussian $\boldsymbol{\mu}$

Figure 11: [Gaussian reward distributions] Sample complexity τ of EllipsoidEst compared against baselines, Successive Elimination (SE) and UGapEc, where we vary δ, setting $\varepsilon = 0.1$ and K as indicated in above the panels. The results are based on 30 random seeds.

B.6 Gaussian reward distributions

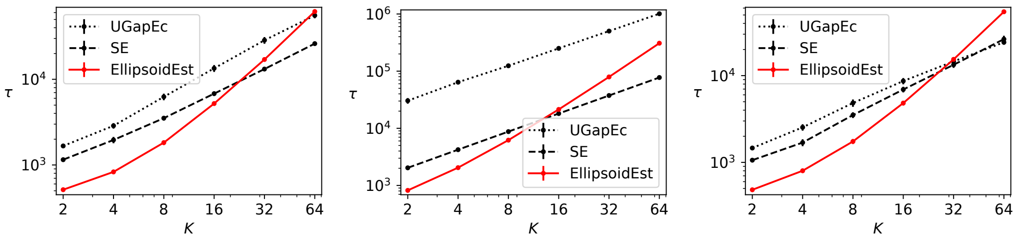
In this section, we assume that each arm's reward follows a Gaussian distribution. The standard deviation of the Gaussian distribution is fixed at $R = 1/2$. Similar to the previous experiments, we consider three configurations for the mean vector $\boldsymbol{\mu}$: Uniform, Clustered, and Gaussian. Hence, the means are in $[0, 1]$, which we assume to be known and set $S = 1/2$ (while shifting the sample by -0.5).

While EllipsoidEst is guaranteed to estimate the best mean within ε error with probability at least $1 - \delta$ in this setting, the PAC guarantees of the baselines—SE and UGapEc—do not formally extend here, as they were originally designed for bounded reward distributions. Nevertheless, we apply them as if each arm's reward follows a Bernoulli distribution, in order to assess the robustness of our findings about EllipsoidEst across different reward models.

Figures 11–12 present results analogous to those in Figures 7–8, but for the case of Gaussian reward distributions. The similarity between the two sets of results suggests that our findings based on Bernoulli rewards are robust to changes in the underlying distribution.

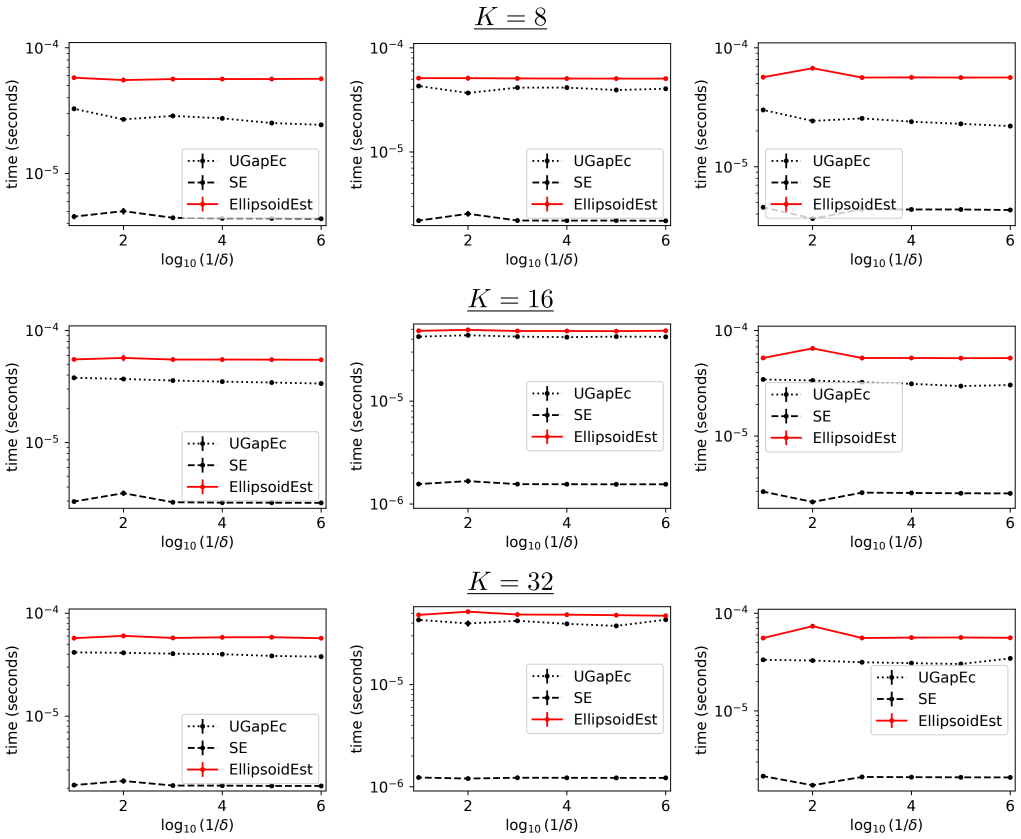
B.7 Computational requirements

Although running time is not the primary concern in best-mean estimation, it is still important that algorithms do not require excessive computation. Here, we compare the running time of EllipsoidEst to that of the baselines. All experiments in Appendix B, including those reported previously, were conducted on a single CPU core with 4 GB of memory and no GPU acceleration, in a cloud environment.



(a) Uniform $\boldsymbol{\mu}$ (b) Clustered $\boldsymbol{\mu}$ (c) Gaussian $\boldsymbol{\mu}$

Figure 12: [Gaussian reward distributions] Sample complexity τ of EllipsoidEst compared against baselines, Successive Elimination (SE) and UGapEc, where we vary K, while setting $\delta = 0.01, \varepsilon = 0.1$. The results are based on 30 random seeds.

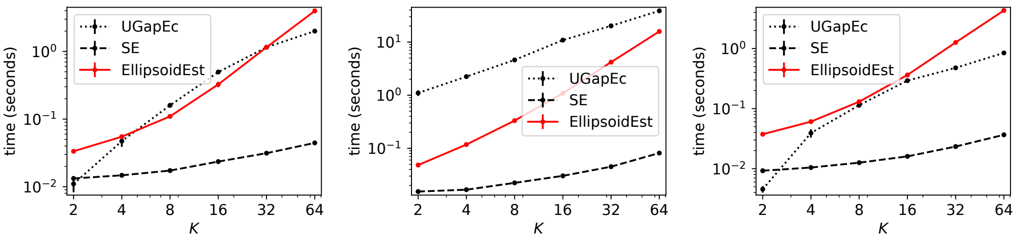


(a) Uniform $\boldsymbol{\mu}$ (b) Clustered $\boldsymbol{\mu}$ (c) Gaussian $\boldsymbol{\mu}$

Figure 13: Running time of EllipsoidEst compared against baselines in the settings of Figure 7

Figures 13–14 report the running times of EllipsoidEst and the baselines in the same settings as Figures 7–8. The reported time is measured until each algorithm returns an output, and it includes the time to generate samples. As a result, the number of samples an algorithm requires is the primary factor affecting its running time.

While the running time of each algorithm can vary significantly with implementation—and our implementations are not fully optimized—, we can already conclude from these results that all algorithms are sufficiently efficient from computational perspectives. In particular, each algorithm requires well under one millisecond per sample.



(a) Uniform $\boldsymbol{\mu}$

(b) Clustered $\boldsymbol{\mu}$

(c) Gaussian $\boldsymbol{\mu}$

Figure 14: Running time of EllipsoidEst compared against baselines in the settings of Figure 8

B.8 Properties of the optimal solution of (P1)

Recall that the difficulty of best mean estimation stems from the non-convexity of the optimization problem (P1). While the prior work on best arm identification has addressed such non-convexity with convex reformulation [12, 21], (P1) does not admit such a convex reformulation, and the lower bound remains inherently nonconvex. This leads to challenges such as possible discontinuities in the optimal allocation and the lack of structure typically exploited in classical settings. In this section, we provide a numerical example where the solution of (P1) has multiple disconnected local minima.

Specifically, Figure 1 shows the optimal value $f(U; \boldsymbol{\mu})$ of (P2) as a function of U for the setting where $\varepsilon = 0.1$, $K = 512$, $\mu_1 = 1$, and $\mu_i = 0.971525$ for $i \neq 1$. In this case, the objective $f(U; \boldsymbol{\mu})$ has two local minima at $U \approx 1.1633$ and $U \approx 1.1745$. For μ_i around the above values, the optimal solution discontinuously changes between these U's.

This phenomenon simply arises from the fact that the feasible region is nonconvex. The Hessian of the left-hand side (LHS) of (3) has eigenvalues $r_i - \sqrt{r_i^2 + 4(U - \mu_i)^2} \leq 0$ and $r_i + \sqrt{r_i^2 + 4(U - \mu_i)^2} \geq 0$ with respect to (r_i, U). This means that the LHS of (3) is not negative-semidefinite and thus the feasible region is nonconvex.

Although nonconvex optimization also appears in the lower-bound analysis of classical best arm identification, an important distinction lies in the availability of a convex reformulation.

In classical best arm identification, the lower bound (e.g., Proposition 16 in [12] and (1) in [21]) takes the form:

$$\max_{w \in \Sigma} \inf_{\lambda \in \mathrm{Alt}(\mu)} \sum_{a \in [K]} w_a d(\mu_a, \lambda_a), \tag{119}$$

which is a nonconvex problem if we first solve the inner minimization. However, this can be equivalently rewritten as a linear program with infinitely many constraints:

$$\max_{x \in \mathbb{R}, w \in \Sigma_K} x \text{ s.t. } x - \sum_{a \in [K]} w_a d(\mu_a, \lambda_a) \leq 0, \forall \lambda \in \mathrm{Alt}(\boldsymbol{\mu}), \tag{120}$$

where the objective and the constraints are linear (and hence convex) in (x, w). While the constraint set indexed by $\lambda \in \mathrm{Alt}(\boldsymbol{\mu})$ is nonconvex, the resulting optimization problem enjoys desirable properties such as convexity (and often uniqueness) of the optimal solution set, and continuity of the optimal allocation with respect to $\boldsymbol{\mu}$. These properties are crucial for algorithms like Track-and-Stop, which rely on the property that the allocation gradually converges to the optimal one.

In contrast, our problem does not admit such a convex reformulation, and the lower bound remains inherently nonconvex, leading to the discontinuity in the optimal allocation as shown with Figure 1.

B.9 Comparison to a deviation inequality for exponential families

Here, we consider a stopping rule based on a deviation inequality for exponential families proposed in [12] for the case of Gaussian distributions. Specifically, Theorem 9 in [12] suggests the stopping rule where the $\beta(t, \delta)$ in Step 10 of Algorithm 1 is replaced with the $\beta_{\mathrm{ef}}(t, \delta)$ defined as follows:

$$\beta_{\mathrm{ef}}(t, \delta) = 2R^2 K C_G \left(\frac{\log(1/\delta)}{K} \right) + 4R^2 \sum_{i \in [K]} \log\left(4 + \log(N_i(t))\right), \tag{121}$$

where

$$C_G(x) = \min_{\xi \in (1/2, 1)} \frac{g(\xi) + x}{\xi} \tag{122}$$

$$g(\xi) = 2\xi - 2\xi \log(4\xi) + \log \zeta(2\xi) - \frac{1}{2} \log(1 - \xi), \tag{123}$$

where ζ is the zeta function. The stopping rule based on β_{ef} guarantees the correctness of the algorithm for Gaussian arms.

Since $\beta(t, \delta) = O(\log N_i(t))$ and $\beta_{\mathrm{ef}}(t, \delta) = O(\log \log N_i(t))$, the algorithm terminates earlier with the stopping rule based on β_{ef} than with the one based on $\beta(t, \delta)$ in the limit of $N_i(t) \to \infty$. However, here we show $\beta(t, \delta) < \beta_{\mathrm{ef}}(t, \delta)$ for moderate values of $N_i(t)$ in typical cases of practical interest.

We first derive a lower bound of $\beta_{\mathrm{ef}}(t, \delta)$. Observe that

$$C_G(x) \geq \min_{\xi \in (1/2, 1)} \frac{g(\xi)}{\xi} + \min_{\xi \in (1/2, 1)} \frac{x}{\xi} \tag{124}$$

$$\geq \eta + x. \tag{125}$$

where $\eta > 1.4169$ can be verified numerically (see Figure 15a). Hence, we have

$$\beta_{\mathrm{ef}}(t, \delta) \geq \underline{\beta}_{\mathrm{ef}}(t, \delta) := 2\eta R^2 K + 2R^2 \log(1/\delta) + 4R^2 \sum_{i \in [K]} \log(4 + \log(N_i(t))). \tag{126}$$

We also derive an upper bound of $\beta(t, \delta)$:

$$\beta(t, \delta) \leq \overline{\beta}(t, \delta) := 2R^2 \log(1/\delta) + \lambda S^2 K + R^2 \sum_{i \in [K]} \log \frac{\lambda + N_i(t)}{\lambda}. \tag{127}$$

Now, consider the case where $R = S = 1$ and $n = N_i(t), \forall i \in [K]$. We also (optimally) set $\lambda = (R/S)^2$ in $\beta(t, \delta)$. Then we have

$$\overline{\beta}(t, \delta) = 2R^2 \log(1/\delta) + R^2 K \left(\log(n + 1) + 1\right) \tag{128}$$

$$\underline{\beta}_{\mathrm{ef}}(t, \delta) = 2R^2 \log(1/\delta) + R^2 K (4 \log(4 + \log(n)) + 2\eta). \tag{129}$$

It can be verified numerically (see Figure 15b) that

$$\log(n + 1) + 1 < 4 \log(4 + \log(n)) + 2\eta, \forall n \leq 549906 \tag{130}$$

In Figure 15c, we compare the values of $\beta(t, \delta)$ and $\beta_{\mathrm{ef}}(t, \delta)$. Although $\beta_{\mathrm{ef}}(t, \delta) = O(\log \log N_i(t))$ can get smaller than $\beta(t, \delta) = O(\log N_i(t))$ for large $N_i(t)$, we observe $\beta(t, \delta) < \beta_{\mathrm{ef}}(t, \delta)$ for moderate values of $N_i(t)$ (specifically, $N_i(t) \leq 5 \times 10^5$) that are of practical interest.

B.10 Ablation

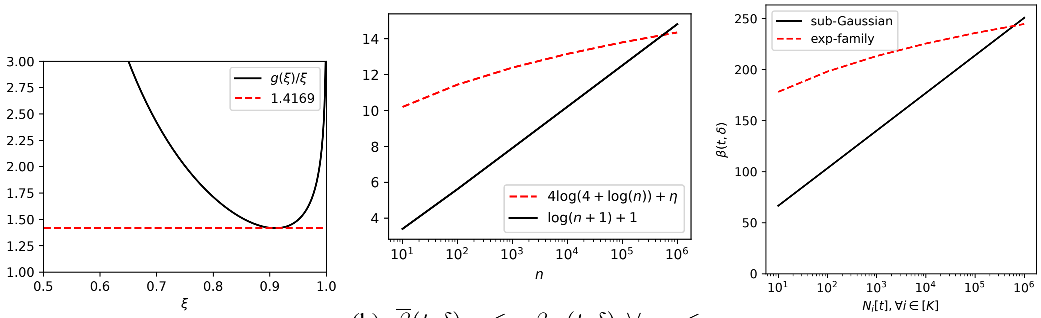
In this section, we conduct the following ablation studies to assess the effectiveness of our stopping rule and sampling strategy against relatively naive choices: (i) We change the stopping condition such that the algorithm terminates when

$$\max_{i \in [K]} L_i \geq U - 2\varepsilon \tag{131}$$

holds. (ii) We fix the sampling strategy to UCB (i.e., the algorithm never enters Phase 2).

Figure 16 shows the sample complexity τ of EllipsoidEst with our proposed stopping rule with the confidence ellipsoid (EllipsoidEst) and the one with the naive stopping rule with (131) (Naive). Observe that the choice of stopping condition can significantly affect performance, especially when $\boldsymbol{\mu}$ is clustered.

On the other hand, while the two-phase sampling strategy is crucial for our theoretical guarantees, we find that its empirical impact is limited in the settings considered in our experiments. Specifically, in the settings of Figure 16, the differences in the stopping times are at most a few percent between the two sampling strategies.

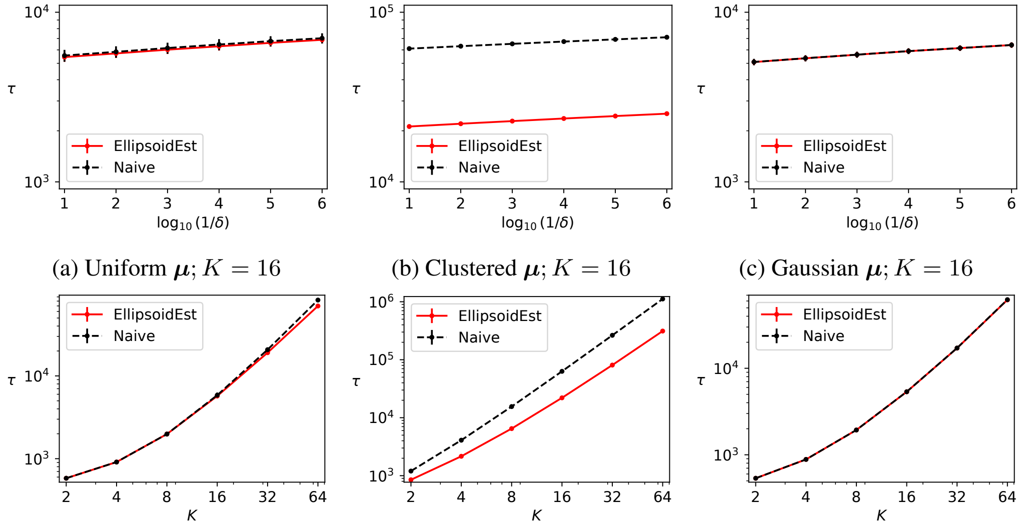


(a) $g(\xi)/\xi \geq \eta > 1.4169$

(b) $\overline{\beta}(t,\delta) \leq \underline{\beta}_{\text{ef}}(t,\delta), \forall n \leq 549906$

(c) $\beta(t,\delta)$

Figure 15: (a)-(b) Numerical verifications of (125) and (130). (c) The value of $\beta(t,\delta)$ in our stopping rule (sub-Gaussian) and $\beta_{\text{ef}}(t,\delta)$ for the bound based on the deviation inequality in [12] (exp-family), where $K = 16$ and $\delta = 0.001$.



(a) Uniform $\boldsymbol{\mu}$; $K = 16$

(b) Clustered $\boldsymbol{\mu}$; $K = 16$

(c) Gaussian $\boldsymbol{\mu}$; $K = 16$

(d) Uniform $\boldsymbol{\mu}$; $\delta = 0.01$

(e) Clustered $\boldsymbol{\mu}$; $\delta = 0.01$

(f) Gaussian $\boldsymbol{\mu}$; $\delta = 0.01$

Figure 16: Ablation study comparing the sample complexity τ of EllipsoidEst with the proposed stopping rule (EllipsoidEst) and the naive stopping rule (Naive), where we vary K and δ as indicated, while setting $\varepsilon = 0.1$. The results are based on 30 random seeds.

C Societal impacts

This work is expected to have both positive and potentially negative societal impacts. On the positive side, our method improves the data efficiency of estimating the mean performance in the best or worst case, which is particularly valuable in domains where data collection is costly or time-consuming. While originally motivated by the need to provide safety guaranties for AI agents, the proposed approach may have broader societal applications in areas such as healthcare, finance, and scientific research—fields where rigorous safety and performance assurances are essential. On the negative side, our method could be misused if the PAC guaranty is misunderstood. In particular, the midpoint of the interval returned by our method should not be applied as a point estimate in use cases that require unbiased estimates.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The contributions and claims are accurately summarized at the end of Section 1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The proposed algorithm has relatively poor performance for a large number of arms, and this is shown and discussed with Figure 3b in Section 7 and with Figure 8 in Appendix B.4.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All the theoretical statements are stated with full set of assumptions, and complete proofs are presented in Section 5 and Appendix A.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All the details of experimental settings are explained in Section 7 or Appendix B, and source code is submitted as the supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We submit the source code as a supplementary material together with README.md, which includes the instructions on how to set up the environment and how to reproduce all of the experimental results reported in the paper. This paper does not use any datasets.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The proposed algorithm has a single hyperparameter λ, and we set its value as $\lambda = (R/S)^2$ based on our analysis. Our experimental settings do not involve train-test split.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: In all figures of experimental results, we plot standard deviations as error bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The computer resources are detailed in Appendix B.7. Also, while the running time of an algorithm is not our primary concern, we report the running time in Appendix B.7.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have carefully read the NeurIPS Code of Ethics and ensured that this paper follows the code.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss both potential positive societal impacts and negative societal impacts in Appendix C.

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not release data or models.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use numpy, scipy, matplotlib, jupyterlab, which are explicitly mentioned in pyproject.toml files. These libraries have BSD-style or MIT licenses.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: README.md provides all of the necessary information to run the source code for reproducing the experimental results.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: We have used LLMs for writing and editing and for implementing standard methods such as bisection.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.