On Elimination Strategies for Bandit Fixed-Confidence Identification

Andrea Tirinzoni*
Meta AI
Paris, France
tirinzoni@fb.com

Rémy Degenne
Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
remy.degenne@inria.fr

Abstract

Elimination algorithms for bandit identification, which prune the plausible correct answers sequentially until only one remains, are computationally convenient since they reduce the problem size over time. However, existing elimination strategies are often not fully adaptive (they update their sampling rule infrequently) and are not easy to extend to combinatorial settings, where the set of answers is exponentially large in the problem dimension. On the other hand, most existing fully-adaptive strategies to tackle general identification problems are computationally demanding since they repeatedly test the correctness of every answer, without ever reducing the problem size. We show that adaptive methods can be modified to use elimination in both their stopping and sampling rules, hence obtaining the best of these two worlds: the algorithms (1) remain fully adaptive, (2) suffer a sample complexity that is never worse of their non-elimination counterpart, and (3) provably eliminate certain wrong answers early. We confirm these benefits experimentally, where elimination improves significantly the computational complexity of adaptive methods on common tasks like best-arm identification in linear bandits.

1 Introduction

The multi-armed bandit is a sequential decision-making task which is now extensively studied (see, e.g., [1] for a recent review). In this problem, an algorithm interacts with its environment by sequentially "pulling" one among $K \in \mathbb{N}$ arms and observing a sample from a corresponding distribution. Among the possible objectives, we focus on *fixed-confidence identification* [2, 3, 4, 5]. In this setting, the algorithm successively collects samples until it decides to stop and return an answer to a given query about the distributions. Its task is to return the correct answer with at most a given probability of error δ, and its secondary goal is to do so while stopping as early as possible. This problem is called "fixed-confidence" as opposed to "fixed-budget", where the goal is to minimize the error probability with at most a given number of samples [6, 7, 8, 9, 10].

The most studied query is *best arm identification* (BAI), where the aim is to return the arm whose distribution has highest mean. A variant is Top-m identification [11], where the goal is to find the m arms with highest means. While these are the most common, other queries have been studied, including thresholding bandits [9], minimum threshold [12], and multiple correct answers [13].

*Work done while at Inria Lille.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Algorithms for fixed-confidence identification can be generally divided into two classes: those based on *adaptive sampling* and those based on *elimination*. Adaptive algorithms [e.g., 8, 11, 4, 14] update their sampling strategy at each round and typically stop when they can simultaneously assess the correctness of every answer. They often enjoy strong theoretical guarantees. For instance, some of them [4, 15, 16, 17] have been shown to be optimal as $\delta \to 0$. However, since they repeatedly test the correctness of every answer, they are often computationally demanding. Elimination-based strategies [e.g., 2, 18, 19, 20, 21] maintain a set of "active" answers (those that are still likely to be the correct one) and stop when only one remains. They typically update their sampling rules and/or the active answers infrequently. This, together with the fact that eliminations reduce the problem size over time, makes them more computationally efficient but also yields large sample complexity in practice. Moreover, while adaptive algorithms for general identification problems (i.e., with arbitrary queries) exist [4, 15, 17], elimination-based strategies are not easy to design at such a level of generality. In particular, they are not easy to extend to structured combinatorial problems (such as Top-m), where the number of answers is exponential in the problem dimension.[2]

In this paper, we design a novel elimination rule for general identification problems which we call *selective* elimination. It can be easily combined with existing adaptive strategies, both in their stopping and sampling rules, making them achieve the best properties of the two classes mentioned above. In particular, we prove that (1) selective elimination never suffers worse sample complexity than the original algorithm, and hence remain asymptotically optimal whenever the base algorithm is; (2) It provably discards some answers much earlier than the stopping time; (3) It improves the computational complexity of the original algorithm when some answers are eliminated early. Experimentally, we compare several existing algorithms for three identification problems (BAI, Top-m, and thresholding bandits) on two bandit structures (linear and unstructured). We find that, coherently across all experiments, existing adaptive strategies achieve significant gains in computation time and, to a smaller extent, in sample complexity when combined with selective elimination.

1.1 Bandit fixed-confidence identification

An algorithm interacts with an environment composed of $K > 1$ *arms*. At each time $t \in \mathbb{N}$, the algorithm picks an arm k_t and observes $X_t^{k_t} \sim \nu_{k_t}$, where ν_{k_t} is the distribution of arm k_t. At a time τ, the algorithm stops and returns an answer $\hat{\imath}$ from a finite set $\mathcal{I}$. Formally, let $\mathcal{F}_t$ be the σ-algebra generated by the observations up to time t. An identification algorithm is composed of

1. *Sampling rule*: the sequence $(k_t)_{t\in\mathbb{N}}$, where k_t is $\mathcal{F}_{t-1}$-measurable.
2. *Stopping rule*: a stopping time τ with respect to $(\mathcal{F}_t)_{t\in\mathbb{N}}$ and a random variable $\hat{\imath} \in \mathcal{I}$, i.e., the answer returned when stopping at time τ.

Note that, while it is common to decouple τ and $\hat{\imath}$, we group them to emphasize that the time at which an algorithm stops depends strongly on the answer it plans on returning.

We assume that the arm distributions depend on some unknown parameter $\theta \in \mathcal{M}$, where $\mathcal{M} \subseteq \mathbb{R}^d$ is the set of possible parameters, and write $\nu_k(\theta)$ for $k \in [K]$ to make this dependence explicit. For simplicity, we shall use θ to refer to the bandit problem $(\nu_k(\theta))_{k\in[K]}$. This assumption allows us to include linear bandits in our analysis. We let $i^\star : \mathcal{M} \to \mathcal{I}$ be the function, known to the algorithm, which returns the unique correct answer for each problem. The algorithm is correct on θ if $\hat{\imath} = i^\star(\theta)$.

Definition 1.1 (δ-correct algorithm)**.** *An algorithm is said to be δ-correct on $\mathcal{M} \subseteq \mathbb{R}^d$ if for all $\theta \in \mathcal{M}$, $\tau < +\infty$ almost surely and $\mathbb{P}_\theta(\hat{\imath} \neq i^\star(\theta)) \leq \delta$.*

We want to design algorithms that, given a value δ, are δ-correct on $\mathcal{M}$ and have minimal expected sample complexity $\mathbb{E}_\theta[\tau]$ for all $\theta \in \mathcal{M}$. A lower bound on $\mathbb{E}_\theta[\tau]$ was proved in [4]. In order to present it, we introduce the concept of *alternative* set to an answer $i \in \mathcal{I}$: $\Lambda(i) := \{\lambda \in \mathcal{M} \mid i^\star(\lambda) \neq i\}$, the set of parameters for which the correct answer is not i. Let us denote by $\mathrm{KL}_k(\theta, \lambda)$ the Kullback-Leibler (KL) divergence between the distribution of arm k under θ and λ. Then the lower bound states that for any algorithm that is δ-correct on $\mathcal{M}$ and any problem $\theta \in \mathcal{M}$,

$$\mathbb{E}_\theta[\tau] \geq \log(1/(2.4\delta))/H^\star(\theta)\ , \text{with } H^\star(\theta) := \max_{\omega\in\Delta_K} \inf_{\lambda\in\Lambda(i^\star(\theta))} \sum_{k\in[K]} \omega^k \,\mathrm{KL}_k(\theta, \lambda)\ . \qquad (1)$$

[2] An elimination strategy for specific unstructured combinatorial problems has been introduced in [22].

Example: BAI in Gaussian linear bandits While our results apply to general queries, we illustrate all statements of this paper on the widely-studied task of BAI in Gaussian linear bandits [19, 14, 23, 16, 24]. In this setting, each arm $k \in [K]$ has a Gaussian distribution $\mathcal{N}(\mu_k(\theta), 1)$ with mean $\mu_k(\theta) = \phi_k^\top \theta$, a linear function of the unknown parameter $\theta \in \mathbb{R}^d$ (and $\mathcal{M} = \mathbb{R}^d$) and of known arm features $\phi_k \in \mathbb{R}^d$. The set of answers is $\mathcal{I} = [K]$ and the correct answer is $i^\star(\theta) := \arg\max_{k \in [K]} \phi_k^\top \theta$.

Finally, for $x \in \mathbb{R}^d$ and $A \in \mathbb{R}^{d \times d}$, we define $\|x\|_A := \sqrt{x^\top A x}$. For $\omega \in \mathbb{R}^K$, let $V_\omega := \sum_{k=1}^K \omega^k \phi_k \phi_k^\top$. With this notation, we have $\sum_{k \in [K]} \omega^k \operatorname{KL}_k(\theta, \lambda) = \frac{1}{2}\|\theta - \lambda\|^2_{V_\omega}$.

1.2 Log-likelihood ratio stopping rules

Most existing adaptive algorithms use a log-likelihood ratio (LLR) test in order to decide when to stop. Informally, they check whether sufficient information has been collected to confidently discard at once all answers except one. Since such LLR tests are crucial for the design of our general elimination rules, we now describe their principle.

Given two parameters $\theta, \lambda \in \mathcal{M}$, the LLR of observations $X_{[t]} = (X_1^{k_1}, \dots, X_t^{k_t})$ between models θ and λ is $L_t(\theta, \lambda) := \log \frac{d\mathbb{P}_\theta}{d\mathbb{P}_\lambda}(X_{[t]}) = \sum_{s=1}^t \log \frac{d\mathbb{P}_\theta}{d\mathbb{P}_\lambda}(X_s^{k_s})$. Let $\hat{\theta}_t := \arg\max_{\lambda \in \mathcal{M}} \log \mathbb{P}_\lambda(X_{[t]})$ be the maximum likelihood estimator of θ from t observations. In Gaussian linear bandits, we have $L_t(\theta, \lambda) = \frac{1}{2}\|\theta - \lambda\|^2_{V_{N_t}} + (\theta - \lambda)^\top V_{N_t}(\hat{\theta}_t - \theta)$, where $N_t^k := \sum_{s=1}^t \mathbb{1}(k_s = k)$. See Appendix C for more details. $L_t(\theta, \lambda)$ is closely related to $\sum_{k=1}^K N_t^k \operatorname{KL}_k(\theta, \lambda)$, a quantity that appears frequently in our results. Indeed, the difference between these quantities is a martingale, which is a lower order term compared to them. The LLR stopping rule was introduced to the bandit literature in [4]. At each step $t \in \mathbb{N}$, the algorithm computes the infimum LLR to the alternative set of $i^\star(\hat{\theta}_t)$ and stops if it exceeds a threshold, i.e., if

$$\inf_{\lambda \in \Lambda(i^\star(\hat{\theta}_t))} L_t(\hat{\theta}_t, \lambda) \geq \beta_{t,\delta} \ , \tag{2}$$

where the function $\beta_{t,\delta}$ can vary, notably based on the shape of the alternative sets. The recommendation rule is then $\hat{\imath} = i^\star(\hat{\theta}_t)$. Informally, the algorithm stops if it has enough information to exclude all points λ for which the answer is not $i^\star(\hat{\theta}_t)$. This stopping rule enforces δ-correctness, provided that the sampling rule ensures $\tau < +\infty$ a.s. and that $\beta_{t,\delta}$ is properly chosen. The most popular choice is to ensure a concentration property of $L_t(\hat{\theta}_t, \theta)$. For example, if for all δ, $\beta_{t,\delta}$ guarantees that

$$\mathbb{P}\left(\exists t \geq 1 : L_t(\hat{\theta}_t, \theta) \geq \beta_{t,\delta}\right) \leq \delta, \tag{3}$$

LLR stopping with that threshold returns a wrong answer with probability at most δ. Such concentration bounds can be found in [25, 26] for linear and unstructured bandits, respectively. This LLR stopping rule is used in many algorithms [4, 14, 15, 16, 24, 17][3]. Some of them have been proven to be *asymptotically optimal*: their sample complexity upper bound matches the lower bound (1) when $\delta \to 0$. However, improvements are still possible: their sample complexity for moderate δ may not be optimal and their computational complexity may be reduced, as we will see.

2 Elimination stopping rules for adaptive algorithms

We show how to modify the stopping rule of adaptive algorithms using LLR stopping to perform elimination. We assume that the alternatives sets $\Lambda(i)$ can be decomposed into a union of sets which we refer to as *alternative pieces* (or simply pieces), with the property that computing the infimum LLR over these sets is computationally easy.

Assumption 2.1. *For all $i \in \mathcal{I}$, there exist pieces $(\Lambda_p(i))_{p \in \mathcal{P}(i)}$, where $\mathcal{P}(i)$ is a finite set of piece indexes, such that $\Lambda(i) = \bigcup_{p \in \mathcal{P}(i)} \Lambda_p(i)$ and $\inf_{\lambda \in \Lambda_p(i)} L_t(\hat{\theta}_t, \lambda)$ can be efficiently computed for all $p \in \mathcal{P}(i)$ and $t > 0$.*

[3] LinGapE [14] does not use LLR stopping explicitly, but its stopping rule is equivalent to it. We can write it as: stop if for all points inside a confidence region a gap is small enough, that is if all those points do not belong to the alternative of $i^\star(\hat{\theta}_t)$. The contrapositive of that statement is exactly LLR stopping.

This assumption is satisfied in many problems of interest, including BAI, Top-m identification, and thresholding bandits (see Appendix B). Indeed, in all applications we consider in this paper, the sets of Assumption 2.1 are half-spaces. In our linear BAI example, the piece indexes are simply arms. For $i, j \in [K]$ we can define $\Lambda_j(i) = \{\lambda \in \mathcal{M} \mid \phi_j^\top \lambda > \phi_i^\top \lambda\}$. Then, $\Lambda(i) = \bigcup_{j \in [K] \setminus \{i\}} \Lambda_j(i)$. Moreover, the infimum LLR (and the corresponding minimizer) can be computed in closed form as [e.g., 20] $\inf_{\lambda \in \Lambda_j(i)} L_t(\hat{\theta}_t, \lambda) = \max\{\hat{\theta}_t^T(\phi_i - \phi_j), 0\}^2 / \|\phi_i - \phi_j\|^2_{V_{N_t}^{-1}}$.

Elimination stopping The main idea is that it is not necessary to exclude all $\Lambda_p(i)$ for $p \in \mathcal{P}(i)$ at the *same time*, as LLR stopping (2) does[4], in order to know that the algorithm can stop and return answer i. Instead, each piece can be pruned as soon as we have enough information to do so.

Definition 2.2. *A set $S \subseteq \mathbb{R}^d$ is said to be eliminated at time t if, for all $\lambda \in S$, $L_t(\hat{\theta}_t, \lambda) \geq \beta_{t,\delta}$.*

From the concentration property (3), we obtain that the probability that $\theta \in S$ and S is eliminated is less than δ. LLR stopping interrupts the algorithm when the alternative set $\Lambda(i^\star(\hat{\theta}_t))$ can be eliminated. In elimination stopping, we eliminate smaller sets gradually, instead of the whole alternative at once. Formally, let us define, for all $i \in \mathcal{I}$,

$$\overline{\mathcal{P}}_t(i; \beta_{t,\delta}) = \left\{ p \in \mathcal{P}(i) : \inf_{\lambda \in \Lambda_p(i)} L_t(\hat{\theta}_t, \lambda) < \beta_{t,\delta} \right\} \tag{4}$$

as the subset of pieces for answer $i \in \mathcal{I}$ whose infimum LLR at time t is below a threshold $\beta_{t,\delta}$. That is, the indexes of pieces that are *not* eliminated at time t. Moreover, we define, for all $i \in \mathcal{I}$, a set of *active pieces* $\mathcal{P}_t^{\mathrm{stp}}(i)$ which is initialized as $\mathcal{P}_0^{\mathrm{stp}}(i) = \mathcal{P}(i)$ (all piece indexes).

Our *selective elimination* rule updates, at each time t, only the active pieces of the empirical answer $i^\star(\hat{\theta}_t)$. That is, for $i = i^\star(\hat{\theta}_t)$, it sets

$$\mathcal{P}_t^{\mathrm{stp}}(i) := \mathcal{P}_{t-1}^{\mathrm{stp}}(i) \cap \overline{\mathcal{P}}_t(i; \beta_{t,\delta}), \tag{5}$$

while it sets $\mathcal{P}_t^{\mathrm{stp}}(i) := \mathcal{P}_{t-1}^{\mathrm{stp}}(i)$ for all $i \neq i^\star(\hat{\theta}_t)$. One might be wondering why not updating all answers at each round. The main reason is computational: as we better discuss at the end of this section, checking LLR stopping requires one minimization for *each* piece $p \in \mathcal{P}(i^\star(\hat{\theta}_t))$, while selective elimination requires only one for each *active* piece $p \in \mathcal{P}_{t-1}^{\mathrm{stp}}(i^\star(\hat{\theta}_t))$. Thus, the latter becomes increasingly more computationally efficient as pieces are eliminated. For completeness, we also analyze the variant, that we call *full elimination*, which updates the active pieces according to (5) for *all* answers $i \in \mathcal{I}$ at each round. While we establish slightly better theoretical guarantees for this rule, it is computationally demanding and, as we shall see in our experiments, it does not significantly improve sample complexity w.r.t. selective elimination, which remains our recommended choice.

Let $\tau_{\mathrm{s.elim}} = \inf_{t \geq 1}\{t \mid \mathcal{P}_t^{\mathrm{stp}}(i^\star(\hat{\theta}_t)) = \emptyset\}$ and $\tau_{\mathrm{f.elim}} := \inf_{t \geq 1}\{t \mid \exists i \in \mathcal{I} : \mathcal{P}_t^{\mathrm{stp}}(i) = \emptyset\}$ be the stopping times of selective and full elimination, respectively. Intuitively, these two rules stop when one of the updated answers has all its pieces eliminated (and return that answer). We show that, as far as $\beta_{t,\delta}$ is chosen to ensure concentration of $\hat{\theta}_t$ to θ, those two stopping rules are δ-correct.

Lemma 2.3 (δ-correctness)**.** *Suppose that $\beta_{t,\delta}$ guarantees* (3) *and that the algorithm verifies that, whenever it stops, there exists $i_\emptyset \in \mathcal{I}$ such that $\mathcal{P}_\tau^{\mathrm{stp}}(i_\emptyset) = \emptyset$ and $\hat{\imath} = i_\emptyset$. Then, $\mathbb{P}_\theta(\hat{\imath} \neq i^\star(\theta)) \leq \delta$.*

All proofs for this section are in Appendix D. If an algorithm verifies the conditions of Lemma 2.3 and has a sampling rule that makes it stop almost surely, then it is δ-correct. Interestingly, we can prove a stronger result than δ-correctness: under the same sampling rule, the elimination stopping rules never trigger later than the LLR one *almost surely*. In other words, any algorithm equipped with elimination stopping suffers a sample complexity that is never worse than the one of the same algorithm equipped with LLR stopping. Let $\tau_{\mathrm{llr}} := \inf_{t \geq 1}\{t \mid \inf_{\lambda \in \Lambda(i^\star(\hat{\theta}_t))} L_t(\hat{\theta}_t, \lambda) \geq \beta_{t,\delta}\}$.

Theorem 2.4. *For any sampling rule, almost surely $\tau_{\mathrm{f.elim}} \leq \tau_{\mathrm{s.elim}} \leq \tau_{\mathrm{llr}}$.*

The proof of this theorem is very simple: if $\tau_{\mathrm{llr}} = t$, then at t all pieces $\Lambda_p(i^\star(\hat{\theta}_t))$ for $p \in \mathcal{P}(i^\star(\hat{\theta}_t))$ can be eliminated, hence $\tau_{\mathrm{s.elim}} \leq t$. The proof that $\tau_{\mathrm{f.elim}} \leq \tau_{\mathrm{s.elim}}$ follows from the observation

[4] Under Assumption 2.1, LLR stopping is written as $\min_{p \in \mathcal{P}(i)} \inf_{\lambda \in \Lambda_p(i)} L_t(\hat{\theta}_t, \lambda) \geq \beta_{t,\delta}$ for $i = i^\star(\hat{\theta}_t)$, which implies that all alternative pieces of answer i are discarded at once.

that full elimination always has less active pieces than selective elimination. Note that all three stopping rules must use the same threshold $\beta_{t,\delta}$ to be comparable. Although simple, Theorem 2.4 has an important implication: we can take any existing algorithm that uses LLR stopping, equip it with elimination stopping instead, and obtain a new strategy that is never worse in terms of sample complexity and for which the original theoretical results on the stopping time still hold.

Finally, it is important to note that, while defining the elimination rule in the general form (5) allows us to unify many settings, storing/iterating over all sets $\mathcal{P}_t^{\mathrm{stp}}(i)$ would be intractable in problems with large number of answers (e.g., top-m identification or thresholding bandits, where the latter is exponential in K). Fortunately, we show in Appendix B that this is not needed and efficient implementations exist for these problems that take only polynomial time and memory.

2.1 Elimination time of alternative pieces

We now show that elimination stopping can indeed discard certain alternative pieces much earlier that the stopping time. While all results so far hold for any distribution and bandit structure, in the remaining we focus on Gaussian linear bandits. Other distribution classes beyond Gaussians could be used with minor modifications (see Appendix C.2) but the Gaussian case simplifies the exposition. Since most existing adaptive sampling rules target the optimal proportions from the lower bound of [4], we unify them under the following assumption.

Assumption 2.5. *Consider the concentration events*

$$E_t := \left\{ \forall s \leq t : L_s(\hat{\theta}_s, \theta) \leq \beta_{t,1/t^2} \right\} . \tag{6}$$

A sampling rule is said to have low information regret if there exists a problem-dependent function $R(\theta, t)$ which is sub-linear in t such that for each time t where E_t holds,

$$\inf_{\lambda \in \Lambda(i^\star(\theta))} \sum_{k \in [K]} N_t^k \, \mathrm{KL}_k(\theta, \lambda) \geq t H^\star(\theta) - R(\theta, t). \tag{7}$$

The left-hand side of (7) can be understood as the information collected by the sampling rule at time t to discriminate θ with all its alternatives. Therefore, Assumption 2.5 requires that information to be comparable (up to a low-order term $R(\theta, t)$) with the maximal one from the lower bound. In Appendix F, we show that this is satisfied by both Track-and-Stop [4] and the approach in [15].

Let $H_p(\omega, \theta) := \inf_{\lambda \in \Lambda_p(i^\star(\theta))} \sum_{k \in [K]} \omega^k \, \mathrm{KL}_k(\theta, \lambda)$, the information that sampling with proportions ω brings to discriminate θ from the alternative piece $\Lambda_p(i^\star(\theta))$. Note that $H^\star(\theta) = \max_{\omega \in \Delta_K} \min_{p \in \mathcal{P}(i^\star(\theta))} H_p(\omega, \theta)$. For $\epsilon \geq 0$, let $\Omega_\epsilon(\theta) := \{\omega \in \Delta_K \mid \inf_{\lambda \in \Lambda(i^\star(\theta))} \sum_k \omega^k \, \mathrm{KL}_k(\theta, \lambda) \geq H^\star(\theta) - \epsilon\}$ be the set of ϵ-optimal proportions.

Theorem 2.6 (Piece elimination)**.** *The stopping time of any sampling rule having low information regret, combined with LLR stopping, satisfies $\mathbb{E}[\tau] \leq \bar{t} + 2$, where $\bar{t}$ is the first integer such that*

$$t \geq \left(\left(\sqrt{\beta_{t,\delta}} + \sqrt{\beta_{t,1/t^2}} \right)^2 + R(\theta, t) \right) / H^\star(\theta). \tag{8}$$

When the same sampling rule is combined with elimination stopping, let τ_p be the time at which $p \in \mathcal{P}(i^\star(\theta))$ is eliminated. Then, $\mathbb{E}[\tau_p] \leq \min\{\bar{t}_p, \bar{t}\} + 2$, where $\bar{t}_p$ is the first integer such that

$$t \geq \max \left\{ \frac{\left(\sqrt{\beta_{t,\delta}} + \sqrt{\beta_{t,1/t^2}} \right)^2}{\min_{\omega \in \Omega_{R(\theta,t)/t}(\theta)} H_p(\omega, \theta)}, G(\theta, t) \right\}, \tag{9}$$

with $G(\theta, t) = 0$ for full elimination and $G(\theta, t) = \frac{4\beta_{t,1/t^2} + R(\theta,t)}{H^\star(\theta)}$ for selective elimination.

First, the bound we obtain on the elimination time of pieces in $\mathcal{P}(i^\star(\theta))$ is not worse than the bound we obtain on the stopping time of LLR stopping. Second, with elimination stopping, such eliminations can actually happen much sooner. Intuitively, sampling rules with low information regret play arms with proportions that are close to the optimal ones. If all of such "good" proportions provide large information for eliminating some piece $p \in \mathcal{P}(i^\star(\theta))$, then p is eliminated much sooner than the actual stopping time (which requires eliminating the worst-case piece in the same set).

While both elimination rules are provably efficient, with full elimination enjoying slighly better guarantees[5], selective elimination provably never worsens (and possibly improves) the computational complexity over LLR stopping. In all applications we consider, implementing LLR stopping requires one minimization for each of the same alternative pieces we use for elimination stopping. Therefore, the total number of minimizations required by LLR stopping is $\sum_{t=1}^{\tau_{\mathrm{llr}}} |\mathcal{P}(i^\star(\hat{\theta}_t))|$ versus $\sum_{t=1}^{\tau_{\mathrm{s.elim}}} |\mathcal{P}_t^{\mathrm{stp}}(i^\star(\hat{\theta}_t))|$ for selective elimination. The second is never larger since $\tau_{\mathrm{s.elim}} \leq \tau_{\mathrm{llr}}$ by Theorem 2.4 and $\mathcal{P}_t^{\mathrm{stp}}(i^\star(\hat{\theta}_t)) \subseteq \mathcal{P}(i^\star(\hat{\theta}_t))$ for all t, and much smaller if eliminations happen early, as we shall verify in experiments. In our linear BAI example we need to perform $(K-1)$ minimizations at each step, one for each sub-optimal arm, in order to implement LLR stopping. On the other hand, we need only $|\mathcal{P}_t^{\mathrm{stp}}(i^\star(\hat{\theta}_t))|$ minimizations with selective elimination, one for each active sub-optimal arm, while full elimination takes $\sum_{i\in[K]} |\mathcal{P}_t^{\mathrm{stp}}(i)|$ to update all the sets.

Note that Theorem 2.6 does not provide a better bound on $\mathbb{E}[\tau]$ for elimination stopping than for LLR stopping. In fact, when evaluating the bound on $\mathbb{E}[\tau_p]$ for the worst-case piece in $p \in \mathcal{P}(i^\star(\theta))$, we recover the one on $\mathbb{E}[\tau]$. This is intuitive since the sampling rule is playing proportions that try to eliminate all alternative pieces at once. The following result formalizes this intuition.

Theorem 2.7. *Suppose that we can write $\beta_{t,\delta} = \log\frac{1}{\delta} + \xi(t,\delta)$ with $\lim_{\delta\to 0} \xi(t,\delta)/\log(1/\delta) = 0$. Then for any sampling rule that satisfies Assumption 2.5,*

$$\mathbb{E}[\tau_{\mathrm{llr}}] \leq \mathbb{E}[\tau_{\mathrm{elim}}] + f(\theta,\delta)\ .$$

with $\lim_{\delta\to 0} f(\theta,\delta)/\log(1/\delta) = 0$. Here τ_{elim} can stand for either full or selective elimination.

See Appendix D.4 for f. This result shows that when the sampling rule is tailored to the LLR stopping rule, the expected LLR and elimination stopping times differ by at most low-order (in $\log(1/\delta)$) terms. As $\delta \to 0$ the two expected stopping times converge to the same value $H^\star(\theta)^{-1}\log(1/\delta)$, which is the asymptotically-optimal sample complexity prescribed by the lower bound (1).

We showed that, for both elimination rules, some pieces of the alternative are discarded sooner than the stopping time, and that the overall sample complexity of the method can only improve over LLR stopping. However, since the sampling rule of the algorithm was not changed, elimination does not change the computational cost of each sampling step, only the cost of checking the stopping rule.

2.2 An example

We compare LLR and elimination stopping on a simple example so as to better quantify the elimination times of Theorem 2.6 and their computational impact (see Appendix H.1 for a full discussion).

Consider BAI in a Gaussian linear bandit instance with unit variance, $d = 2$, and arbitrary number of arms $K \geq 3$ (see Figure 1). The arm features are $\phi_1 = (1,0)^T$, $\phi_2 = (0,1)^T$, and, for all $i = 3,\dots,K$, $\phi_i = (a_i, b_i)^T$ with a_i, b_i arbitrary values in $(-1,0)$ such that $\|\phi_i\|_2 = 1$. The true parameter is $\theta = (1, 1-\varepsilon)^T$, for $\varepsilon \in (0, 1/2)$ a possibly very small value. Arm 1 is optimal with mean $\mu_1(\theta) = 1$, while arm 2 is sub-optimal with mean $\mu_2(\theta) = 1-\varepsilon$. For all other arms $i = 3,\dots,K$, $\mu_i(\theta) \leq 0$.

Let $\omega \in \Delta_K$ be any allocation. Recall that in BAI each piece index is simply an arm, and $\mathcal{P}(i^\star(\theta)) = \mathcal{P}(1) = \{2,\dots,K\}$. Let $k \in \mathcal{P}(1)$ be any sub-optimal arm. The distance to the k-th alternative piece $H_k(\omega,\theta)$ can be computed in closed form as $H_k(\omega,\theta) = ((\phi_1 - \phi_k)^T\theta)^2/(2\|\phi_1 - \phi_k\|^2_{V_\omega^{-1}})$. The optimal allocation is $\omega^\star = \arg\max_\omega \min_k H_k(\omega,\theta) = (1/2, 1/2, 0, \dots, 0)^\top$.

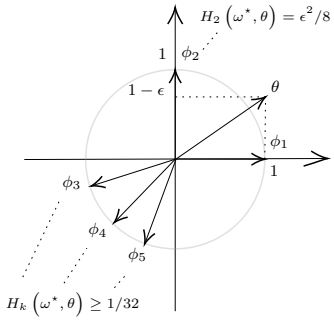

Figure 1: Example of BAI instance with $d = 2$ and $K = 5$.

The intuition why this example is interesting is as follows. Any correct strategy is required to discriminate between arm 1 and 2 (i.e., to figure out that arm 1 is optimal), which requires roughly $O(1/\varepsilon^2)$ samples from both. An optimal strategy plays these two arms nearly with the same proportions. Since ϕ_1 and ϕ_2 form the canonical basis of $\mathbb{R}^2$, the samples collected by this strategy are informative for estimating the

[5]Note that $G(\theta,t)$ for selective elimination contributes only a finite (in δ) sample complexity.

mean reward of *every* arm, even those than are not played. Then, since arms $3, \dots, K$ have at least a sub-optimality gap of 1, an elimination-based strategy discards them with a number of samples not scaling with $1/\varepsilon^2$. This means that a non-elimination strategy runs for $O(1/\varepsilon^2)$ steps over the original problem with K arms, while an elimination-based one quickly reduces the problem to one with only 2 arms. The main impact is computational: since most algorithms need to compute some statistics for each active arm at each round (e.g., closest alternatives, confidence intervals, etc.), the computational complexity of a non-elimination algorithm is at least $O(K/\varepsilon^2)$, while the one of an elimination-based variant is roughly $O(K + 1/\varepsilon^2)$, a potentially very large improvement.

We now quantify the elimination times and computational complexity on this example. Since such quantities depend on the specific sampling rule, we do it for an oracle strategy that samples according to $\omega^\star$. Similar results can be derived for any low information regret sampling rule (see Appendix H.1).

Proposition 2.8. *For any $K \geq 3$ and $\varepsilon \in (0, 1/2)$, for any $\delta \in (0, 1)$, the oracle strategy combined with LLR stopping satisfies on the example instance*

$$\mathbb{E}[\tau] \geq \Omega\left(\frac{\log(1/\delta)}{\varepsilon^2}\right).$$

On the same instance, for the oracle strategy with elimination at stopping and a threshold $\beta_{t,\delta} = \log(1/\delta) + O(\log(t))$, the expected elimination time of any piece (i.e., arm) $k \geq 3$ is

$$\mathbb{E}[\tau_k] \leq \widetilde{O}(\log(1/\delta)) \qquad \textit{for full elimination},$$

$$\mathbb{E}[\tau_k] \leq \widetilde{O}\left(\log(1/\delta) + \frac{1}{\varepsilon^2}\right) \qquad \textit{for selective elimination}.$$

Moreover, the expected per-round computation time of the oracle strategy with LLR stopping is $\Omega(K)$, while it is at most $O(K^2\varepsilon^2)$ for full elimination and $O(K\varepsilon^2 + K/\log(1/\delta))$ for selective elimination.

3 Elimination at sampling

We show how to adapt sampling rules in order to accommodate piece elimination. There are two reasons for doing this: first, adapting the sampling to ignore pieces that have been discarded could reduce the sample complexity; second, the amount of computations needed to update the sampling strategy is often proportional to the number of pieces and decreasing it can reduce the overall time.

We start from an algorithm using LLR stopping, for which we change the stopping rule as above. The sampling strategies that we can adapt are those that aggregate information from each alternative piece. For example, in linear BAI, methods that mimic the lower bound allocation (1), like Track-and-Stop [4], LinGame [16], or FWS [17], and even LinGapE [14], all compute distances or closest points to each piece in the decomposition $\{\lambda \mid \phi_j^\top \lambda \geq \phi_{i^\star(\hat{\theta}_t)}^\top \lambda\}$. Eliminating pieces at sampling simply means omitting from such computations the arms that were deemed sub-optimal. Algorithm 1 shows how Track-and-Stop [4] can be modified to incorporate elimination at sampling and stopping.

Algorithm 1 Track-and-Stop [4]: vanilla (left) and with selective elimination (right)

while not stopped **do**

  for $p \in \mathcal{P}(i^\star(\hat{\theta}_t))$ **do** ▷ stopping
    $L_{p,t} = \inf_{\lambda \in \Lambda_p(i^\star(\hat{\theta}_t))} L_t(\hat{\theta}_t, \lambda)$

  end for
  if $\forall p \in \mathcal{P}(i^\star(\hat{\theta}_t)) : L_{p,t} > \beta_{t,\delta}$ **then** STOP
  $w_t = \arg\max_\omega \min_{p \in \mathcal{P}(i^\star(\hat{\theta}_t))} H_p(\omega, \hat{\theta}_t)$
  if $\exists k : N_t^k < \sqrt{t}$ pull $k_{t+1} = \arg\min_k N_t^k$
  else pull $k_{t+1} = \arg\min_k (N_t^k - t w_t^k)$

end while

while not stopped **do**
  Set $\mathcal{P}_t^{\mathrm{stp}}(i^\star(\hat{\theta}_t)) = \mathcal{P}_{t-1}^{\mathrm{stp}}(i^\star(\hat{\theta}_t))$
  for $p \in \mathcal{P}_{t-1}^{\mathrm{stp}}(i^\star(\hat{\theta}_t))$ **do** ▷ stopping
    $L_{p,t} = \inf_{\lambda \in \Lambda_p(i^\star(\hat{\theta}_t))} L_t(\hat{\theta}_t, \lambda)$
    if $L_{p,t} > \beta_{t,\delta}$ **delete** p from $\mathcal{P}_t^{\mathrm{stp}}(i^\star(\hat{\theta}_t))$
  end for
  if $\mathcal{P}_t^{\mathrm{stp}}(i^\star(\hat{\theta}_t)) = \emptyset$ **then** STOP
  $w_t = \arg\max_\omega \min_{p \in \mathcal{P}_t^{\mathrm{smp}}(i^\star(\hat{\theta}_t))} H_p(\omega, \hat{\theta}_t)$
  if $\exists k : N_t^k < \sqrt{t}$ pull $k_{t+1} = \arg\min_k N_t^k$
  else pull $k_{t+1} = \arg\min_k (N_t^k - t w_t^k)$
  Update $\mathcal{P}_{t+1}^{\mathrm{smp}}(i^\star(\hat{\theta}_t))$ (Algorithm 3)
end while

Similarly to elimination stopping, the idea is to maintain sets of active pieces at sampling $\mathcal{P}_t^{\mathrm{smp}}(i)$ for each $i \in \mathcal{I}$. Note that these are different from the ones introduced in Section 2 for the stopping

rule. The set is updated at each step like we did for the stopping sets, but with a different threshold $\alpha_{t,\delta}$ (see Appendix E for details). Additionally, we reset it very infrequently at steps $t \in \{\bar{t}_0^{2^j}\}_{j\geq 0}$, where $\bar{t}_0 \geq 2$. Formally, let us define the helper sets $\tilde{\mathcal{P}}_t^{\mathrm{smp}}(i)$ as $\tilde{\mathcal{P}}_0^{\mathrm{smp}}(i) := \mathcal{P}(i)$ and

$$\tilde{\mathcal{P}}_t^{\mathrm{smp}}(i) := \begin{cases} \tilde{\mathcal{P}}_{t-1}^{\mathrm{smp}}(i) \cap \overline{\mathcal{P}}_t(i;\alpha_{t,\delta}) & \text{if } t \notin \{\bar{t}_0^{2^j}\}_{j\geq 0} \\ \overline{\mathcal{P}}_t(i;\alpha_{t,\delta}) & \text{otherwise,} \end{cases}$$

where $\overline{\mathcal{P}}_t$ was defined in (4). Let $\bar{t}_j := \bar{t}_0^{2^j}$ be the time step at which the j-th reset is performed and $j(t) := \lfloor \log_2 \log_{\bar{t}_0} t \rfloor$ be the index of the last reset before t. We define $\mathcal{P}_t^{\mathrm{smp}}(i) := \tilde{\mathcal{P}}_t^{\mathrm{smp}}(i) \cap \tilde{\mathcal{P}}_{\bar{t}_{j(t)}-1}^{\mathrm{smp}}(i)$, such that $\mathcal{P}_t^{\mathrm{smp}}(i)$ is the intersection of all active pieces from the second-last reset up to t, i.e., $\mathcal{P}_t^{\mathrm{smp}}(i) = \bigcap_{s=\bar{t}_{j(t)-1}}^{t} \overline{\mathcal{P}}_s(i;\alpha_{s,\delta})$. Since the resets are very infrequent, this definition only drops a small number of rounds from the intersection (less than $\sqrt{t}$). The detailed procedure to update these sets is summarized in Algorithm 3. As before, we can instantiate both selective and full elimination.

The reason for the resets is two-fold. First, they ensure that the algorithm stops almost surely as required by Definition 1.1. In fact, without resets, it might happen with some small (less than δ) probability that pieces containing the true parameter are eliminated, in which case the sampling rule could diverge. Second, they guarantee that the thresholds $(\alpha_{s,\delta})_{s=\bar{t}_{j(t)-1}}^{t}$ used in $\mathcal{P}_t^{\mathrm{smp}}(i)$ are within a constant factor of each other. This is crucial to relate the LLR of active pieces at different times.

3.1 Properties

We consider a counterpart of Assumption 2.5 for sampling rules combined with piece elimination.

Assumption 3.1. *There exists a sub-linear (in t) problem-dependent function $R(\theta, t)$ such that, for each time t where E_t (defined in Equation 6) holds,*

$$\min_{p \in \mathcal{P}_t^{\mathrm{smp}}(i^\star(\theta))} H_p(N_t, \theta) \geq \max_{\omega \in \Delta_K} \sum_{s=1}^{t} \min_{p \in \mathcal{P}_{s-1}^{\mathrm{smp}}(i^\star(\theta))} H_p(\omega, \theta) - R(\theta, t).$$

Intuitively, the sampling rule maximizes the information for discriminating θ with all its alternatives from the sequence of active pieces $(\mathcal{P}_{s-1}^{\mathrm{smp}}(i^\star(\theta)))_{s=1}^{t}$. We prove in Appendix F that the algorithms for which we proved Assumption 2.5 also satisfy Assumption 3.1 when their sampling rules are combined with either full or selective elimination.

Theorem 3.2. *Consider a sampling rule that verifies Assumption 3.1 and uses either full or selective elimination with the sets $\mathcal{P}_t^{\mathrm{smp}}$. Then, Assumption 2.5 holds as well. Moreover, when using the same elimination rule at stopping, such a sampling rule verifies Theorem 2.6, i.e., it enjoys the same guarantees as without elimination at sampling.*

The proof is in Appendix E. Theorem 3.2 shows that for an algorithm using elimination at sampling and stopping, we get bounds on the times at which pieces of $\Lambda(i^\star(\theta))$ are discarded from the stopping rule which are not worse than those we obtained for the same algorithm without elimination at sampling. This result is non-trivial. We know that the sampling rule collects information to discriminate θ with its closest alternatives, and eliminating a piece cannot make the resulting "optimal" proportions worse at this task. However, it could make them worse at discriminating θ with alternatives that are not the closest. This would imply that the elimination times for certain pieces could actually increase w.r.t. not eliminating at sampling. Theorem 3.2 guarantees that this does not happen: eliminating pieces at sampling cannot worsen our guarantees. We shall see in our experiments that eliminating pieces in both the sampling and stopping rules often yields improved sample complexity.

4 Experiments

Our experiments aim at addressing the following questions: (1) how do existing adaptive strategies behave when combined with elimination at stopping and (when possible) at sampling? How do they compare with native elimination-based methods? (2) What is the difference between selective and full elimination? (3) How do LLR and elimination stopping compare as a function of δ?[6]

[6] Our code is available at `https://github.com/AndreaTirinzoni/bandit-elimination`.

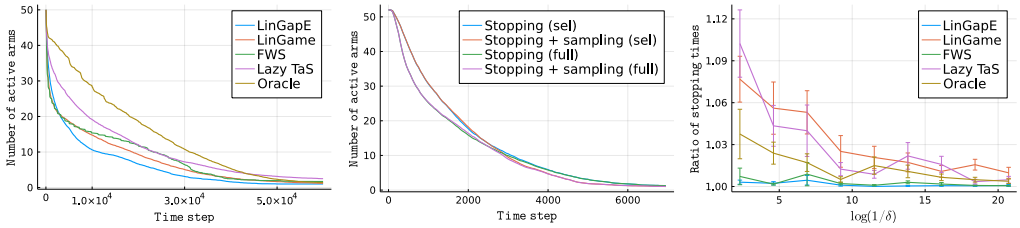

Figure 2: Experiments on linear instances with $K = 50$, $d = 10$, averaged over 100 runs, with the right plot showing standard deviations. (left) How different adaptive algorithms eliminate arms in BAI when using elimination stopping. (middle) LinGame on BAI when combined with full and selective elimination rules, either only at stopping or both at stopping and at sampling. (right) Ratio between the LLR and elimination stopping times of different algorithms as a function of $\log(1/\delta)$.

We ran experiments on two bandit structures: linear (where $d < K$) and unstructured (where $K = d$ and the arms are the canonical basis of $\mathbb{R}^d$). For each of them, we considered 3 pure exploration problems: BAI, Top-m, and online sign identification (OSI) [9, 27], also called thresholding bandits. All experiments use $\delta = 0.01$ and are averaged over 100 runs.

We combined adaptive algorithms which are natively based on LLR stopping with our elimination stopping rules and, whenever possible, we extended their sampling rule to use elimination. The selected baselines are the following. For linear BAI, LinGapE [14], LinGame [16], Frank-Wolfe Sampling (FWS) [17], Lazy Track-and-Stop (TaS) [24], XY-Adaptive [19], and RAGE [20] (the latter two are natively elimination based). For linear Top-m, m-LinGapE [28], MisLid [29], FWS, Lazy TaS[7], and LinGIFA [28]. For linear OSI, LinGapE[8], LinGame, FWS, and Lazy TaS. For unstructured instances linear algorithms are still applicable, and we further implemented LUCB [11], UGapE [8], and the Racing algorithm [18] for BAI and Top-m. We also tested an "oracle" sampling rule which uses the optimal proportions from the lower bound. Due to space constraints, we present only the results on linear structures. Those on unstructured problems can be found in Appendix G. The first experiments use randomly generated instances with $K = 50$ arms and dimension $d = 10$.

Comparison of elimination times. We analyze how different adaptive algorithms eliminate pieces when combined with selective elimination at stopping. To this purpose we focus on BAI, where the sets of pieces can be conveniently reduced to a set of active arms, those that are still likely to be the optimal one. Figure 2*(left)* shows how the set of active arms evolves over time for the 5 adaptive baselines. Notably, many arms are eliminated very quickly, with most baselines able to halve the set of active arms in the first 3000 steps. The problem size is quickly reduced over time. As we shall see in the last experiment, this will yield significant computational gains. We further note that the "oracle" strategy, which plays fixed proportions, seems the slowest at eliminating arms. The reason is that the optimal proportions from the lower bound focus on discriminating the "hardest" arms, while the extra randomization in adaptive rules might indeed eliminate certain "easier" arms sooner.

Full versus selective elimination. We combine the different algorithms with full and selective elimination, both at sampling and stopping. Due to space constrains, Figure 2*(middle)* shows the results only for LinGame (see Appendix G for the others). We note that full elimination seems faster at discarding arms in earlier steps, as we would expect theoretically. However, it never stops earlier than its selective counterpart. Moreover, its computational overhead is not advantageous. Overall, we concluded that our selective elimination rule is the best choice and we shall thus focus on it in the remaining. Finally, we remark that combining the sampling rule with elimination (no matter of what type) seems to discard arms faster in later steps, and could eventually make the algorithm stop sooner.

LLR versus elimination stopping. We now compare LLR and elimination stopping as a function of δ. We know from theory that both stopping rules allow to achieve asymptotic optimality. Hence for asymptotically optimal sampling rules the resulting stopping times with LLR and elimination should tend to the same quantity as $\delta \to 0$. Figure 2*(right)*, where we report the ratio between the LLR stopping time and the elimination one for different algorithms, confirms that this is the case. Some algorithms (LinGapE and FWS) seem to benefit less from elimination stopping than the others,

[7] Lazy TaS, while analyzed only for BAI, can be applied to any problem since it is a variant of Track-and-Stop.
[8] LinGapE was originally proposed only for BAI in [14], but its extension to OSI is trivial.

	Algorithm	No elimination (LLR)		Elim. stopping		Elim. stopping + sampling	
		Samples	Time	Samples	Time	Samples	Time
BAI	LinGapE	33.19 ± 8.7	0.23	33.11 ± 8.7	0.2	29.89 ± 8.6	$0.18(-22\%)$
	LinGame	45.34 ± 14.2	0.23	43.67 ± 13.4	0.21	32.49 ± 8.1	$0.18(-22\%)$
	FWS	42.26 ± 60.1	0.73	42.25 ± 60.1	0.7	32.62 ± 18.0	$0.45(-38\%)$
	Lazy TaS	76.33 ± 65.8	0.15	74.08 ± 65.8	0.13	64.48 ± 81.8	$0.12(-20\%)$
	Oracle	56.36 ± 9.1	0.05	55.36 ± 9.3	0.02		
	XY-Adaptive					87.08 ± 29.1	0.44
	RAGE					106.87 ± 30.7	0.02
Top-m ($m = 5$)	m-LinGapE	63.69 ± 11.1	0.56	63.48 ± 11.0	0.41	59.57 ± 9.4	$0.24(-57\%)$
	MisLid	87.77 ± 20.4	0.55	85.95 ± 20.5	0.4	69.58 ± 16.0	$0.25(-55\%)$
	FWS	78.28 ± 65.0	3.0	78.23 ± 65.0	2.85	77.79 ± 65.0	$0.97(-67\%)$
	Lazy TaS	161.43 ± 96.9	0.57	159.86 ± 96.9	0.43	146.06 ± 82.6	$0.36(-36\%)$
	Oracle	102.45 ± 16.1	0.2	101.53 ± 16.4	0.08		
	LinGIFA	58.31 ± 10.8	2.46	58.31 ± 10.8	2.33		
OSI	LinGapE	17.31 ± 2.3	0.22	17.29 ± 2.2	0.19	14.71 ± 2.0	$0.17(-23\%)$
	LinGame	23.77 ± 4.1	0.25	23.05 ± 3.9	0.21	14.87 ± 2.0	$0.19(-24\%)$
	FWS	15.26 ± 2.0	0.83	15.24 ± 2.0	0.81	14.99 ± 2.1	$0.56(-32\%)$
	Lazy TaS	35.11 ± 10.2	0.32	33.98 ± 9.7	0.3	23.51 ± 5.6	$0.24(-25\%)$
	Oracle	29.1 ± 4.8	0.06	28.65 ± 5.0	0.03		

Table 1: Experiments on linear instances with $K = 50$, $d = 20$. The "Time" columns report average times per iteration in milliseconds. The percentage in the last column is the change w.r.t. the time without elimination. Each entry reports the mean across 100 runs plus/minus standard deviation (which is omitted for compute times due to space constraints). Algorithms for which the third column is missing cannot be combined with elimination at sampling, while algorithms for which the first two columns are missing are natively elimination-based. Samples are scaled down by a factor 10^3.

i.e., they achieve smaller ratios of stopping times. We believe this to be a consequence of their mostly "greedy" nature, while the extra randomization of the other algorithms might help in this aspect.

Sample complexities and computation times. We finally compare our baselines in all three exploration tasks, in terms of sample complexity and computation time. For this experiment, we selected a larger linear instance with $K = 50$ and $d = 20$, randomly generated (see the protocol in Appendix G). From the results in Table 1, we highlight three points. (1) The computation times of all adaptive algorithms decrease when using selective elimination stopping instead of LLR and further decrease when also using elimination at sampling. In the case of Top-m (i.e., the hardest combinatorial problem), most adaptive algorithms become at least twice faster with elimination at stopping and sampling instead of LLR. (2) Elimination at sampling improves the sample complexity of all algorithms. (3) For BAI, the natively elimination-based algorithm RAGE, which updates its strategy infrequently, is the fastest in terms of computation time but the slowest in terms of samples. Adaptive algorithms using elimination achieve run times that are within an order of magnitude of those of RAGE, while outperforming it in terms of sample complexity by a factor 2 to 3.

5 Conclusion

We proposed a selective elimination rule, which successively prunes the pieces of the empirical answer, that can be easily combined with existing adaptive algorithms for general identification problems. We proved that it reduces their computational complexity, it never worsens their sample complexity guarantees, and it provably discards certain answers early. Our experiments on different pure exploration problems and bandit structures show that existing adaptive algorithms often benefit from a reduced sample complexity when combined with selective elimination, while achieving significant gains in computation time. Moreover, they show that selective elimination is overall better (in terms of samples vs time) than its full variant which repeatedly updates the pieces of all answers.

Interesting directions for future work include investigating whether better guarantees on the stopping time can be derived for algorithms combined with elimination as compared to their LLR counterparts, and designing adaptive algorithms which are specifically tailored for elimination.

References

[1] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[2] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Pac bounds for multi-armed bandit and markov decision processes. In *International Conference on Computational Learning Theory*, pages 255–270. Springer, 2002.

[3] Kevin Jamieson and Robert Nowak. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In *2014 48th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2014.

[4] Aurélien Garivier and Emilie Kaufmann. Optimal best arm identification with fixed confidence. In *Conference on Learning Theory*, pages 998–1027, 2016.

[5] Lijie Chen and Jian Li. On the optimal sample complexity for best arm identification. *arXiv preprint arXiv:1511.03774*, 2015.

[6] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *International conference on Algorithmic learning theory*, pages 23–37. Springer, 2009.

[7] Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. Best arm identification in multi-armed bandits. In *COLT*, pages 41–53. Citeseer, 2010.

[8] Victor Gabillon, Mohammad Ghavamzadeh, and Alessandro Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In *NIPS-Twenty-Sixth Annual Conference on Neural Information Processing Systems*, 2012.

[9] Alexandra Carpentier and Andrea Locatelli. Tight (lower) bounds for the fixed budget best arm identification bandit problem. In *Conference on Learning Theory*, pages 590–604. PMLR, 2016.

[10] Yasin Abbasi-Yadkori, Peter Bartlett, Victor Gabillon, Alan Malek, and Michal Valko. Best of both worlds: Stochastic & adversarial best-arm identification. In *Conference on Learning Theory*, pages 918–949. PMLR, 2018.

[11] Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. Pac subset selection in stochastic multi-armed bandits. In *ICML*, volume 12, pages 655–662, 2012.

[12] Emilie Kaufmann, Wouter Koolen, and Aurélien Garivier. Sequential test for the lowest mean: From thompson to murphy sampling. *arXiv preprint arXiv:1806.00973*, 2018.

[13] Rémy Degenne and Wouter M Koolen. Pure exploration with multiple correct answers. *arXiv preprint arXiv:1902.03475*, 2019.

[14] Liyuan Xu, Junya Honda, and Masashi Sugiyama. A fully adaptive algorithm for pure exploration in linear bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 843–851. PMLR, 2018.

[15] Rémy Degenne, Wouter M Koolen, and Pierre Ménard. Non-asymptotic pure exploration by solving games. In *Advances in Neural Information Processing Systems*, pages 14492–14501, 2019.

[16] Rémy Degenne, Pierre Ménard, Xuedong Shang, and Michal Valko. Gamification of pure exploration for linear bandits. In *International Conference on Machine Learning*, pages 2432–2442. PMLR, 2020.

[17] Po-An Wang, Ruo-Chun Tzeng, and Alexandre Proutiere. Fast pure exploration via frank-wolfe. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[18] Emilie Kaufmann and Shivaram Kalyanakrishnan. Information complexity in bandit subset selection. In *Conference on Learning Theory*, pages 228–251. PMLR, 2013.

[19] Marta Soare, Alessandro Lazaric, and Rémi Munos. Best-arm identification in linear bandits. *Advances in Neural Information Processing Systems*, 27:828–836, 2014.

[20] Tanner Fiez, Lalit Jain, Kevin G Jamieson, and Lillian Ratliff. Sequential experimental design for transductive linear bandits. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[21] Chao Tao, Saúl Blanco, and Yuan Zhou. Best arm identification in linear bandits with linear dimension dependency. In *International Conference on Machine Learning*, pages 4877–4886. PMLR, 2018.

[22] Shouyuan Chen, Tian Lin, Irwin King, Michael R Lyu, and Wei Chen. Combinatorial pure exploration of multi-armed bandits. *Advances in neural information processing systems*, 27:379–387, 2014.

[23] Mohammadi Zaki, Avi Mohan, and Aditya Gopalan. Explicit best arm identification in linear bandits using no-regret learners. *arXiv preprint arXiv:2006.07562*, 2020.

[24] Yassir Jedra and Alexandre Proutiere. Optimal best-arm identification in linear bandits. *arXiv preprint arXiv:2006.16073*, 2020.

[25] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

[26] Stefan Magureanu, Richard Combes, and Alexandre Proutiere. Lipschitz bandits: Regret lower bound and optimal algorithms. In *Conference on Learning Theory*, pages 975–999. PMLR, 2014.

[27] Reda Ouhamma, Rémy Degenne, Pierre Gaillard, and Vianney Perchet. Online Sign Identification: Minimization of the Number of Errors in Thresholding Bandits. In *NeurIPS 2021 - 35th International Conference on Neural Information Processing Systems*, NeurIPS 2021 - 35th International Conference on Neural Information Processing Systems, pages 1–25, Virtual, Canada, December 2021.

[28] Clémence Réda, Emilie Kaufmann, and Andrée Delahaye-Duriez. Top-m identification for linear bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 1108–1116. PMLR, 2021.

[29] Clémence Réda, Andrea Tirinzoni, and Rémy Degenne. Dealing with misspecification in fixed-confidence linear top-m identification. In *35th Conference on Neural Information Processing Systems*, Virtual, 2021.

[30] Andrea Tirinzoni, Matteo Pirotta, Marcello Restelli, and Alessandro Lazaric. An asymptotically optimal primal-dual incremental algorithm for contextual linear bandits. *Advances in Neural Information Processing Systems*, 33:1417–1427, 2020.

[31] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17(1):1–42, 2016.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] In appendix D and E.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In appendix G.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

Table of Contents

A Notation **15**

B Identification Problems **15**
B.1 Best-arm Identification 16
B.2 Top-m Identification 17
B.3 Thresholding Bandits 18

C Log-likelihood ratio in exponential families **19**
C.1 Log-likelihood ratio for Gaussian linear models 21
C.2 Beyond Gaussians 22

D Proofs of Section 2 **22**
D.1 Proof of Lemma 2.3 22
D.2 Proof of Theorem 2.4 23
D.3 Proof of Theorem 2.6 23
D.4 Proof of Theorem 2.7 26

E Proofs of Section 3 **26**
E.1 On Assumption E.1 27
E.2 Proof of Theorem 3.2 28

F Existing Algorithms Satisfy Assumption 2.5 and 3.1 **30**
F.1 Sampling Rules 30
F.2 Common Assumptions 30
F.3 Assumption 2.5 Holds 32
F.4 Assumption 3.1 Holds 33

G Experiment Details and Additional Results **37**
G.1 Reproducibility details 37
G.2 Bandit instances 37
G.3 Additional Results 38

H Examples **39**
H.1 Running example 39
H.2 An oracle strategy with elimination at stopping provably reduces the computational complexity w.r.t. LLR stopping 41
H.3 Low information regret with elimination at stopping provably discards many arms early 42

I Additional details **44**

A Notation

Symbol	Meaning
$[K] = \{1, 2, \dots, K\}$	Set of K arms
Δ_K	K-dimensional simplex
$d \in \mathbb{N}_{>0}$	Dimension of parameter space
$\mathcal{M} \subseteq \mathbb{R}^d$	Set of possible reward parameters
$\mathbb{P}_\theta$	Distribution of observations in bandit $\theta \in \mathcal{M}$
$\nu_k(\theta)$	Reward distribution of arm k in bandit $\theta \in \mathcal{M}$
$\mu_k(\theta) := \mathbb{E}_{x \sim \nu_k(\theta)}[x]$	Mean reward of arm k in bandit $\theta \in \mathcal{M}$
$\mathcal{I}$	Set of answers
$i^\star(\theta)$	Correct answer for bandit problem $\theta \in \mathcal{M}$
$\Lambda(i) := \{\lambda \in \mathcal{M} : i^\star(\lambda) \neq i\}$	Set of alternatives to answer $i \in \mathcal{I}$
$\mathcal{P}(i)$	Set of alternative piece indexes for answer $i \in \mathcal{I}$
$P_i := \lvert\mathcal{P}(i)\rvert$	Number of pieces for answer $i \in \mathcal{I}$
$\Lambda_p(i)$	Piece $p \in \mathcal{P}(i)$ for answer $i \in \mathcal{I}$
$X_{[t]} := (X_1^{k_1}, \dots, X_t^{k_t})$	Vector of t observations
$L_t(\theta, \lambda) := \log \frac{d\mathbb{P}_\theta}{d\mathbb{P}_\lambda}(X_{[t]})$	LLR of t observations between θ and λ
$\mathrm{KL}_k(\theta, \lambda) := \mathrm{KL}(\nu_k(\theta), \nu_k(\lambda))$	KL divergence between $\nu_k(\theta)$ and $\nu_k(\lambda)$
$\hat{\theta}_t := \arg\max_{\lambda \in \mathcal{M}} d\mathbb{P}_\lambda(X_{[t]})$	Maximum likelihood estimator for θ
$\hat{\mu}_t^k := \frac{1}{N_t^k} \sum_{s=1}^t X_s^{k_s} \mathbb{1}(k_s = k)$	Empirical mean of arm k (different from $\mu_k(\hat{\theta}_t)$)
$H_p(\omega, \theta) := \inf_{\lambda \in \Lambda_p(i^\star)} \sum_{k \in [K]} \omega^k \mathrm{KL}_k(\theta, \lambda)$	Information of $\omega \in \Delta_K$ for piece $\Lambda_p(i^\star)$
$H^\star(\theta) := \max_{\omega \in \Delta_K} \min_{p \in \mathcal{P}(i^\star)} H_p(\omega, \theta)$	Optimal constant from the lower bound (1)
$\Omega_\epsilon(\theta)$	Set of ϵ-optimal proportions

Table 2: The notation adopted in this paper.

B Identification Problems

In this section, we show that popular identification problems satisfy Assumption 2.1 and are thus suitable for elimination-based algorithms. We shall focus on *Gaussian linear bandits*, where, for any $\theta \in \mathcal{M}$ and $k \in [K]$, $\nu_k(\theta)$ is Gaussian with unit variance and linear mean $\mu_k(\theta) = \theta^T \phi_k$. For each identification problem, we first show how to decompose the sets of alternatives into pieces for which the closest alternatives can be found efficiently. Moreover, we report the closed-form equations for computing such closest alternatives in Gaussian linear bandits and in the special case of unstructured bandits (where $\mathcal{M} = \mathbb{R}^K$ and $\phi_k = e_k$, the canonical basis of $\mathbb{R}^k$). Finally, we show how to efficiently implement elimination strategies in each of these identification problems even when enumerating over all possible answers is intractable (e.g., for problems where the number of answers is exponential in the problem dimension).

The LLR in Gaussian linear bandits For all identification problems presented later, we need to show that $\inf_{\lambda \in \Lambda_p(i)} L_t(\hat{\theta}, \lambda)$ can be computed efficiently for any piece. In Gaussian linear bandits, such a log-likelihood ratio is actually equivalent to a KL divergence (see Corollary C.6),

$$L_t(\hat{\theta}, \lambda) = \sum_{k \in [K]} N_t^k \, \mathrm{KL}_k(\hat{\theta}_t, \lambda) = \frac{1}{2} \|\hat{\theta}_t - \lambda\|_{V_t}^2,$$

which in turn is a quadratic form weighted by the design matrix $V_t := \sum_{s=1}^t \phi_{k_s} \phi_{k_s}^T$. Therefore, with greater generality, in rest of this section we shall focus on showing that $\inf_{\lambda \in \Lambda_p(i)} \|\theta - \lambda\|_{V_N}^2$ can be computed efficiently for any $\theta, \lambda \in \mathcal{M}$, piece $\Lambda_p(i)$, and (positive-definite) matrix $V_N := \sum_{k \in [K]} N^k \phi_k \phi_k^T$ with $N \in \mathbb{R}_{\geq 0}^K$. In all cases, this will require minimizing quadratic forms over half-spaces.

B.1 Best-arm Identification

In BAI, the goal is to find the arm with largest mean. The set of answers is therefore $\mathcal{I} = [K]$ and the correct answer of $\theta \in \mathcal{M}$ is $i^\star(\theta) = \arg\max_{k\in[K]} \theta^T \phi_k$.

Decomposition into pieces For each $i \in \mathcal{I}$, the set of alternatives $\Lambda(i)$ can be decomposed into half-spaces,

$$\Lambda(i) = \bigcup_{k\in[K], k\neq i} \left\{\lambda \in \mathcal{M} : \lambda^T \phi_k > \lambda^T \phi_i\right\}.$$

Therefore, we can take $\mathcal{P}(i) = [K] \setminus \{i\}$ with $P_i = K - 1$ and $\Lambda_p(i) = \{\lambda \in \mathcal{M} : \lambda^T \phi_p > \lambda^T \phi_i\}$ for $p \in [K] \setminus \{i\}$.

Closest alternatives Take any $j, k \in [K]$ with $j \neq k$. For linear problems, for any $\theta \in \mathbb{R}^d$,

$$\inf_{\lambda\in\Lambda_j(k)} \|\theta - \lambda\|_N^2 = \begin{cases} \frac{(\theta^T(\phi_k - \phi_j))^2}{\|\phi_k - \phi_j\|^2_{V_N^{-1}}} & \text{if } \theta^T(\phi_k - \phi_j) \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

For the special case of unstructured problem, for any $\theta \in \mathbb{R}^K$,

$$\inf_{\lambda\in\Lambda_j(k)} \|\theta - \lambda\|_{V_N}^2 = \begin{cases} \frac{N_j N_k}{N_j + N_k}(\theta^T(\phi_k - \phi_j))^2 & \text{if } \theta^T(\phi_k - \phi_j) \geq 0 \text{ and } N_j + N_k > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Efficient implementation In BAI, for each answer (i.e., arm) $i \in \mathcal{I}$, the piece indexes $p \in \mathcal{P}(i)$ are themselves answers (different than i). Implementing the elimination stopping rule in its general form requires storing and iterating over $K(K-1)$ elements (all items in $\mathcal{P}(i)$ for each $i \in \mathcal{I}$). However, much better implementations exist that require storing at most K elements (one for each answer). Here we propose two such implementations: the first, for full elimination, is more statistically-efficient, while the second one (for selective elimination) is more computationally-efficient.

Full elimination (statistically-efficient implementation) Due to the structure of the problem, whenever we eliminate one piece $\Lambda_p(i)$, we actually know that the mean reward of arm p cannot be better than that of arm i. In other words, p cannot be the right answer. Therefore, we can maintain a list of active arms $\mathcal{I}_t$ which is initialized as $\mathcal{I}_0 = [K]$ and updated as

$$\mathcal{I}_t := \mathcal{I}_{t-1} \setminus \left\{ j \in \mathcal{I}_{t-1} \middle| \max_{i\neq j} \inf_{\lambda\in\Lambda_j(i)} L_t(\hat{\theta}_t, \lambda) \geq \beta_{t,\delta} \right\}.$$

Then, we stop whenever $|\mathcal{I}_t| = 1$ and return the single arm left active. Due to the inner maximization, this implementation requires performing $|\mathcal{I}_{t-1}|(K-1)$ minimizations over half-spaces at each step to check elimination.

Proposition B.1. *An algorithm using the statistically-efficient implementation above never discards pieces later than (and thus never stops later than) the full elimination rule of* (5) *almost surely.*

Proof. If the algorithm eliminates a piece $\Lambda_p(i)$ at time t with (5),

$$\beta_{t,\delta} \leq \inf_{\lambda\in\Lambda_p(i)} L_t(\hat{\theta}_t, \lambda) \leq \max_{j\neq p} \inf_{\lambda\in\Lambda_p(j)} L_t(\hat{\theta}_t, \lambda).$$

This implies that the elimination condition in the statistically efficient implementation triggers as well. □

Selective elimination (computationally-efficient implementation) An even simpler implementation is to check elimination, at each time t, only for pieces related to the empirical optimal arm $i^\star(\hat{\theta}_t)$. That is, we update $\mathcal{I}_t$ as

$$\mathcal{I}_t := \mathcal{I}_{t-1} \setminus \left\{ j \in \mathcal{I}_{t-1} \middle| \inf_{\lambda\in\Lambda_j(i^\star(\hat{\theta}_t))} L_t(\hat{\theta}_t, \lambda) \geq \beta_{t,\delta} \right\}.$$

This requires linear (in K) per-round memory and time complexity. Moreover, checking this stopping rule is more time-efficient than the LLR one. The latter requires to perform $K-1$ tests at each step, while the elimination one only performs $O(|\mathcal{I}_{t-1}|)$ tests at each round t, thus becoming faster as arms are eliminated.

Proposition B.2. *An algorithm using the computationally-efficient implementation above never discards pieces later than (and thus never stops later than) the selective elimination rule almost surely.*

Proof. If the algorithm eliminates a piece of the empirical optimal arm at time t using the selective elimination rule, the arm corresponding to that piece is also eliminated from the set $\mathcal{I}_t$ above. □

B.2 Top-m Identification

In top-m identification, the goal is to find the $m > 0$ arms with largest mean. BAI is therefore a special case of this problem when $m = 1$. The set of answers is $\mathcal{I} = \{\mathcal{S} \subseteq [K] : |\mathcal{S}| = m\}$ with size $|\mathcal{I}| = \binom{K}{m}$ and the correct answer of $\theta \in \mathcal{M}$ is $i^\star(\theta) = \arg\max^m_{k\in[K]} \theta^T\phi_k$, where we use $\arg\max^m : \mathbb{R}^K \mapsto \binom{K}{m}$ to denote the function returning the set of m largest values.

Decomposition into pieces Let us denote each $i \in \mathcal{I}$ as a tuple $i = (k_1, \dots, k_m)$ of m arms. Similarly to BAI, it is known that the set of alternatives $\Lambda(i)$ can be decomposed into half-spaces [29],

$$\Lambda(i) = \bigcup_{j\in i, k\in[K]\setminus i} \left\{\lambda \in \mathcal{M} : \lambda^T\phi_k > \lambda^T\phi_j\right\}.$$

Therefore, we have $\mathcal{P}(i) = i \times ([K] \setminus i)$ with $P_i = m(K-m)$ and $\Lambda_p(i) = \{\lambda \in \mathcal{M} : \lambda^T\phi_k > \lambda^T\phi_j\}$ when $p = (j, k)$.

Closest alternatives Note that each set $\Lambda_p(i)$ is still a half-space of the same form as the one we have for BAI. Hence, the same closed form expression for the closest alternative derived for BAI can be adopted for top-m identification (see the closed-form expressions in the previous section).

Efficient implementation Note that, differently from BAI, here the set of answers is of combinatorial size. It is therefore intractable to store and enumerate all sets of active pieces $\mathcal{P}_t(i)$. However, thanks to the structure of the problem, this is not necessary and there exists an efficient implementation for the elimination stopping rule. First note that, while there are $m(K-m)$ pieces for each of $\binom{K}{m}$ possible answers, the total number of half-spaces is only $K(K-1)$, one for each couple of different arms. With some abuse of notation, let us denote by $\Lambda_{k,j} := \left\{\lambda \in \mathcal{M} : \lambda^T\phi_k > \lambda^T\phi_j\right\}$ the half-space associated with arms k and j. The elimination stopping rule, which checks whether all the pieces in $\mathcal{P}(i)$ for some answer i have been discarded, is equivalent to checking whether there exist m arms $k_1, \dots, k_m$ such that Λ_{k,k_l} has been eliminated for all $k \notin \{k_1, \dots, k_m\}$ and $l \in [m]$. Therefore, in our implementations we will only store whether each half-space Λ_{k,k_l} has been eliminated or not. As before, we now see two possible implementations, one more computationally efficient and the other more statistically efficient.

Full elimination (statistically-efficient implementation) The idea is to check, at each time step t, the elimination condition for all half-spaces which have not been previously discarded. In particular, for each arm $j \in [K]$ we keep a set $\mathcal{S}_t(j)$ storing those arms which are "worse" than j. Formally, we initially set $\mathcal{S}_0(j) = \emptyset$ and update it as

$$\mathcal{S}_t(j) := \begin{cases} \mathcal{S}_{t-1}(j) \cup \left\{k \notin \mathcal{S}_{t-1}(j) \cup \{j\} \,\middle|\, \inf_{\lambda\in\Lambda_{k,j}} L_t(\hat{\theta}_t, \lambda) \geq \beta_{t,\delta}\right\} & \text{if } |\mathcal{S}_{t-1}(j)| < K-m, \\ \mathcal{S}_{t-1}(j) & \text{otherwise.} \end{cases}$$

That is, when a half-space $\Lambda_{k,j}$ is eliminated, we conclude that arm k is "worse" than arm j and thus add the former to $\mathcal{S}_t(j)$. In order to decide when to stop, we use the following intuition: whenever we find that $|\mathcal{S}_t(j)| \geq K-m$ for some arm $j \in [K]$, then we know that j must be in the top-m arms of θ and we can thus stop updating the set $\mathcal{S}_t(j)$. Therefore, we can stop whenever there exist m

arms satisfying this property. This can be checked efficiently by keeping track of how many arms reach the condition $|\mathcal{S}_t(j)| \geq K - m$ and stopping when the number of such arms reaches m. This approach takes $O(K(K-m))$ memory in the worst-case to store the sets $\mathcal{S}_t(j)$. At each step, it performs exactly $\sum_{j:|\mathcal{S}_{t-1}(j)|<K-m}(K - |\mathcal{S}_{t-1}(j)| - 1)$ minimizations over half-spaces to check the elimination conditions, which gives $O(K(K-1))$ time complexity in the worst-case.

Proposition B.3. *An algorithm using the statistically-efficient implementation above never discards pieces later than (and thus never stops later than) the full elimination rule of* (5) *almost surely.*

Proof. Note that the elimination condition for single half-spaces is exactly the same in the general elimination rule of (5) and in its implementation above. If (5) eliminates a piece $\Lambda_p(i)$ at time t, this implies some half-space $\Lambda_{k,j}$ is eliminated. Then, we have two possible cases: if $|\mathcal{S}_{t-1}(j)| < K - m$, then we have $k \in \mathcal{S}_t(j)$ by the condition above, i.e., k is detected as "worse" than j and it will be never checked again. On the other hand, if $|\mathcal{S}_{t-1}(j)| \geq K - m$, then j has already been labeled as belonging to the final answers. Thus, no minimization over its corresponding half-spaces (including the one for k) will be checked anymore, which is the same as saying that $\Lambda_{k,j}$ has already been eliminated. □

Selective elimination (computationally-efficient implementation) Similarly to what we did for BAI, the most computationally-efficient implementation consists in checking the elimination condition only for the alternative pieces (i.e., the half-spaces) of the empirical correct answer at each step. We modify the update rule of the statistically-efficient implementation as

$$\mathcal{S}_t(j) := \mathcal{S}_{t-1}(j) \cup \left\{ k \notin i^\star(\hat{\theta}_t) \cup \mathcal{S}_{t-1}(j) \middle| \inf_{\lambda \in \Lambda_{k,j}} L_t(\hat{\theta}_t, \lambda) \geq \beta_{t,\delta} \right\}$$

if $j \in i^\star(\hat{\theta}_t)$ and $|\mathcal{S}_{t-1}(j)| < K - m$, and $S_t(j) := \mathcal{S}_{t-1}(j)$. That is, at each step we only check elimination for half-spaces associated with the top-m arms of $\hat{\theta}_t$, excluding those that have already been eliminated and those that have already reached the threshold for being among the final answer. Note that this implementation performs $\sum_{j \in i^\star(\hat{\theta}_t), |\mathcal{S}_{t-1}(j)|<K-m}(K - |i^\star(\hat{\theta}_t) \cup \mathcal{S}_{t-1}(j)|) \leq m(K-m)$ minimizations over half-spaces at each step t. In constrast, the LLR stopping rule always performs $m(K-m)$ minimizations and is thus less efficient.

Proposition B.4. *An algorithm using the computationally-efficient implementation above never discards pieces later than (and thus never stops later than) the selective elimination rule almost surely.*

Proof. The proof is the same as for BAI: if an arm is discarded by the selective elimination rule, then it is also discarded from the sets above. □

B.3 Thresholding Bandits

In the thresholding bandit problem, the goal is to learn whether the mean of each arm is above or below some given threshold. As usual, without loss of generality, we shall take zero as our threshold, for which the problem reduces to learning the sign of the mean reward of each arm. Let $\text{sign}(x) := \mathbb{1}(x \geq 0)$. Then, the set of answers is $\mathcal{I} = \{0,1\}^K$ with size $|\mathcal{I}| = 2^K$. The correct answer of problem $\theta \in \mathcal{M}$ is $i^\star(\theta) = (\text{sign}(\theta^T \phi_k))_{k \in [K]}$.

Decomposition into pieces For each $i \in \{0,1\}^K$ (represented as a K-dimensional binary vector), the set of alternatives $\Lambda(i)$ can be decomposed into pieces as

$$\Lambda(i) = \bigcup_{k \in [K]} \left\{ \lambda \in \mathcal{M} : \text{sign}(\lambda^T \phi_k) \neq i^k \right\}.$$

Therefore, we have $\mathcal{P}(i) = [K]$ with $P_i = K$ and $\Lambda_p(i) = \{\lambda \in \mathcal{M} : \text{sign}(\lambda^T \phi_p) \neq i^p\}$. As for BAI, the computation of the closest alternative over such pieces can be performed efficiently.

Closest alternatives Let $\theta \in \mathcal{M}$ and $N \in \mathbb{R}^K_{\geq 0}$. The computation of the closest alternatives over pieces $\Lambda_p(i)$ can be reduced to the following optimization problem. For any arm $k \in [K]$ and any $b \in \{0, 1\}$, we need to find

$$\inf_{\lambda \in \mathcal{M}: \mathrm{sign}(\lambda^T \phi_k) \neq b} \|\theta - \lambda\|^2_{V_N}.$$

It is easy to see that this is zero when $b \neq \mathrm{sign}(\theta^T \phi_k)$ (since θ itself is feasible). In case $b = \mathrm{sign}(\theta^T \phi_k)$, for unstructured problems ($\mathcal{M} = \mathbb{R}^K$), the solution is to take λ equal to θ at all components except the k-th one, where it is set to zero. This gives

$$\inf_{\lambda \in \mathbb{R}^K: \mathrm{sign}(\lambda^T \phi_k) \neq b} \|\theta - \lambda\|^2_{V_N} = \begin{cases} N^k (\theta^T \phi_k)^2 & \text{if } \mathrm{sign}(\theta^T \phi_k) = b, \\ 0 & \text{otherwise.} \end{cases}$$

In the linear case, again under the assumption that V_N is positive definite, we get

$$\inf_{\lambda \in \mathbb{R}^d: \mathrm{sign}(\lambda^T \phi_k) \neq b} \|\theta - \lambda\|^2_{V_N} = \begin{cases} \frac{(\theta^T \phi_k)^2}{\|\phi_k\|^2_{V_N^{-1}}} & \text{if } \mathrm{sign}(\theta^T \phi_k) = b, \\ 0 & \text{otherwise.} \end{cases}$$

Efficient implementation For this problem, implementing the general elimination stopping rule would require storing and iterating over $2^K K$ pieces, which is clearly intractable. However, this problem introduces a high redundancy in the alternative pieces that we can exploit for an efficient implementation which takes only linear (in K) time and space. Differently from BAI and top-m identificaiton, the procedure highlighted below is *exactly* an implementation of the "theoretical" elimination rules presented in the main paper, with the full and selective elimination rules reducing to the same thing.

Note that, for any $p \in [K]$ and $i, j \in \{0, 1\}^K$ such that $i^p = j^p$, we have $\Lambda_p(i) = \Lambda_p(j)$. That is, whenever we eliminate some piece $\Lambda_p(i)$ for $p \in [K]$ and $i \in \{0, 1\}^K$, we actually eliminate all problems in $\mathcal{M}$ whose sign of the p-th mean reward is different from i^p. In other words, we learn that the p-th position of the correct answer for θ is indeed i^p. Therefore, an efficient implementation is as follows: we keep a set $\mathcal{A}_t$ of active arms (those for which we still have to learn the corresponding component in the correct answer). This set is initialized as $\mathcal{A}_0 = [K]$ and updated as

$$\mathcal{A}_t := \mathcal{A}_{t-1} \setminus \left\{ j \in \mathcal{A}_{t-1} \middle| \max_{i \in \{0,1\}^K} \inf_{\lambda \in \Lambda_j(i)} L_t(\hat{\theta}_t, \lambda) \geq \beta_{t,\delta} \right\}.$$

While the maximization over 2^K elements might appear intractable, the structure of the problem allows us to entirely avoid it. Note that, for fixed j, the sets $\Lambda_j(i)$ are fully specified by the j-th component of i. Moreover, the inf is zero whenever $i^j \neq \mathrm{sign}(\hat{\theta}_t^T \phi_j)$ since that would imply $\hat{\theta}_t \in \Lambda_j(i)$. Therefore, the elimination condition can be equivalently rewritten in the convenient form

$$\mathcal{A}_t := \mathcal{A}_{t-1} \setminus \left\{ j \in \mathcal{A}_{t-1} \middle| \inf_{\lambda \in \mathcal{M}: \mathrm{sign}(\lambda^T \phi_j) \neq \mathrm{sign}(\hat{\theta}_t^T \phi_j)} L_t(\hat{\theta}_t, \lambda) \geq \beta_{t,\delta} \right\}.$$

Moreover, whenever the elimination condition above triggers for some arm $j \in [K]$, we set a variable $S_j := \mathrm{sign}(\hat{\theta}_t^T \phi_j)$ with the correct sign for the j-th component. We stop whenever $\mathcal{A}_t = \emptyset$ (i.e., when all signs have been learned) and return $\hat{i} := (S_1, S_2, \ldots, S_K)$. Similarly to BAI, this requires to perform only $|\mathcal{A}_{t-1}|$ tests at each step t. On the other hand, the LLR stopping rule would perform K tests at each step.

C Log-likelihood ratio in exponential families

We suppose in this section that all arms have distributions in a one-parameter exponential family (the same for all arms, for simpler notations). An arm distribution can thus be described by any one of three parameters: the arm feature vector $\phi_k \in \mathbb{R}^d$, the arm mean $\mu_k(\theta) = \phi_k^\top \theta$ and its natural parameter $\eta_k(\theta)$. These two last are functions of the model θ. For two models θ and λ, let f be a function such that the KL between the arm distributions with those parameters is $d_f(\eta_k(\lambda), \eta_k(\theta))$, where d_f is the Bregman divergence associated to f. Let f^* be the convex conjugate of f. The

Kullback-Leibler divergence between the arm distributions under models θ and λ is also equal to $d_{f^*}(\mu_k(\theta), \mu_k(\lambda))$.

If $\eta_k(\theta)$ is the natural parameter of that arm k, we have $\mu_k(\theta) = f'(\eta_k(\theta))$, and since $(f^*)' = (f')^{-1}$ we have $\eta_k(\theta) = (f^*)'(\mu_k(\theta))$.

Lemma C.1. *For all $\theta, \lambda \in \mathcal{M}$, the quantity $L_t(\theta, \lambda) - \sum_{k=1}^K N_t^k \operatorname{KL}_k(\theta, \lambda)$ is a martingale if the observations come from the model θ. This does not depend on the hypothesis that the distributions belong to an exponential family but only requires $\mathbb{P}_\theta \ll \mathbb{P}_\lambda$.*

Proof. We can expand the LLR to obtain a sum over times and write the KL as an expected log-likelihood ratio,

$$L_t(\theta, \lambda) - \sum_{k=1}^K N_t^k \operatorname{KL}_k(\theta, \lambda) = \sum_{s=1}^t \left(\log \frac{d\mathbb{P}_\theta}{d\mathbb{P}_\lambda}(X_s^{k_s}) - \mathbb{E}_{X \sim \nu_{k_s}(\theta)} \left[\log \frac{d\mathbb{P}_\theta}{d\mathbb{P}_\lambda}(X) \right] \right) .$$

The martingale property is then immediate. □

Many of our proofs depend on the informal statement that the martingale $L_t(\theta, \lambda) - \sum_{k=1}^K N_t^k \operatorname{KL}_k(\theta, \lambda)$ concentrates, and is thus a lower order term which is negligible for t large enough.

Lemma C.2. *For all θ, λ and all $X_{[t]}$,*

$$L_t(\theta, \lambda) = \sum_{k=1}^K N_t^k \operatorname{KL}_k(\theta, \lambda) - \sum_{k=1}^K N_t^k (\eta_k(\lambda) - \eta_k(\theta))(\hat{\mu}_{t,k} - \mu_k(\theta)) .$$

Proof. Write $\log \frac{d\mathbb{P}_\theta}{d\mathbb{P}_\lambda}(X) = \eta(\theta)X - f(\eta(\theta)) - (\eta(\lambda)X - f(\eta(\lambda)))$ and develop the Bregman divergence (the KL) on the right. □

Lemma C.3. *If the distribution of arm k for model θ is σ^2-sub-Gaussian, then for all λ,*

$$\frac{1}{2\sigma^2}(\mu_k(\lambda) - \mu_k(\theta))^2 \leq \operatorname{KL}_k(\lambda, \theta) ,$$

$$\frac{1}{2\sigma^2} \sum_{k=1}^K N_t^k (\mu_k(\hat{\theta}_t) - \mu_k(\theta))^2 \leq L_t(\hat{\theta}_t, \theta)$$

Proof. We first prove that the sub-Gaussian hypothesis is equivalent to both these inequalities:

$$\forall \lambda, d_f(\eta_k(\lambda), \eta_k(\theta)) \leq \frac{1}{2}\sigma^2(\eta_k(\lambda) - \eta_k(\theta))^2 ,$$

$$\forall \lambda, d_{f^*}(\mu_k(\lambda), \mu_k(\theta)) \geq \frac{1}{2\sigma^2}(\mu_k(\lambda) - \mu_k(\theta))^2 .$$

The first result is then a simple consequence of that second inequality and the equality $\operatorname{KL}(\lambda, \theta) = d_{f^*}(\mu_k(\lambda), \mu_k(\theta))$. The second result of the lemma can be obtained by applying the first one to $\lambda = \hat{\theta}_t$ for all arms, then summing over arms.

The cumulant generating function at parameter $\eta_k(\theta)$ is $\xi \mapsto d_f(\eta_k(\theta) + \xi, \eta_k(\theta))$. The sub-Gaussian hypothesis is that this function is lower than $\frac{1}{2}\sigma^2\xi^2$. For the second inequality, we first remark that the convex conjugate of $\xi \mapsto d_f(\eta_k(\theta) + \xi, \eta_k(\theta))$ is $x \mapsto d_{f^*}(\mu_k(\theta) + x, \mu_k(\theta))$, and write

$$\begin{aligned} d_{f^*}(\mu_k(\lambda), \mu_k(\theta)) &= \sup_\xi \xi(\mu_k(\lambda) - \mu_k(\theta)) - d_f(\eta_k(\theta) + \xi, \eta_k(\theta)) \\ &\geq \sup_\xi \xi(\mu_k(\lambda) - \mu_k(\theta)) - \frac{1}{2}\sigma^2\xi^2 \\ &= \frac{1}{2\sigma^2}(\mu_k(\lambda) - \mu_k(\theta))^2 . \end{aligned}$$

□

Corollary C.4. *If the distribution of arm k for model θ is σ^2-sub-Gaussian, then for all λ,*

$$\left|\sum_{k=1}^{K} N_t^k(\eta_k(\lambda)-\eta_k(\theta))(\mu_k(\hat{\theta}_t)-\mu_k(\theta))\right| \le 2\sqrt{L_t(\hat{\theta}_t,\theta)}\sqrt{\sum_{k=1}^{K} N_t^k \frac{1}{2}\sigma^2(\eta_k(\lambda)-\eta_k(\theta))^2}\,.$$

Remark: for Gaussians with variance σ^2, we also have $\sum_{k=1}^{K} N_t^k \frac{1}{2}\sigma^2(\eta_k(\lambda)-\eta_k(\theta))^2 = \sum_{k=1}^{K} N_t^k \operatorname{KL}_k(\theta,\lambda)$. But that is not the case in general, and the sub-Gaussian assumption tells us that the sum of squares is larger than a KL, while we would like the reverse inequality.

Proof. Apply the Cauchy-Schwarz inequality, then Lemma C.3:

$$\begin{aligned}\left|\sum_{k=1}^{K} N_t^k(\eta_k(\lambda)-\eta_k(\theta))(\mu_k(\hat{\theta}_t)-\mu_k(\theta))\right| &\le \sqrt{\sum_{k=1}^{K} N_t^k(\eta_k(\lambda)-\eta_k(\theta))^2 \sum_{k=1}^{K} N_t^k(\mu_k(\hat{\theta}_t)-\mu_k(\theta))^2} \\ &\le 2\sqrt{L_t(\hat{\theta}_t,\theta)}\sqrt{\sum_{k=1}^{K} N_t^k \frac{1}{2}\sigma^2(\eta_k(\lambda)-\eta_k(\theta))^2}\,.\end{aligned}$$

□

C.1 Log-likelihood ratio for Gaussian linear models

Lemma C.5. *For any $\theta \in \mathcal{M}$ and $k \in [K]$, let $\nu_k(\theta)$ be Gaussian with unit variance and linear mean $\mu_k(\theta) = \theta^T\phi_k$. Then for any $\theta, \lambda \in \mathcal{M}$, any $t > 0$ and any sequence of observations,*

$$L_t(\theta,\lambda) = \sum_{k\in[K]} N_t^k \operatorname{KL}_k(\theta,\lambda) - (\lambda-\theta)^\top V_t(\hat{\theta}_t-\theta)\,.$$

Proof. Apply Lemma C.2 to the Gaussian case, where $\eta(\theta) = \mu(\theta) = \phi_k^\top\theta$.

$$L_t(\theta,\lambda) = \sum_{k\in[K]} N_t^k \operatorname{KL}_k(\theta,\lambda) - \sum_{k=1}^{K} N_t^k(\phi_k^\top\lambda-\phi_k^\top\theta)(\hat{\mu}_{t,k}-\phi_k^\top\theta)$$

For Gaussian linear models, we have

$$\sum_{k=1}^{K} N_t^k(\phi_k^\top\lambda-\phi_k^\top\theta)\phi_k^\top\hat{\theta}_t = (\lambda-\theta)^\top V_{N_t}\hat{\theta}_t = (\lambda-\theta)^\top\sum_{k=1}^{K} N_t^k\hat{\mu}_{t,k}\phi_k = \sum_{k=1}^{K} N_t^k(\phi_k^\top\lambda-\phi_k^\top\theta)\hat{\mu}_{t,k}\,.$$

We can use this to replace the sum involving $\hat{\mu}_{t,k}$ in the expression of $L_t(\theta,\lambda)$ by one involving $\hat{\theta}_t$.

$$\begin{aligned}L_t(\theta,\lambda) &= \sum_{k\in[K]} N_t^k \operatorname{KL}_k(\theta,\lambda) - (\lambda-\theta)^\top(\sum_{k=1}^{K} N_t^k\phi_k\phi_k^\top)(\hat{\theta}_t-\theta) \\ &= \sum_{k\in[K]} N_t^k \operatorname{KL}_k(\theta,\lambda) - (\lambda-\theta)^\top V_t(\hat{\theta}_t-\theta)\,.\end{aligned}$$

□

Corollary C.6. *For the linear Gaussian model of Lemma C.5, for any $\lambda \in \mathcal{M}$,*

$$L_t(\hat{\theta}_t,\lambda) = \sum_{k\in[K]} N_t^k \operatorname{KL}_k(\hat{\theta}_t,\lambda).$$

Lemma C.7. *For any $\theta \in \mathcal{M}$ and $k \in [K]$, let $\nu_k(\theta)$ be Gaussian with unit variance and linear mean $\mu_k(\theta) = \theta^T\phi_k$. Then, for any $\lambda \in \mathcal{M}$ and $t > 0$,*

$$\left(\sqrt{\sum_{k\in[K]} N_t^k \operatorname{KL}_k(\theta,\lambda)} - \sqrt{L_t(\hat{\theta}_t,\theta)}\right)^2 \le L_t(\hat{\theta}_t,\lambda) \le \left(\sqrt{\sum_{k\in[K]} N_t^k \operatorname{KL}_k(\theta,\lambda)} + \sqrt{L_t(\hat{\theta}_t,\theta)}\right)^2.$$

Proof. We decompose the LLR as

$$\begin{aligned} L_t(\hat{\theta}_t, \lambda) &= L_t(\theta, \lambda) + L_t(\hat{\theta}_t, \theta) \\ &= \sum_{k \in [K]} N_t^k \operatorname{KL}_k(\theta, \lambda) - (\lambda - \theta)^\top V_t(\hat{\theta}_t - \theta) + L_t(\hat{\theta}_t, \theta) \end{aligned}$$

The second term is bounded by Corollary C.4 (with $\sigma^2 = 1$) and the remark below it. We get

$$\begin{aligned} L_t(\hat{\theta}_t, \lambda) &\leq \sum_{k \in [K]} N_t^k \operatorname{KL}_k(\theta, \lambda) + 2\sqrt{L_t(\hat{\theta}_t, \theta)}\sqrt{\sum_{k=1}^K N_t^k \operatorname{KL}_k(\theta, \lambda)} + L_t(\hat{\theta}_t, \theta) \\ &= \left(\sqrt{\sum_{k=1}^K N_t^k \operatorname{KL}_k(\theta, \lambda)} + \sqrt{L_t(\hat{\theta}_t, \theta)}\right)^2 . \end{aligned}$$

The proof of the lower bound is similar. □

C.2 Beyond Gaussians

While the proofs in the next two sections are specialized to Gaussian rewards, it is possible to extend them to more exponential families under slight assumptions, similarly to what was done in [15]. If the arm distributions are known to belong to a σ^2-sub-Gaussian exponential family, with the additional restriction that the distribution parameters should belong to a compact subset of the open interval on which the family is defined, then there exists a constant c such that

$$\frac{1}{\sigma^2} \sum_{k=1}^K N_t^k \operatorname{KL}_k(\theta, \lambda) \leq \sum_{k=1}^K N_t^k \frac{1}{2}(\eta_k(\lambda) - \eta_k(\theta))^2 \leq c \sum_{k=1}^K N_t^k \operatorname{KL}_k(\theta, \lambda) .$$

And $\sum_{k=1}^K N_t^k \operatorname{KL}_k(\theta, \lambda)$ is also close to $\|\lambda - \theta\|_{V_t}^2$, up to constant factors.

We can then recover bounds on the likelihood ratio of the same shape as in Lemma C.7, up to constant factors depending on c and σ^2. The proofs of Appendix D and Appendix E then proceed similarly, up to the additional constants.

D Proofs of Section 2

D.1 Proof of Lemma 2.3

Proof. Fix any $\theta \in \mathcal{M}$ and let $\hat{i}$ be the answer returned by the algorithm at the stopping time τ. Then,

$$\begin{aligned} \mathbb{P}\left(\hat{i} \neq i^\star(\theta)\right) &\overset{(a)}{\leq} \mathbb{P}\left(\exists i \neq i^\star(\theta) : \mathcal{P}_\tau(i) = \emptyset\right) \\ &\overset{(b)}{\leq} \mathbb{P}\left(\exists i \neq i^\star(\theta), \forall p \in \mathcal{P}(i), \exists t \leq \tau : \inf_{\lambda \in \Lambda_p(i)} L_t(\hat{\theta}_t, \lambda) \geq \beta_{t,\delta}\right) \\ &\overset{(c)}{\leq} \mathbb{P}\left(\exists i \neq i^\star(\theta), \forall p \in \mathcal{P}(i), \exists t \geq 1 : \inf_{\lambda \in \Lambda_p(i)} L_t(\hat{\theta}_t, \lambda) \geq \beta_{t,\delta}\right) \end{aligned}$$

where (a) is from the definition of stopping rule, (b) is from the fact that, if $\mathcal{P}_\tau(i)$ is empty, then all the pieces of i have been eliminated at some times before τ, and (c) follows trivially by relaxing the condition $t \leq \tau$ to $t \geq 1$. Take any $t \geq 1$. Now note that, for any wrong answer $i \neq i^\star(\theta)$, $\theta \in \Lambda(i)$. By definition of the decomposition into pieces of $\Lambda(i)$, this means that there exists $\bar{p}_i \in \mathcal{P}(i)$ such that $\theta \in \Lambda_{\bar{p}_i}(i)$. Therefore, continuing the chain of inequalities above, we get that

$$\begin{aligned} \mathbb{P}\left(\hat{i} \neq i^\star(\theta)\right) &\overset{(d)}{\leq} \mathbb{P}\left(\exists i \neq i^\star(\theta), \exists t \geq 1 : \inf_{\lambda \in \Lambda_{\bar{p}_i}(i)} L_t(\hat{\theta}_t, \lambda) \geq \beta_{t,\delta}\right) \\ &\overset{(e)}{\leq} \mathbb{P}\left(\exists t \geq 1 : L_t(\hat{\theta}_t, \theta) \geq \beta_{t,\delta}\right) \overset{(f)}{\leq} \delta, \end{aligned}$$

where (d) holds since the event under which all pieces for i have been eliminated implies that $\bar{p}_i$ has been eliminated as well, (e) holds since $\theta \in \Lambda_{\bar{p}_i}(i)$, and (f) is from the assumption on threshold $\beta_{t,\delta}$. This concludes the proof. □

D.2 Proof of Theorem 2.4

Theorem 2.4 was proved in the main text. We prove here a more general result about the monotonicity of the stopping time with respect to the piece decomposition. We show that if two algorithms use the same sampling rules and use elimination stopping based on different piece decompositions, the algorithm using the finer decomposition (in the sense defined below) will stop earlier.

Definition D.1. *We say that a piece decomposition* $(\Lambda_p(i))_{i\in\mathcal{I},p\in\mathcal{P}(i)}$ *is finer than another one* $(\tilde{\Lambda}_p(i))_{i\in\mathcal{I},p\in\tilde{\mathcal{P}}(i)}$ *if for all* $i \in \mathcal{I}, p \in \tilde{\mathcal{P}}(i)$*, there exists a set* $S \subseteq \mathcal{P}(i)$ *such that* $\tilde{\Lambda}_p(i) = \bigcup_{q\in S} \Lambda_q(i)$.

Theorem D.2. *Let* $\mathcal{D} = (\Lambda_p(i))_{i\in\mathcal{I},p\in\mathcal{P}(i)}$ *be a finer decomposition than* $\tilde{\mathcal{D}} = (\tilde{\Lambda}_p(i))_{i\in\mathcal{I},p\in\tilde{\mathcal{P}}(i)}$*. For* $i \in \mathcal{I}, \tilde{p} \in \tilde{\mathcal{P}}(i)$*, let* $\tau_{\tilde{p}}$ *and* $\tilde{\tau}_{\tilde{p}}$ *be the times at which* $\tilde{\Lambda}_{\tilde{p}}(i)$ *is eliminated by the two corresponding algorithms (in the sense of Definition 2.2). Then almost surely* $\tau_{\tilde{p}} \leq \tilde{\tau}_{\tilde{p}}$*.*

Roughly, if a piece in the tilde decomposition corresponds to several pieces in the other, then it is faster to eliminate it as several pieces than as one piece.

Proof. It is enough to prove that whenever the elimination stopping rule of $\tilde{\Lambda}_{\tilde{p}}(i)$ triggers for $\tilde{\mathcal{D}}$, it triggers for $\mathcal{D}$ too. Then, let $t \geq 1$ and suppose that

$$\inf_{\lambda\in\Lambda_{\tilde{p}}(i)} L_t(\hat{\theta}_t, \lambda) \geq \beta_{t,\delta}.$$

Let $S_{\tilde{p}} \subseteq \mathcal{P}(i)$ be the set corresponding to $\tilde{p}$ in the definition of "finer decomposition". We first argue that

$$\inf_{\lambda\in\Lambda_p(i)} L_t(\hat{\theta}_t, \lambda) = \inf_{q\in S_p} \inf_{\lambda\in\Lambda_q(i)} L_t(\hat{\theta}_t, \lambda)\ .$$

Indeed, this is simply writing the infimum over a union as the infimum of an infimum. Hence if the $\tilde{p}$ piece is eliminated in $\tilde{\mathcal{D}}$, we have

$$\inf_{q\in S_p} \inf_{\lambda\in\Lambda_q(i)} L_t(\hat{\theta}_t, \lambda) \geq \beta_{t,\delta}$$

and every piece $\Lambda_p(i)$ for $p \in S_{\tilde{p}}$ is eliminated as well in $\mathcal{D}$. We get that the set $\tilde{\Lambda}_{\tilde{p}}$ is also eliminated in $\mathcal{D}$. □

Corollary: the finest possible decomposition is the one in which $\mathcal{P}(i) = \Lambda(i)$ and $\Lambda_p(i) = \{p\}$, i.e. every point of $\Lambda(i)$ is its own piece. This is not a computationally usable decomposition, but its the theoretically best one for the sample complexity metric (for a fixed sampling rule).

Theorem 2.4 for full elimination compared to LLR stopping follows from Theorem D.2 by setting $\tilde{\mathcal{P}}(i) = \{0\}$ and $\tilde{\Lambda}_p(i) = \Lambda(i)$. Then the elimination stopping rule uses a finer decomposition than the LLR stopping rule.

D.3 Proof of Theorem 2.6

We first present two important lemmas which will be used to prove the main statement for full elimination and selective elimination, respectively. For full elimination, the following lemma shows that if a piece has not been eliminated the information collected by the algorithm about it must be small.

Lemma D.3. *Consider an algorithm that uses the full elimination stopping rule* (5)*. Let* $p \in \mathcal{P}(i^\star(\theta))$*, then, for each time* t *such that* E_t *(Equation 6) holds and* $p \in \mathcal{P}_t(i^\star(\theta))$*,*

$$\sqrt{\inf_{\lambda\in\Lambda_p(i^\star)} \sum_{k\in[K]} N_t^k \operatorname{KL}_k(\theta, \lambda)} < \sqrt{\beta_{t,\delta}} + \sqrt{\beta_{t,1/t^2}}.$$

On the other hand, if the algorithm uses the LLR stopping rule, for each time t *such that* E_t *holds and the algorithm did not stop,*

$$\sqrt{\inf_{\lambda\in\Lambda(i^\star)} \sum_{k\in[K]} N_t^k \operatorname{KL}_k(\theta, \lambda)} < \sqrt{\beta_{t,\delta}} + \sqrt{\beta_{t,1/t^2}}.$$

Proof. Since $p \in \mathcal{P}_t(i^\star(\theta))$ (i.e., p has not been eliminated at time t), we have from (5) and Lemma C.7 that

$$\beta_{t,\delta} > \inf_{\lambda \in \Lambda_p(i^\star)} L_t(\hat{\theta}_t, \lambda) \geq \inf_{\lambda \in \Lambda_p(i^\star)} \left(\sqrt{\sum_{k \in [K]} N_t^k \operatorname{KL}_k(\theta, \lambda)} - \sqrt{L_t(\hat{\theta}_t, \theta)} \right)^2 .$$

By definition of event E_t, this implies

$$\sqrt{\inf_{\lambda \in \Lambda_p(i^\star)} \sum_{k \in [K]} N_t^k \operatorname{KL}_k(\theta, \lambda)} < \sqrt{\beta_{t,\delta}} + \sqrt{\beta_{t,1/t^2}}.$$

This yields the first statement. The second result can be shown analogously by using the full set of alternatives to θ.

□

Selective elimination updates, at each step t, only the set of active pieces of the empirical answer $i^\star(\hat{\theta}_t)$. Therefore, to bound the elimination times of the pieces $\Lambda_p(i^\star)$ for $p \in \mathcal{P}(i^\star)$ we need to show that $i^\star(\hat{\theta}_t) \neq i^\star$ after a certain time. We show that Assumption 2.5 alone is sufficient to guarantee this.

Lemma D.4. *Consider a sampling rule satisfying Assumption 2.5. Under event E_t, a sufficient condition for $i^\star(\hat{\theta}_t) = i^\star$ is*

$$t \geq \frac{4\beta_{t,1/t^2} + R(\theta, t)}{H^\star(\theta)}. \tag{10}$$

Proof. First note that, under E_t, if

$$\inf_{\lambda \in \Lambda(i^\star(\hat{\theta}_t))} L_t(\hat{\theta}_t, \lambda) > \beta_{t,1/t^2},$$

then $i^\star(\hat{\theta}_t) = i^\star$. In fact, if this was not the case, we would have $\theta \in \Lambda(i^\star(\hat{\theta}_t))$ and thus $L_t(\hat{\theta}_t, \theta) > \beta_{t,1/t^2}$, which is a contradiction with event E_t itself. Let us now look for a sufficient condition on t to satisfy this inequality. Take t and suppose it does not satisfy it. Then,

$$\begin{aligned}
\sqrt{\beta_{t,1/t^2}} \geq \sqrt{\inf_{\lambda \in \Lambda(i^\star(\hat{\theta}_t))} L_t(\hat{\theta}_t, \lambda)} &\overset{(a)}{\geq} \sqrt{\inf_{\lambda \in \Lambda(i^\star(\hat{\theta}_t))} \sum_{k \in [K]} N_t^k \operatorname{KL}_k(\theta, \lambda)} - \sqrt{L_t(\hat{\theta}_t, \theta)} \\
&\overset{(b)}{\geq} \sqrt{\inf_{\lambda \in \Lambda(i^\star)} \sum_{k \in [K]} N_t^k \operatorname{KL}_k(\theta, \lambda)} - \sqrt{\beta_{t,1/t^2}} \\
&\overset{(c)}{\geq} \sqrt{tH^\star(\theta) - R(\theta, t)} - \sqrt{\beta_{t,1/t^2}},
\end{aligned}$$

where (a) is from Lemma C.7, (b) from E_t and the fact that either $i^\star(\hat{\theta}_t) = i^\star$ or the infimum is zero, and (c) from Assumption 2.5. Rearranging this inequality yields the desired condition on t. □

Proof of Theorem 2.6. We start by proving the sample complexity bound for LLR stopping. Since it is a bound on the elimination time of the whole alternative $\Lambda(i^\star)$, by definition it also holds for the elimination times of its pieces obtained with either full or selective elimination. We then move to full elimination stopping and selective elimination stopping. Finally, we prove that the left-hand side of the maximum in (15) is never larger than the bound for LLR stopping.

LLR stopping Let $t > 0$ such that E_t holds and the algorithm did not stop. The second statement in Lemma D.3 yields

$$\sqrt{\inf_{\lambda \in \Lambda(i^\star)} \sum_{k \in [K]} N_t^k \operatorname{KL}_k(\theta, \lambda)} < \sqrt{\beta_{t,\delta}} + \sqrt{\beta_{t,1/t^2}}.$$

Moreover, from Assumption 2.5

$$\inf_{\lambda \in \Lambda(i^\star)} \sum_{k \in [K]} N_t^k \, \mathrm{KL}_k(\theta, \lambda) \geq t H^\star(\theta) - R(\theta, t).$$

The combination of these two inequalities directly yields the stated inequality on t. The result in expectation is obtained by applying Lemma 19 in [29] together with $\mathbb{P}(\neg E_t) \leq 1/t^2$.

Full elimination stopping Let $t > 0$ such that E_t holds. By Assumption 2.5,

$$H^\star(\theta) \leq \inf_{\lambda \in \Lambda(i^\star)} \sum_{k \in [K]} \frac{N_t^k}{t} \, \mathrm{KL}_k(\theta, \lambda) + \frac{R(\theta, t)}{t}$$

This implies that $\frac{N_t}{t} \in \Omega_{R(\theta,t)/t}$.

Now fix a piece $p \in \mathcal{P}(i^\star)$ such that $p \in \mathcal{P}_t(i^\star)$. Under event E_t, we know from Lemma D.3 that

$$\sqrt{\inf_{\lambda \in \Lambda_p(i^\star)} \sum_{k \in [K]} N_t^k \, \mathrm{KL}_k(\theta, \lambda)} < \sqrt{\beta_{t,\delta}} + \sqrt{\beta_{t,1/t^2}}.$$

Since $\frac{N_t}{t} \in \Omega_{R(\theta,t)/t}$,

$$\inf_{\lambda \in \Lambda_p(i^\star)} \sum_{k \in [K]} N_t^k \, \mathrm{KL}_k(\theta, \lambda) \geq t \min_{\omega \in \Omega_{R(\theta,t)/t}} \inf_{\lambda \in \Lambda_p(i^\star)} \sum_{k \in [K]} \omega^k \, \mathrm{KL}_k(\theta, \lambda).$$

Combining the last two displays, we obtain that, if j is not eliminated at time t and E_t holds, t itself must satisfy

$$t < \frac{\left(\sqrt{\beta_{t,\delta}} + \sqrt{\beta_{t,1/t^2}}\right)^2}{\min_{\omega \in \Omega_{R(\theta,t)/t}} \inf_{\lambda \in \Lambda_p(i^\star)} \sum_{k \in [K]} \omega^k \, \mathrm{KL}_k(\theta, \lambda)}.$$

The result in expectation is obtained by applying Lemma 19 in [29] together with $\mathbb{P}(\neg E_t) \leq 1/t^2$.

Selective elimination stopping Given Lemma D.4, the proof of Theorem 2.6 for elimination stopping is very simple. Simply take a time t such that E_t holds and which verifies the condition in Lemma D.4. Then, for such a t the first claim of Lemma D.3 can be verified analogously since the empirical answer (the one for which the set of active pieces is updated) is exactly the correct answer. Given Lemma D.3, the same derivation as in the proof for full elimination can be carried out. This yields the following sufficient condition on the time t to eliminate a piece $p \in \mathcal{P}(i^\star)$:

$$t \geq \max \left\{ \frac{\left(\sqrt{\beta_{t,\delta}} + \sqrt{\beta_{t,1/t^2}}\right)^2}{\min_{\omega \in \Omega_{R(\theta,t)/t}} \inf_{\lambda \in \Lambda_p(i^\star)} \sum_{k \in [K]} \omega^k \, \mathrm{KL}_k(\theta, \lambda)}, \frac{4\beta_{t,1/t^2} + R(\theta, t)}{H^\star(\theta)} \right\}.$$

Comparison of the bounds for full elimination and LLR stopping We finally prove that a sufficient condition for $t \geq \bar{t}_p$, with $\bar{t}_p$ associated to full elimination, is $t \geq \bar{t}$.

Take any $p \in \mathcal{P}(i^\star)$. By definition of the set Ω_ϵ, we have that, for any $\omega \in \Omega_\epsilon$,

$$\inf_{\lambda \in \Lambda_p(i^\star)} \sum_{k \in [K]} \omega^k \, \mathrm{KL}_k(\theta, \lambda) \geq \inf_{\lambda \in \Lambda(i^\star)} \sum_{k \in [K]} \omega^k \, \mathrm{KL}_k(\theta, \lambda) \geq H^\star(\theta) - \epsilon.$$

This implies that

$$\min_{\omega \in \Omega_\epsilon} \inf_{\lambda \in \Lambda_p(i^\star)} \sum_{k \in [K]} \omega^k \, \mathrm{KL}_k(\theta, \lambda) \geq \max_{\omega \in \Delta_K} \inf_{\lambda \in \Lambda(i^\star)} \sum_{k \in [K]} \omega^k \, \mathrm{KL}_k(\theta, \lambda) - \epsilon.$$

A sufficient condition for satisfying the inequality for piece elimination is thus

$$t \geq \frac{\left(\sqrt{\beta_{t,\delta}} + \sqrt{\beta_{t,1/t^2}}\right)^2}{\max_{\omega \in \Delta_K} \inf_{\lambda \in \Lambda(i^\star)} \sum_{k \in [K]} \omega^k \, \mathrm{KL}_k(\theta, \lambda) - R(\theta, t)/t}.$$

Let $\gamma \in (0,1)$. For $R(\theta,t)/t \leq \gamma H^\star(\theta)$ we have that

$$t \geq \frac{\left(\sqrt{\beta_{t,\delta}} + \sqrt{\beta_{t,1/t^2}}\right)^2}{(1-\gamma)\max_{\omega\in\Delta_K}\inf_{\lambda\in\Lambda(i^\star)}\sum_{k\in[K]}\omega^k \operatorname{KL}_k(\theta,\lambda)}$$

suffices. Therefore, taking the maximum between the condition above and $t \geq \frac{R(\theta,t)}{\gamma H^\star(\theta)}$ and optimizing over $\gamma \in (0,1)$, the inequality for piece elimination is verified if

$$t \geq \frac{1}{H^\star(\theta)} \inf_{\gamma\in(0,1)} \max\left\{\frac{\left(\sqrt{\beta_{t,\delta}} + \sqrt{\beta_{t,1/t^2}}\right)^2}{(1-\gamma)}, \frac{R(\theta,t)}{\gamma}\right\}$$

Optimizing over γ, which amounts to setting $\gamma = \frac{R(\theta,t)}{\left(\sqrt{\beta_{t,\delta}}+\sqrt{\beta_{t,1/t^2}}\right)^2 + R(\theta,t)}$, yields the desired statement.

□

D.4 Proof of Theorem 2.7

Proof. Take any time t and suppose that E_t holds while the algorithm did not stop with τ_{llr} yet. Using the first result of Theorem 2.6 yields

$$t < H^\star(\theta)^{-1}\left(\left(\sqrt{\beta_{t,\delta}} + \sqrt{\beta_{t,1/t^2}}\right)^2 + R(\theta,t)\right) .$$

From Lemma 19 in [29], we get that $\mathbb{E}[\tau_{\mathrm{llr}}] \leq \bar{t} + 2$, where $\bar{t}$ is the first time that does not satisfy the inequality above (it is a function of θ and δ). This means that

$$\mathbb{E}[\tau_{\mathrm{llr}}] \leq H^\star(\theta)^{-1}\left(\left(\sqrt{\beta_{t,\delta}} + \sqrt{\beta_{t,1/t^2}}\right)^2 + R(\theta,t)\right) + 3.$$

Now let us take the same algorithm with τ_{elim}. We know it is δ-correct, so the standard lower bound states that

$$\mathbb{E}[\tau_{\mathrm{elim}}] \geq H^\star(\theta)^{-1}\log(1/2.4\delta).$$

We get that

$$\begin{aligned}
&H^\star(\theta)\mathbb{E}[\tau_{\mathrm{llr}}] \\
&\leq H^\star(\theta)\mathbb{E}[\tau_{\mathrm{elim}}] + 3H^\star(\theta) + \left(\sqrt{\beta_{\bar{t},\delta}} + \sqrt{\beta_{\bar{t},1/\bar{t}^2}}\right)^2 + R(\theta,\bar{t}) - \log\frac{1}{2.4\delta} \\
&= H^\star(\theta)\mathbb{E}[\tau_{\mathrm{elim}}] + 3H^\star(\theta) + \beta_{\bar{t},\delta} - \log\frac{1}{\delta} + 2\sqrt{\beta_{\bar{t},\delta}\beta_{\bar{t},1/\bar{t}^2}} + \beta_{\bar{t},1/\bar{t}^2} + R(\theta,\bar{t}) + \log(2.4) .
\end{aligned}$$

We now use the hypothesis $\beta_{\bar{t},\delta} = \log\frac{1}{\delta} + \xi(\bar{t},\delta)$:

$$H^\star(\theta)\mathbb{E}[\tau_{\mathrm{llr}}] \leq H^\star(\theta)\mathbb{E}[\tau_{\mathrm{elim}}] + 3H^\star(\theta) + \xi(\bar{t},\delta) + 2\sqrt{\beta_{\bar{t},\delta}\beta_{\bar{t},1/\bar{t}^2}} + \beta_{\bar{t},1/\bar{t}^2} + R(\theta,\bar{t}) + \log(2.4) .$$

□

E Proofs of Section 3

In order to tune our resets, we need an hypothesis on the shape of the threshold, used to compare thresholds at different times.

Assumption E.1. *There exist two positive reals c_1 and c_2 that may depend on the parameters of the problem known to the algorithm (d, K, etc.) and a function $\overline{\log}$ such that the threshold $\beta_{t,\delta}$ verifies*

$$\overline{\log}(1/\delta) + c_1\log(t) \leq \beta_{t,\delta} \leq \overline{\log}(1/\delta) + c_2\log(t) .$$

Furthermore, $\overline{\log}$ verifies that there exists x_0 such that for all $a \geq 2$ and $x \geq x_0$, $\overline{\log}(x^a) \leq a\overline{\log}(x)$.

The function $\overline{\log}$ represents an almost logarithmic function and corresponds to the function of order $\log + \sqrt{\log}$ in [25] or $\log + d \log\log$ in [30, 29]. Other thresholds have a $O(\log\log(t))$ dependence instead of $O(\log(t))$, and our analysis could be extended to them in a similar way.

With that assumption, we set $\bar{t}_0 = \max\{2, \sqrt{x_0}\}$ and $\alpha_{t,\delta} := \left(\sqrt{\beta_{t,\delta} + (4c_2 - c_1)\log(t)} + 4\sqrt{\frac{c_2}{c_1}\beta_{t,1/t^2}}\right)^2$.

The reset times we defined are taylored to a threshold $\beta_{t,\delta}$ of order $\log(1/\delta) + O(\log(t))$ as in Assumption E.1. They are chosen such that $\beta_{t,1/t^2}$ roughly doubles from one reset to the next. If the thresold has a different dependence in t (for example $O(\log\log t)$) then the reset times can be adapted.

E.1 On Assumption E.1

We illustrate first how Assumption E.1 covers the threshold obtained by the most common technique: first get a concentration bound valid for one time of the form $f(1/\delta)$, where usually $f(x) = \log(x) + C\log\log(x)$ (for C a dimension-dependent constant), then obtain a time-uniform concentration threshold by setting for example $\beta_{t,\delta} = f(\frac{\pi^2 t^2}{6\delta})$.

Let us suppose then that $\beta_{t,\delta} = f(\frac{C' t^2}{\delta})$ for $f(x) = \log(x) + C\log\log(x)$ and $C' \geq 1$. We will now prove that if verifies Assumption E.1.

$$\begin{aligned}
\overline{\log}(C't^2/\delta) &= \log\frac{C't^2}{\delta} + C\log(\log\frac{C'}{\delta} + \log t^2) \\
&\geq \log\frac{C'}{\delta} + C\log\log\frac{C'}{\delta} + 2\log(t) \\
&= \overline{\log}\frac{1}{\delta} + 2\log(t)\ .
\end{aligned}$$

where $\overline{\log}(x) = \log(C'x) + C\log\log(C'x)$. Using the concavity of $\log$, we also have an upper bound

$$\begin{aligned}
\overline{\log}(C't^2/\delta) &= \log\frac{C't^2}{\delta} + C\log(\log\frac{C'}{\delta} + \log t^2) \\
&= \log\frac{C'}{\delta} + C\log\log\frac{C'}{\delta} + 2\log(t) + C(\log(\log\frac{C'}{\delta} + \log t^2) - \log\log\frac{C'}{\delta}) \\
&\leq \overline{\log}\frac{1}{\delta} + 2(C+1)\log(t)\ .
\end{aligned}$$

We have found a function $\overline{\log}$ such that for $c_1 = 2$ and $c_2 = 2(C+1)$,

$$\overline{\log}(\frac{1}{\delta}) + c_1\log(t) \leq \beta_{t,\delta} \leq \overline{\log}(\frac{1}{\delta}) + c_2\log(t)\ .$$

It remains to show the condition on $\overline{\log}$, i.e. find x_0 such that for $a \geq 2$ and $x \geq x_0$, $\overline{\log}(x^a) \leq a\overline{\log}(x)$.

$$\begin{aligned}
\overline{\log}(x^a) = \log(C'x^a) + C\log\log(C'x^a) &\leq \log((C'x)^a) + C\log\log((C'x)^a) \\
&= a\log(C'x) + C\log\log(C'x) + C\log a
\end{aligned}$$

It remains to find x_0 such that for $x \geq x_0$, $\log\log(C'x) + \log a \leq a\log\log(C'x)$. We find that $x_0 = \exp(a^{1/(a-1)})/C'$ is suitable. Since $a \geq 2$, we have $x_0 \leq e^2/C'$.

Lemma E.2. *Under Assumption E.1, for any time t and $0 \leq j \leq j(t)$,*

$$\begin{aligned}
\beta_{t,\delta} &\leq \beta_{\bar{t}_j,\delta} + (2^{j(t)+1-j}c_2 - c_1)\log(\bar{t}_j),\ , \\
\beta_{t,1/t^2} &\leq \frac{c_2}{c_1}2^{j(t)-j+1}\beta_{\bar{t}_j,1/\bar{t}_j^2}.
\end{aligned}$$

Proof. We have $\bar{t}_0 = \max\{2, \sqrt{x_0}\}$. Note that for all j, $\bar{t}_j = \bar{t}_0^{2^j}$. By using first the upper bound of Assumption E.1, then $t \leq \bar{t}_0^{2^{j(t)+1}}$ and at the end the lower bound of Assumption E.1,

$$\begin{aligned}\beta_{t,\delta} \leq \overline{\log}(\frac{1}{\delta}) + c_2 \log(t) \leq \overline{\log}(\frac{1}{\delta}) + c_2 \log(\bar{t}_0^{2^{j(t)+1}}) &= \overline{\log}(\frac{1}{\delta}) + c_2 \log((\bar{t}_0^{2^j})^{2^{j(t)+1-j}}) \\ &= \overline{\log}(\frac{1}{\delta}) + c_2 2^{j(t)+1-j} \log(\bar{t}_0^{2^j}) \\ &\leq \beta_{\bar{t}_j,\delta} + \log(\bar{t}_j)(2^{j(t)+1-j} c_2 - c_1)\,.\end{aligned}$$

We have $\bar{t}_j \geq \bar{t}_0 \geq \sqrt{x_0}$, hence the inequality $\overline{\log}(x^a) \leq a\overline{\log}(x)$ can be used. Then

$$\begin{aligned}\beta_{t,1/t^2} \leq \overline{\log}(t^2) + c_2 \log(t) &\leq \overline{\log}(\bar{t}_0^{2^{j(t)+2}}) + c_2 \log(\bar{t}_0^{2^{j(t)+1}}) \\ &= \overline{\log}((((\bar{t}_0^{2^j})^2)^{2^{j(t)+1-j}}) + c_2 \log((\bar{t}_0^{2^j})^{2^{j(t)+1-j}}) \\ &= \overline{\log}(((\bar{t}_j^2)^{2^{j(t)+1-j}}) + c_2 2^{j(t)+1-j} \log(\bar{t}_j) \\ &\leq 2^{j(t)+1-j}\overline{\log}((\bar{t}_j^2) + c_2 2^{j(t)+1-j} \log(\bar{t}_j) \\ &\leq \frac{c_2}{c_1}\left(2^{j(t)+1-j}\overline{\log}((\bar{t}_j^2) + c_1 2^{j(t)+1-j} \log(\bar{t}_j)\right) \\ &\leq \frac{c_2}{c_1} 2^{j(t)+1-j} \beta_{\bar{t}_j,1/\bar{t}_j^2}\,.\end{aligned}$$

□

E.2 Proof of Theorem 3.2

We now derive an important result for sampling rules combined with elimination (either full or selective). It essentially shows that we cannot eliminate the closest alternative piece to θ from the sampling rule without making the algorithm stop.

Lemma E.3. *Let $t \geq \bar{t}_1$ be any time step at which the algorithm did not stop. Suppose that some piece index $p \in \mathcal{P}(i^\star)$ of the true correct answer $i^\star$ has been eliminated from the sampling rule (i.e., $p \notin \mathcal{P}_t^{\mathrm{smp}}(i^\star)$). Then, under event E_t (see Equation 6),*

$$\inf_{\lambda \in \Lambda_p(i^\star)} \sum_{k \in [K]} N_t^k \,\mathrm{KL}_k(\theta, \lambda) > \inf_{\lambda \in \Lambda(i^\star)} \sum_{k \in [K]} N_t^k \,\mathrm{KL}_k(\theta, \lambda).$$

Proof. Let us proceed by contradiction: suppose that $p \notin \mathcal{P}_t^{\mathrm{smp}}(i^\star)$ while $\inf_{\lambda \in \Lambda_p(i^\star)} \sum_{k \in [K]} N_t^k \,\mathrm{KL}_k(\theta, \lambda) = \inf_{\lambda \in \Lambda(i^\star)} \sum_{k \in [K]} N_t^k \,\mathrm{KL}_k(\theta, \lambda)$ and E_t holds.

Since $\mathcal{P}_t^{\mathrm{smp}}(i^\star)$ is the intersection of all active sets from $\bar{t}_{j(t)-1}$ to t, if $p \notin \mathcal{P}_t^{\mathrm{smp}}(i^\star)$, then there exists s with $\bar{t}_{j(t)-1} \leq s \leq t$ such that

$$\inf_{\lambda \in \Lambda_p(i)} L_s(\hat{\theta}_s, \lambda) \geq \alpha_{s,\delta}.$$

Therefore,

$$\begin{aligned}\sqrt{\alpha_{s,\delta}} &\overset{(a)}{\leq} \sqrt{\inf_{\lambda \in \Lambda_p(i^\star)} L_s(\hat{\theta}_s, \lambda)} \overset{(b)}{\leq} \sqrt{\inf_{\lambda \in \Lambda_p(i^\star)} \sum_{k \in [K]} N_s^k \,\mathrm{KL}_k(\theta, \lambda)} + \sqrt{L_s(\hat{\theta}_s, \theta)} \\ &\overset{(c)}{\leq} \sqrt{\inf_{\lambda \in \Lambda_p(i^\star)} \sum_{k \in [K]} N_s^k \,\mathrm{KL}_k(\theta, \lambda)} + \sqrt{\beta_{t,1/t^2}} \\ &\overset{(d)}{\leq} \sqrt{\inf_{\lambda \in \Lambda_p(i^\star)} \sum_{k \in [K]} N_t^k \,\mathrm{KL}_k(\theta, \lambda)} + \sqrt{\beta_{t,1/t^2}} \\ &\overset{(e)}{=} \sqrt{\inf_{\lambda \in \Lambda(i^\star)} \sum_{k \in [K]} N_t^k \,\mathrm{KL}_k(\theta, \lambda)} + \sqrt{\beta_{t,1/t^2}},\end{aligned}$$

where (a) is from the elimination condition, (b) uses Lemma C.7, (c) uses that E_t holds, (d) uses that the number of pulls of each arm is non-decreasing in time, and (e) holds by our assumption. Recall that, for any $\theta, \lambda \in \mathbb{R}^d$, $\sum_{k\in[K]} N_t^k \operatorname{KL}_k(\theta,\lambda) = \frac{1}{2}\|\theta-\lambda\|_{V_t}^2$. Therefore, by the triangle inequality,

$$\sqrt{\sum_{k\in[K]} N_t^k \operatorname{KL}_k(\theta,\lambda)} = \frac{1}{\sqrt{2}}\|\theta-\lambda\|_{V_t} \le \frac{1}{\sqrt{2}}\|\theta-\hat{\theta}_t\|_{V_t} + \frac{1}{\sqrt{2}}\|\hat{\theta}_t-\lambda\|_{V_t}$$
$$= \sqrt{L_t(\hat{\theta}_t,\lambda)} + \sqrt{L_t(\hat{\theta}_t,\theta)}.$$

Combining this with the previous chain of inequalities,

$$\sqrt{\alpha_{s,\delta}} \le \sqrt{\inf_{\lambda\in\Lambda(i^\star)} L_t(\hat{\theta}_t,\lambda)} + \sqrt{L_t(\hat{\theta}_t,\theta)} + \sqrt{\beta_{t,1/t^2}} \le \sqrt{\inf_{\lambda\in\Lambda(i^\star)} L_t(\hat{\theta}_t,\lambda)} + 2\sqrt{\beta_{t,1/t^2}}$$
$$\le \sqrt{\inf_{\lambda\in\Lambda(i^\star(\hat{\theta}_t))} L_t(\hat{\theta}_t,\lambda)} + 2\sqrt{\beta_{t,1/t^2}},$$

where we used again that E_t holds to concentrate the LLR between $\hat{\theta}_t$ and θ. The last inequality is easy to check since, if $i^\star \neq i^\star(\hat{\theta}_t)$, then $\inf_{\lambda\in\Lambda(i^\star)} L_t(\hat{\theta}_t,\lambda) = 0$. Finally, since the algorithm did not stop at t, it must be that $\inf_{\lambda\in\Lambda(i^\star)} L_t(\hat{\theta}_t,\lambda) < \beta_{t,\delta}$ (otherwise all alternative pieces of $i^\star(\hat{\theta}_t)$ would be eliminated at t). Therefore,

$$\sqrt{\alpha_{s,\delta}} < \sqrt{\beta_{t,\delta}} + 2\sqrt{\beta_{t,1/t^2}}.$$

Since $s \ge \bar{t}_{j(t)-1}$, from Lemma E.2 we have that

$$\beta_{t,\delta} \le \beta_{\bar{t}_{j(t)-1},\delta} + (4c_2 - c_1)\log(\bar{t}_{j(t)-1}) \le \beta_{s,\delta} + (4c_2-c_1)\log(s),$$
$$\beta_{t,1/t^2} \le 4\frac{c_2}{c_1}\beta_{\bar{t}_{j(t)-1},1/\bar{t}^2_{j(t)-1}} \le 4\frac{c_2}{c_1}\beta_{s,1/s^2}.$$

Plugging this into our previous bound, we conclude that

$$\sqrt{\alpha_{s,\delta}} < \sqrt{\beta_{s,\delta} + (4c_2-c_1)\log(s)} + 4\sqrt{\frac{c_2}{c_1}\beta_{s,1/s^2}}.$$

This is clearly a contradiction w.r.t. our definition of $\alpha_{t,\delta}$. □

Proof of Theorem 3.2. Take any step $t \ge \bar{t}_1$ where E_t holds. Lemma E.3 ensures that all pieces of $i^\star$ which are at minimal distance from θ, i.e., such that

$$\inf_{\lambda\in\Lambda_p(i^\star)} \sum_{k\in[K]} N_t^k \operatorname{KL}_k(\theta,\lambda) = \inf_{\lambda\in\Lambda(i^\star)} \sum_{k\in[K]} N_t^k \operatorname{KL}_k(\theta,\lambda)$$

are not eliminated for the sampling rule at time t, i.e., $p \in \mathcal{P}_t^{\mathrm{smp}}(i^\star)$. This implies that

$$\min_{p\in\mathcal{P}_t^{\mathrm{smp}}(i^\star)} \inf_{\lambda\in\Lambda_p(i^\star)} \sum_{k\in[K]} N_t^k \operatorname{KL}_k(\theta,\lambda) = \inf_{\lambda\in\Lambda(i^\star)} \sum_{k\in[K]} N_t^k \operatorname{KL}_k(\theta,\lambda).$$

Thus, if Assumption 3.1 holds, it must be that

$$\inf_{\lambda\in\Lambda(i^\star)} \sum_{k\in[K]} N_t^k \operatorname{KL}_k(\theta,\lambda) \ge \max_{\omega\in\Delta_K} \sum_{s=1}^{t} \min_{p\in\mathcal{P}_{s-1}^{\mathrm{smp}}(i^\star)} \inf_{\lambda\in\Lambda_p(i^\star)} \sum_{k\in[K]} \omega^k \operatorname{KL}_k(\theta,\lambda) - R(\theta,t)$$
$$\ge \max_{\omega\in\Delta_K} \sum_{s=1}^{t} \inf_{\lambda\in\Lambda(i^\star)} \sum_{k\in[K]} \omega^k \operatorname{KL}_k(\theta,\lambda) - R(\theta,t) = tH^\star(\theta) - R(\theta,t),$$

where the second inequality is trivial from $\mathcal{P}_{s-1}^{\mathrm{smp}}(i^\star) \subseteq \mathcal{P}(i^\star)$ for all $s \ge 1$. Therefore, we proved that the condition of Assumption 2.5 holds as well for all $t \ge \bar{t}_1$. We can now have it for all t by adding $\bar{t}_1 H^\star(\theta)$ to $R(\theta,t)$ to obtain another regret function which verifies the condition of Assumption 2.5. The second statement is a direct consequence of the fact that Theorem 2.6 holds for any sampling rule that satisifes the latter assumption. □

F Existing Algorithms Satisfy Assumption 2.5 and 3.1

In this section, we show that existing sampling rules that target the optimal value of the max-inf game from the general lower bound of [31] satisfy Assumption 2.5. Moreover, we show that, when such sampling rules are combined with elimination (as in Section 3), they also satisfy Assumption 3.1. We do that explicitly for two algorithms: the game-theoretic approach based on no-regret learners of [15] and an optimistic variant of the Track-and-Stop algorithm of [31]. These two algorithms are sufficiently general to be representative of many existing approaches for pure exploration: the latter represents those that repeatedly solve the optimization problem from the lower bound to get optimal allocations, while the former represents those that solve such problem incrementally. We now describe these two algorithmic techniques. Then, we show, through a unified proof, that they satisfy Assumption 2.5 and 3.1.

F.1 Sampling Rules

We describe the sampling rules of interest while trying to keep some of their design choices (e.g., confidence intervals, tracking, optimization, etc.) as general as possible. We do this because these sampling rules have been adapted to different pure exploration problems and bandit structures in the literature, for which such components would be different. This will allow us to have unified proofs that are actually agnostic to the specific setting.

F.2 Common Assumptions

We first state some assumptions on the main common components of these algorithms.

Bounded closest alternatives First, we shall make a mild regularity assumption about the considered identification problem: the distance between any closest alternative and θ is bounded.

Assumption F.1. *There exists a constant $B > 0$ such that, for any $\omega \in \mathbb{R}_{\geq 0}$ and any subset $\Lambda \subseteq \mathcal{M}$ which is exactly the union of arbitrary pieces, there exists a closest alternative*

$$\lambda_\omega \in \arg\min_{\lambda \in \Lambda} \sum_{k=1}^{K} \omega^K \, \mathrm{KL}_k(\theta, \lambda)$$

such that

$$\max_{k \in [K]} \mathrm{KL}_k(\theta, \lambda_\omega) \leq B.$$

We note that this assumption is satisfied in the identification problems we consider (see Appendix B), for Gaussian rewards where pieces are half-spaces.

Tracking All the sampling rules we consider choose a sequence of proportions $(\omega_t)_{t \geq 1}$, where $\omega_t \in \Delta_K$, and use tracking to select the next arm to play based on these. Formally, the arm played at time t is

$$k_t := \mathrm{Track}\left(\sum_{s=1}^{t} \omega_s, N_{t-1}\right), \tag{11}$$

where $\mathrm{Track} : \mathbb{R}_+^K \times \mathbb{R}_+^K \to [K]$ is some tracking function. To gain generality, we shall keep the tracking function implicit in the remainder, while only requiring the following assumption (which is actually guaranteed by existing methods).

Assumption F.2. *There exists a constant $C_{\mathrm{track}} > 0$ such that*

$$\forall t > 0, k \in [K] : N_t^k \geq \sum_{s=1}^{t} \omega_s^k - C_{\mathrm{track}}.$$

For instance, the widely-adopted cumulative tracking,

$$\mathrm{Track}\left(\sum_{s=1}^{t} \omega_s, N_{t-1}\right) = \arg\min_{k \in [K]} \left(N_{t-1}^k - \sum_{s=1}^{t} \omega_s^k\right),$$

satisfies this assumption.

Confidence intervals These sampling rules maintain confidence intervals $(c_t^k)_{t\geq 1, k\in[K]}$ about the expected return of each arm. We shall also keep them implicit as their specific form depends on the bandit structure under consideration (e.g., linear vs unstructured). We will only require the following assumption, which is satisfied by common choices as described below.

Assumption F.3. *Under event E_t,*

$$\forall s \leq t : \mathrm{KL}_k(\hat{\theta}_s, \lambda) - c_s^k \leq \mathrm{KL}_k(\theta, \lambda) \leq \mathrm{KL}_k(\hat{\theta}_s, \lambda) + c_s^k.$$

Moreover, there exists a sub-linear (in t) function $C_{\mathrm{conf}}(t)$ such that

$$\sum_{s=1}^{t} \sum_{k\in[K]} \omega_s^k c_{s-1}^k \leq C_{\mathrm{conf}}(t).$$

In bandits with linear structure, the confidence intervals typically take the form $c_t^k \propto \|\phi_k\|_{V_t^{-1}}$. In this case finding an upper bound on $\sum_{s=1}^{t} \sum_{k\in[K]} \omega_s^k c_{s-1}^k$ would reduce to applying an elliptical potential lemma [25] plus the tracking property, and the resulting upper bound would be $C_{\mathrm{conf}}(t) \propto \sqrt{dt}$. In unstructured bandits we can take $c_t^k \propto 1/\sqrt{N_t^k}$ and finding $C_{\mathrm{conf}}(t)$ would require applying the standard pigeon-hole principle plus the tracking property, for which one obtains $C_{\mathrm{conf}}(t) \propto \sqrt{Kt}$.

F.2.1 Optimistic Track-and-Stop

Here we describe an optimistic variant of the Track-and-Stop algorithm by [31]. It was originally introduced by [15] in order to get rid of forced exploration, one of the main causes behind the poor empirical performance of Track-and-Stop. The idea is to solve, at each time step t, an optimistic variant of the optimization problem from the lower bound,

$$\omega_t := \operatorname*{arg\,max}_{\omega\in\Delta_K} \inf_{\lambda\in\Lambda(i^\star(\hat{\theta}_{t-1}))} \sum_{k\in[K]} \omega^k \left(\mathrm{KL}_k(\hat{\theta}_{t-1}, \lambda) + c_{t-1}^k\right), \tag{12}$$

where c_{t-1}^k is a per-arm confidence interval that satisfies Assumption F.3, thus ensuring optimism $\mathrm{KL}_k(\hat{\theta}_{t-1}, \lambda) + c_{t-1}^k \geq \mathrm{KL}_k(\theta, \lambda)$ with high probability. The solution to this optimization problem yields proportions ω_t which are tracked by the sampling rule. Then, the arm played at time t is given by the tracking rule (11).

Combining with elimination In order to combine this sampling rule with elimination, we simply redefine the optimization problem as

$$\omega_t := \operatorname*{arg\,max}_{\omega\in\Delta_K} \min_{p\in\mathcal{P}_{t-1}^{\mathrm{smp}}(i^\star(\hat{\theta}_{t-1}))} \inf_{\lambda\in\Lambda_p(i^\star(\hat{\theta}_{t-1}))} \sum_{k\in[K]} \omega^k \left(\mathrm{KL}_k(\hat{\theta}_{t-1}, \lambda) + c_{t-1}^k\right). \tag{13}$$

That is, we simply replace a minimization over the whole alternative with one over the active pieces only.

F.2.2 Game-Theoretic Approach with No-Regret Learners

The idea behind the game-theoretic approach of [15] is to avoid recomputing the full optimistic problem (12) at each step, while solving it incrementally by means of no-regret learners. Given some online-learning algorithm $\mathcal{L}$ working on the K-dimensional simplex, the sampling rule works as follows. At each time t, first $\mathcal{L}$ outputs new proportions ω_t. Then, we compute the closest alternative

$$\hat{\lambda}_t := \operatorname*{arg\,min}_{\lambda\in\Lambda(i^\star(\hat{\theta}_{t-1}))} \sum_{k\in[K]} \omega_t^k \,\mathrm{KL}_k(\hat{\theta}_{t-1}, \lambda).$$

Finally, $\mathcal{L}$ is updated with the (concave) gain function

$$g_t(\omega) := \sum_{k\in[K]} \omega^k \left(\mathrm{KL}_k(\hat{\theta}_{t-1}, \hat{\lambda}_t) + c_{t-1}^k\right).$$

Finally, the sampling rule uses tracking exactly as in (11) to decide the next arm to pull. As before, we will only require the tracking rule to satisfy Assumption F.2 without specifying an explicit form. Similarly, we will keep the learner implicit as far as it satisfies the following no-regret property.

Assumption F.4. *The learner $\mathcal{L}$ is no regret: there exists a sub-linear (in t) function $C_{\mathcal{L}}(t)$ such that, for any $t \geq 1$ and any sequence of gains $\{g_s(\omega)\}_{s \leq t}$,*

$$\max_{w \in \Delta_K} \sum_{s=1}^{t} \big(g_s(w) - g_s(w_s)\big) \leq C_{\mathcal{L}}(t)\,.$$

Combining with elimination For the game-theoretic approach, we only need to redefine the closest alternative used in the gains as

$$(\hat{p}_t, \hat{\lambda}_t) := \operatorname*{arg\,min}_{p \in \mathcal{P}^{\mathrm{smp}}_{t-1}(i^\star(\hat{\theta}_{t-1})), \lambda \in \Lambda_p(i^\star(\hat{\theta}_{t-1}))} \sum_{k \in [K]} \omega_t^k \,\mathrm{KL}_k(\hat{\theta}_{t-1}, \lambda).$$

F.3 Assumption 2.5 Holds

We need to show that, for any time $t \geq 1$ where the good event E_t (Equation 6) holds, the two sampling rules presented above satisfy

$$tH^\star(\theta) \leq \inf_{\lambda \in \Lambda(i^\star)} \sum_{k \in [K]} N_t^k \,\mathrm{KL}_k(\theta, \lambda) + R(\theta, t)$$

for suitable choices of the function $R(\theta, t)$. We shall write a unified proof for these two sampling rules while explicitly mentioning where they differ. Let us suppose that their algorithmic components satisfy the assumptions stated above, i.e., Assumption F.2 for tracking, Assumption F.3 for the confidence intervals, and Assumption F.4 for the no-regret learner.

Take any time step $t \geq 1$ and suppose that E_t holds. Let $\lambda_t := \operatorname{arg\,min}_{\lambda \in \Lambda(i^\star)} \sum_{k \in [K]} N_t^k \,\mathrm{KL}_k(\theta, \lambda)$. Then, using the tracking property (Assumption F.2) together with Assumption F.1,

$$\begin{aligned}
\inf_{\lambda \in \Lambda(i^\star)} \sum_{k \in [K]} N_t^k \,\mathrm{KL}_k(\theta, \lambda) &= \sum_{k \in [K]} N_t^k \,\mathrm{KL}_k(\theta, \lambda_t) \\
&\geq \sum_{k \in [K]} \sum_{s=1}^{t} \omega_s^k \,\mathrm{KL}_k(\theta, \lambda_t) - C_{\mathrm{track}} \sum_{k \in [K]} \mathrm{KL}_k(\theta, \lambda_t) \\
&\geq \inf_{\lambda \in \Lambda(i^\star)} \sum_{k \in [K]} \sum_{s=1}^{t} \omega_s^k \,\mathrm{KL}_k(\theta, \lambda) - C_{\mathrm{track}} \sum_{k \in [K]} \mathrm{KL}_k(\theta, \lambda_t) \\
&\geq \inf_{\lambda \in \Lambda(i^\star)} \sum_{k \in [K]} \sum_{s=1}^{t} \omega_s^k \,\mathrm{KL}_k(\theta, \lambda) - C_{\mathrm{track}} KB.
\end{aligned}$$

We can now lower bound the first term as

$$\begin{aligned}
\inf_{\lambda \in \Lambda(i^\star)} \sum_{k \in [K]} \sum_{s=1}^{t} \omega_s^k \,\mathrm{KL}_k(\theta, \lambda) &\overset{(a)}{\geq} \sum_{s=1}^{t} \inf_{\lambda \in \Lambda(i^\star)} \sum_{k \in [K]} \omega_s^k \,\mathrm{KL}_k(\theta, \lambda) \\
&\overset{(b)}{\geq} \sum_{s=1}^{t} \inf_{\lambda \in \Lambda(i^\star(\hat{\theta}_{s-1}))} \sum_{k \in [K]} \omega_s^k \,\mathrm{KL}_k(\theta, \lambda) \\
&\overset{(c)}{\geq} \sum_{s=1}^{t} \inf_{\lambda \in \Lambda(i^\star(\hat{\theta}_{s-1}))} \sum_{k \in [K]} \omega_s^k \left(\mathrm{KL}_k(\hat{\theta}_{s-1}, \lambda) - c_{s-1}^k\right) \\
&= \sum_{s=1}^{t} \inf_{\lambda \in \Lambda(i^\star(\hat{\theta}_{s-1}))} \sum_{k \in [K]} \omega_s^k \left(\mathrm{KL}_k(\hat{\theta}_{s-1}, \lambda) + c_{s-1}^k\right) - 2 \sum_{s=1}^{t} \sum_{k \in [K]} \omega_s^k c_{s-1}^k \\
&\overset{(d)}{\geq} \sum_{s=1}^{t} \inf_{\lambda \in \Lambda(i^\star(\hat{\theta}_{s-1}))} \sum_{k \in [K]} \omega_s^k \left(\mathrm{KL}_k(\hat{\theta}_{s-1}, \lambda) + c_{s-1}^k\right) - 2C_{\mathrm{conf}}(t),
\end{aligned}$$

where (a) is from the concavity of the infimum, (b) holds since either $\Lambda(i^\star(\hat{\theta}_{s-1})) = \Lambda(i^\star)$ or $\theta \in \Lambda(i^\star(\hat{\theta}_{s-1}))$ (in which case the infimum would be zero), (c) is from the validity of the confidence intervals under E_t, and (d) is from Assumption F.3. Now note that, when applying the game-theoretic approach (Section F.2.2), the first term on the right-hand side is exactly the sum of gains fed into the learner. Thus, using the no-regret property (Assumption F.4),

$$\begin{aligned}
&\inf_{\lambda \in \Lambda(i^\star)} \sum_{k \in [K]} \sum_{s=1}^{t} \omega_s^k \, \mathrm{KL}_k(\theta, \lambda) \\
&\geq \max_{\omega \in \Delta_K} \sum_{s=1}^{t} \inf_{\lambda \in \Lambda(i^\star(\hat{\theta}_{s-1}))} \sum_{k \in [K]} \omega^k \left(\mathrm{KL}_k(\hat{\theta}_{s-1}, \lambda) + c_{s-1}^k \right) - 2C_{\mathrm{conf}}(t) - C_{\mathcal{L}}(t).
\end{aligned}$$

If instead we are applying optimistic Track-and-Stop (Section F.2.1), the first term on the right-hand side is exactly the sum of optimal values of the objective functions maximized by the algorithm. Thus,

$$\begin{aligned}
&\inf_{\lambda \in \Lambda(i^\star)} \sum_{k \in [K]} \sum_{s=1}^{t} \omega_s^k \, \mathrm{KL}_k(\theta, \lambda) \\
&= \sum_{s=1}^{t} \max_{\omega \in \Delta_K} \inf_{\lambda \in \Lambda(i^\star(\hat{\theta}_{s-1}))} \sum_{k \in [K]} \omega^k \left(\mathrm{KL}_k(\hat{\theta}_{s-1}, \lambda) + c_{s-1}^k \right) - 2C_{\mathrm{conf}}(t) \\
&\geq \max_{\omega \in \Delta_K} \sum_{s=1}^{t} \inf_{\lambda \in \Lambda(i^\star(\hat{\theta}_{s-1}))} \sum_{k \in [K]} \omega^k \left(\mathrm{KL}_k(\hat{\theta}_{s-1}, \lambda) + c_{s-1}^k \right) - 2C_{\mathrm{conf}}(t).
\end{aligned}$$

Therefore, we only need to lower bound the first term above, which is common between the two considered algorithms. We have

$$\begin{aligned}
&\max_{\omega \in \Delta_K} \sum_{s=1}^{t} \inf_{\lambda \in \Lambda(i^\star(\hat{\theta}_{s-1}))} \sum_{k \in [K]} \omega^k \left(\mathrm{KL}_k(\hat{\theta}_{s-1}, \lambda) + c_{s-1}^k \right) \\
&\overset{(e)}{\geq} \max_{\omega \in \Delta_K} \sum_{s=1}^{t} \inf_{\lambda \in \Lambda(i^\star)} \sum_{k \in [K]} \omega^k \left(\mathrm{KL}_k(\hat{\theta}_{s-1}, \lambda) + c_{s-1}^k \right) \\
&\overset{(f)}{\geq} \max_{\omega \in \Delta_K} \sum_{s=1}^{t} \inf_{\lambda \in \Lambda(i^\star)} \sum_{k \in [K]} \omega^k \, \mathrm{KL}_k(\theta, \lambda) \\
&\overset{(g)}{\geq} t H^\star(\theta),
\end{aligned}$$

where (e) follows from the same reasoning as step (b) above, (f) is from the fact that confidence intervals are valid under E_t, and (g) is from the definition of the game in the lower bound. Putting all together, we proved that optimistic Track-and-Stop satisfies Assumption 2.5 with

$$R(\theta, t) = C_{\mathrm{track}} K B + 2C_{\mathrm{conf}}(t),$$

while the game-theoretic approach satisfies it with

$$R(\theta, t) = C_{\mathrm{track}} K B + 2C_{\mathrm{conf}}(t) + C_{\mathcal{L}}(t).$$

F.4 Assumption 3.1 Holds

We now show that Assumption 3.1 holds for these sampling rules combined with elimination as formally explained above. The main steps are very similar as before, with the additional complications posed by eliminating pieces at sampling.

Recall that we want to show that, for any time $t \geq 1$ where the good event E_t (Equation 6) holds, the two sampling rules presented above satisfy

$$\max_{\omega \in \Delta_K} \sum_{s=1}^{t} \min_{p \in \mathcal{P}_{s-1}^{\mathrm{smp}}(i^\star)} \inf_{\lambda \in \Lambda_p(i^\star)} \sum_{k \in [K]} \omega^k \, \mathrm{KL}_k(\theta, \lambda) \leq \min_{p \in \mathcal{P}_t^{\mathrm{smp}}(i^\star)} \inf_{\lambda \in \Lambda_p(i^\star)} \sum_{k \in [K]} N_t^k \, \mathrm{KL}_k(\theta, \lambda) + R(\theta, t)$$

for suitable choices of the function $R(\theta, t)$.

Take any time step $t \geq 1$ and suppose that E_t holds. Let

$$(p_t, \lambda_t) \in \operatorname*{arg\,min}_{p \in \mathcal{P}_t^{\mathrm{smp}}(i^\star), \lambda \in \Lambda_p(i^\star)} \sum_{k \in [K]} N_t^k \, \mathrm{KL}_k(\theta, \lambda)$$

be the closest piece and alternative at time t. Then, using the tracking property (Assumption F.2) and Assumption F.1,

$$\begin{aligned}
\min_{p \in \mathcal{P}_t^{\mathrm{smp}}(i^\star)} \inf_{\lambda \in \Lambda_p(i^\star)} \sum_{k \in [K]} N_t^k \, \mathrm{KL}_k(\theta, \lambda) &= \sum_{k \in [K]} N_t^k \, \mathrm{KL}_k(\theta, \lambda_t) \\
&\geq \sum_{k \in [K]} \sum_{s=1}^{t} \omega_s^k \, \mathrm{KL}_k(\theta, \lambda_t) - C_{\mathrm{track}} K B \\
&\geq \min_{p \in \mathcal{P}_t^{\mathrm{smp}}(i^\star)} \inf_{\lambda \in \Lambda_p(i^\star)} \sum_{k \in [K]} \sum_{s=1}^{t} \omega_s^k \, \mathrm{KL}_k(\theta, \lambda) - C_{\mathrm{track}} K B.
\end{aligned}$$

Recall that $\bar{t}_j := \bar{t}_0^{2^j}$ is the time step at which the j-th reset is performed and $j(t) := \lfloor \log_2 \log_{\bar{t}_0} t \rfloor$ is the index of the last reset before t. Let $\bar{t} := \bar{t}_{j(t)-1}$ be the time of the second-last reset before t. Note that

$$\bar{t} := \bar{t}_0^{2^{j(t)-1}} = \bar{t}_0^{\frac{1}{2} 2^{\lfloor \log_2 \log_{\bar{t}_0} t \rfloor}} = \sqrt{\bar{t}_0^{2^{\lfloor \log_2 \log_{\bar{t}_0} t \rfloor}}} = \sqrt{\bar{t}_{j(t)}} \leq \sqrt{t}.$$

We can now lower bound the first term as

$$\begin{aligned}
&\min_{p \in \mathcal{P}_t^{\mathrm{smp}}(i^\star)} \inf_{\lambda \in \Lambda_p(i^\star)} \sum_{k \in [K]} \sum_{s=1}^{t} \omega_s^k \, \mathrm{KL}_k(\theta, \lambda) \\
&\overset{(a)}{\geq} \sum_{s=1}^{t} \min_{p \in \mathcal{P}_t^{\mathrm{smp}}(i^\star)} \inf_{\lambda \in \Lambda_p(i^\star)} \sum_{k \in [K]} \omega_s^k \, \mathrm{KL}_k(\theta, \lambda) \\
&\overset{(b)}{\geq} \sum_{s=\bar{t}+1}^{t} \min_{p \in \mathcal{P}_t^{\mathrm{smp}}(i^\star)} \inf_{\lambda \in \Lambda_p(i^\star)} \sum_{k \in [K]} \omega_s^k \, \mathrm{KL}_k(\theta, \lambda) \\
&\overset{(c)}{\geq} \sum_{s=\bar{t}+1}^{t} \min_{p \in \mathcal{P}_t^{\mathrm{smp}}(i^\star(\hat{\theta}_{s-1}))} \inf_{\lambda \in \Lambda_p(i^\star(\hat{\theta}_{s-1}))} \sum_{k \in [K]} \omega_s^k \, \mathrm{KL}_k(\theta, \lambda) \\
&\overset{(d)}{\geq} \sum_{s=1}^{t} \min_{p \in \mathcal{P}_{s-1}^{\mathrm{smp}}(i^\star(\hat{\theta}_{s-1}))} \inf_{\lambda \in \Lambda_p(i^\star(\hat{\theta}_{s-1}))} \sum_{k \in [K]} \omega_s^k \, \mathrm{KL}_k(\theta, \lambda) - B\sqrt{t} \\
&\overset{(e)}{\geq} \sum_{s=1}^{t} \min_{p \in \mathcal{P}_{s-1}^{\mathrm{smp}}(i^\star(\hat{\theta}_{s-1}))} \inf_{\lambda \in \Lambda_p(i^\star(\hat{\theta}_{s-1}))} \sum_{k \in [K]} \omega_s^k \left(\mathrm{KL}_k(\hat{\theta}_{s-1}, \lambda) - c_{s-1}^k \right) - B\sqrt{t} \\
&\overset{(f)}{\geq} \sum_{s=1}^{t} \min_{p \in \mathcal{P}_{s-1}^{\mathrm{smp}}(i^\star(\hat{\theta}_{s-1}))} \inf_{\lambda \in \Lambda_p(i^\star(\hat{\theta}_{s-1}))} \sum_{k \in [K]} \omega_s^k \left(\mathrm{KL}_k(\hat{\theta}_{s-1}, \lambda) + c_{s-1}^k \right) - 2C_{\mathrm{conf}}(t) - B\sqrt{t},
\end{aligned}$$

where (a) is from the concavity of the infimum, (b) drops the fist $\bar{t}$ rounds, (c) uses Lemma F.6, (d) uses that $\mathcal{P}_t^{\mathrm{smp}}$ is contained in all active sets from $\bar{t}$ to t and completes the sum with the first $\bar{t}$ rounds (while bounding $\bar{t} \leq \sqrt{t}$), (e) is from the validity of the confidence intervals under E_t, and (f) is from Assumption F.3.

Now note that, when applying the game-theoretic approach (Section F.2.2), the first term on the right-hand side is exactly the sum of gains fed into the learner. Thus, using the no-regret property

(Assumption F.4),

$$\min_{p\in\mathcal{P}_t^{\mathrm{smp}}(i^\star)} \inf_{\lambda\in\Lambda_p(i^\star)} \sum_{k\in[K]} \sum_{s=1}^{t} \omega_s^k \,\mathrm{KL}_k(\theta,\lambda)$$
$$\geq \max_{\omega\in\Delta_K} \sum_{s=1}^{t} \min_{p\in\mathcal{P}_{s-1}^{\mathrm{smp}}(i^\star(\hat\theta_{s-1}))} \inf_{\lambda\in\Lambda_p(i^\star(\hat\theta_{s-1}))} \sum_{k\in[K]} \omega^k \left(\mathrm{KL}_k(\hat\theta_{s-1},\lambda)+c_{s-1}^k\right)$$
$$- 2C_{\mathrm{conf}}(t) - C_{\mathcal{L}}(t) - B\sqrt{t}.$$

If instead we are applying optimistic Track-and-Stop (Section F.2.1), the first term on the right-hand side is exactly the sum of optimal values of the objective functions maximized by the algorithm. Thus,

$$\min_{p\in\mathcal{P}_t^{\mathrm{smp}}(i^\star)} \inf_{\lambda\in\Lambda_p(i^\star)} \sum_{k\in[K]} \sum_{s=1}^{t} \omega_s^k \,\mathrm{KL}_k(\theta,\lambda)$$
$$\geq \sum_{s=1}^{t} \max_{\omega\in\Delta_K} \min_{p\in\mathcal{P}_{s-1}^{\mathrm{smp}}(i^\star(\hat\theta_{s-1}))} \inf_{\lambda\in\Lambda_p(i^\star(\hat\theta_{s-1}))} \sum_{k\in[K]} \omega^k \left(\mathrm{KL}_k(\hat\theta_{s-1},\lambda)+c_{s-1}^k\right) - 2C_{\mathrm{conf}}(t) - B\sqrt{t}$$
$$\geq \max_{\omega\in\Delta_K} \sum_{s=1}^{t} \min_{p\in\mathcal{P}_{s-1}^{\mathrm{smp}}(i^\star(\hat\theta_{s-1}))} \inf_{\lambda\in\Lambda_p(i^\star(\hat\theta_{s-1}))} \sum_{k\in[K]} \omega^k \left(\mathrm{KL}_k(\hat\theta_{s-1},\lambda)+c_{s-1}^k\right) - 2C_{\mathrm{conf}}(t) - B\sqrt{t}.$$

Therefore, we only need to lower bound the first term above, which is common between the two considered algorithms. We have

$$\max_{\omega\in\Delta_K} \sum_{s=1}^{t} \min_{p\in\mathcal{P}_{s-1}^{\mathrm{smp}}(i^\star(\hat\theta_{s-1}))} \inf_{\lambda\in\Lambda_p(i^\star(\hat\theta_{s-1}))} \sum_{k\in[K]} \omega^k \left(\mathrm{KL}_k(\hat\theta_{s-1},\lambda)+c_{s-1}^k\right)$$
$$\overset{(g)}{\geq} \max_{\omega\in\Delta_K} \sum_{s=\bar t+1}^{t} \min_{p\in\mathcal{P}_{s-1}^{\mathrm{smp}}(i^\star(\hat\theta_{s-1}))} \inf_{\lambda\in\Lambda_p(i^\star(\hat\theta_{s-1}))} \sum_{k\in[K]} \omega^k \left(\mathrm{KL}_k(\hat\theta_{s-1},\lambda)+c_{s-1}^k\right)$$
$$\overset{(h)}{\geq} \max_{\omega\in\Delta_K} \sum_{s=\bar t+1}^{t} \min_{p\in\mathcal{P}_{s-1}^{\mathrm{smp}}(i^\star)} \inf_{\lambda\in\Lambda_p(i^\star)} \sum_{k\in[K]} \omega^k \left(\mathrm{KL}_k(\hat\theta_{s-1},\lambda)+c_{s-1}^k\right)$$
$$\overset{(i)}{\geq} \max_{\omega\in\Delta_K} \sum_{s=\bar t+1}^{t} \min_{p\in\mathcal{P}_{s-1}^{\mathrm{smp}}(i^\star)} \inf_{\lambda\in\Lambda_p(i^\star)} \sum_{k\in[K]} \omega^k \,\mathrm{KL}_k(\theta,\lambda)$$
$$\overset{(j)}{\geq} \max_{\omega\in\Delta_K} \sum_{s=1}^{t} \min_{p\in\mathcal{P}_{s-1}^{\mathrm{smp}}(i^\star)} \inf_{\lambda\in\Lambda_p(i^\star)} \sum_{k\in[K]} \omega^k \,\mathrm{KL}_k(\theta,\lambda) - B\sqrt{t},$$

where (g) drops the first $\bar t$ rounds, (h) uses Lemma F.7, (i) is from the fact that confidence intervals are valid under E_t, and (j) adds the missing first $\bar t$ rounds. Putting all together, we proved that optimistic Track-and-Stop satisfies Assumption 2.5 with

$$R(\theta,t) = C_{\mathrm{track}}KB + 2C_{\mathrm{conf}}(t) + 2B\sqrt{t},$$

while the game-theoretic approach satisfies it with

$$R(\theta,t) = C_{\mathrm{track}}KB + 2C_{\mathrm{conf}}(t) + C_{\mathcal{L}}(t) + 2B\sqrt{t}.$$

Lemma F.5. *Under event E_t, for any $s\in\mathbb{N}$ with $\bar t_{j(t)-1}\leq s\leq t$,*

$$L_s(\hat\theta_s,\theta) < \alpha_{s,\delta}.$$

Moreover, for any $s,s'\in\mathbb{N}$ with $\bar t_{j(t)-1}\leq s'\leq s\leq t$,

$$L_{s'}(\hat\theta_{s'},\hat\theta_s) < \alpha_{s',\delta}.$$

Proof. Using the good event E_t followed by an application of Lemma E.2 together with $s \geq \bar{t}_{j(t)-1}$,

$$L_s(\hat{\theta}_s, \theta) \leq \beta_{t,1/t^2} \leq 4\frac{c_2}{c_1}\beta_{\bar{t}_{j(t)-1},1/\bar{t}^2_{j(t)-1}} \leq 4\frac{c_2}{c_1}\beta_{s,1/s^2} < \alpha_{s,\delta}.$$

This proves the first claim. To prove the second one, note that

$$\begin{aligned}
\sqrt{L_{s'}(\hat{\theta}_{s'}, \hat{\theta}_s)} &\overset{(a)}{=} \sqrt{\sum_{k\in[K]} N^k_{s'} \operatorname{KL}_k(\hat{\theta}_{s'}, \hat{\theta}_s)} \overset{(b)}{\leq} \sqrt{\sum_{k\in[K]} N^k_{s'} \operatorname{KL}_k(\hat{\theta}_{s'}, \theta)} + \sqrt{\sum_{k\in[K]} N^k_{s'} \operatorname{KL}_k(\hat{\theta}_s, \theta)} \\
&\overset{(c)}{\leq} \sqrt{\sum_{k\in[K]} N^k_{s'} \operatorname{KL}_k(\hat{\theta}_{s'}, \theta)} + \sqrt{\sum_{k\in[K]} N^k_{s} \operatorname{KL}_k(\hat{\theta}_s, \theta)} \\
&\overset{(d)}{=} \sqrt{L_{s'}(\hat{\theta}_{s'}, \theta)} + \sqrt{L_s(\hat{\theta}_s, \theta)} \overset{(e)}{\leq} 2\sqrt{\beta_{t,1/t^2}},
\end{aligned}$$

where (a) is from Corollary C.6, (b) is from the triangle inequality (recall that the sum of KLs is a norm), (c) is from the fact that the pull counts are non-decreasing and $s \geq s'$, (d) is again from Corollary C.6, and (e) is from event E_t. Using Lemma E.2 as before, we have $2\sqrt{\beta_{t,1/t^2}} \leq 2\sqrt{\frac{c_2}{c_1}\beta_{s',1/s'^2}} < \sqrt{\alpha_{s',\delta}}$. This proves the second statement. □

Lemma F.6. *Under event E_t, for any $i \in \mathcal{I}$ and $\omega \in \Delta_K$,*

$$\min_{p\in\mathcal{P}^{\text{smp}}_t(i^\star)} \inf_{\lambda\in\Lambda_p(i^\star)} \sum_{k\in[K]} \omega^k \operatorname{KL}_k(\theta, \lambda) \geq \min_{p\in\mathcal{P}^{\text{smp}}_t(i)} \inf_{\lambda\in\Lambda_p(i)} \sum_{k\in[K]} \omega^k \operatorname{KL}_k(\theta, \lambda).$$

Proof. The statement follows trivially if $i = i^\star$. So suppose $i \neq i^\star$. Since i is not the answer of θ, by the union property of the decomposition into pieces, there exists $p \in \mathcal{P}(i)$ such that $\theta \in \Lambda_p(i)$. By Lemma F.5, for any $\bar{t}_{j(t)-1} \leq s \leq t$,

$$\inf_{\lambda\in\Lambda_p(i)} L_s(\hat{\theta}_s, \lambda) \leq L_s(\hat{\theta}_s, \theta) < \alpha_{s,\delta}.$$

This implies that $p \in \mathcal{P}^{\text{smp}}_t(i)$ since such set is defined as the intersection of all active sets from $\bar{t}_{j(t)-1}$ to t. Finally, we conclude that

$$\min_{p\in\mathcal{P}^{\text{smp}}_t(i)} \inf_{\lambda\in\Lambda_p(i)} \sum_{k\in[K]} \omega^k \operatorname{KL}_k(\theta, \lambda) \leq \sum_{k\in[K]} \omega^k \operatorname{KL}_k(\theta, \theta) = 0,$$

and thus our result follows trivially. □

Lemma F.7. *Under event E_t, for any $\bar{t}_{j(t)-1} \leq s \leq t$, $i \in \mathcal{I}$, and $\omega \in \Delta_K$,*

$$\min_{p\in\mathcal{P}^{\text{smp}}_s(i^\star(\hat{\theta}_s))} \inf_{\lambda\in\Lambda_p(i^\star(\hat{\theta}_s))} \sum_{k\in[K]} \omega^k \operatorname{KL}_k(\hat{\theta}_s, \lambda) \geq \min_{p\in\mathcal{P}^{\text{smp}}_s(i)} \inf_{\lambda\in\Lambda_p(i)} \sum_{k\in[K]} \omega^k \operatorname{KL}_k(\hat{\theta}_s, \lambda).$$

Proof. The proof is very similar to the one of Lemma F.6. The statement follows trivially if $i = i^\star(\hat{\theta}_s)$. So suppose $i \neq i^\star(\hat{\theta}_s)$. Since i is not the answer of $\hat{\theta}_s$, by the union property of the decomposition into pieces, there exists $p \in \mathcal{P}(i)$ such that $\hat{\theta}_s \in \Lambda_p(i)$. By Lemma F.5, for any $\bar{t}_{j(t)-1} \leq s' \leq s \leq t$,

$$\inf_{\lambda\in\Lambda_p(i)} L_{s'}(\hat{\theta}_{s'}, \lambda) \leq L_{s'}(\hat{\theta}_{s'}, \hat{\theta}_s) < \alpha_{s',\delta}.$$

This implies that $p \in \mathcal{P}^{\text{smp}}_s(i)$ since such set is defined as the intersection of all active sets from $\bar{t}_{j(t)-1}$ to s. Therefore, we conclude that

$$\min_{p\in\mathcal{P}^{\text{smp}}_s(i)} \inf_{\lambda\in\Lambda_p(i)} \sum_{k\in[K]} \omega^k \operatorname{KL}_k(\hat{\theta}_s, \lambda) \leq \sum_{k\in[K]} \omega^k \operatorname{KL}_k(\hat{\theta}_s, \hat{\theta}_s) = 0,$$

and thus our result follows trivially. □

G Experiment Details and Additional Results

G.1 Reproducibility details

We provide the main details to reproduce our experiments. For all the details, we refer the reader to our implementation at `https://github.com/AndreaTirinzoni/bandit-elimination`.

In all experiments, we used $\delta = 0.01$ and a heuristic threshold $\beta_{t,\delta} = \log(1/\delta) + \log(1+t)$ for all elimination rules and LLR stopping. This is slightly larger than the heuristic threshold proposed by [4] and adopted in many recent works. We implemented the elimination rules as described in Appendix B. When using elimination at both stopping and sampling, we maintained only one set of active pieces (the one for stopping) instead of keeping a separate set with very lazy resets for sampling as suggested by theory. That set is shared by both sampling and stopping rules, and is never reset.

Computational infrastructure All experiments were run on a Dell XPS 13 laptop with an Intel Core i7-7560U (2.40GHz) CPU and 8GB of RAM.

G.2 Bandit instances

We provide details on how we generated the bandit instances considered in the experiments presented in the main paper and later in this section.

Linear instances (experiments of Figure 2 and Table 3) We set $K = 50$ and $d = 10$. The true parameter is $\theta = [1, 1, \dots, 1]^T$, while we generated the arm features randomly. The arm feature of the first arm is $\phi_1 = [1, 0, 0, \dots, 0]^T$. Then, up to reaching 50 arms, we repeated the following procedure. First, we generated a 3-dimensional vector $v \in \mathbb{R}^3$ by drawing its elements uniformly in $[-1, 1]^3$ and then normalizing to have unit norm. Then, we added 3 feature vectors $v_1 = [0, v, 0, 0, 0, 0, 0, 0]$, $v_2 = [0, 0, 0, 0, v, 0, 0, 0]$, and $v_3 = [0, 0, 0, 0, 0, 0, 0, v]$, but only if $v_1^T\theta \in [0, 0.8]$. In this way, we obtain linear instances with 50 arms where arm 1 is optimal with value $\mu_1(\theta) = 1$, while all other arms have minimum sub-optimality of 0.2 and maximum sub-optimality gap of 0.8.

Linear instances (experiments of Table 1) We set $K = 50$ and $d = 20$. The first 10 arms are set to the canonical basis of $\mathbb{R}^{10}$, i.e., $\phi_k = e_k$ for $k = 1, \dots, 10$. The generation of the true parameter θ and of the remaining 40 arms is slightly different from BAI/Top-m and OSI.

For BAI and Top-m, the true parameter θ has the first element equal to 1, elements from the second to the fifth equal to 0.9, and elements from the sixth to the tenth equal to 0.8. The remaining 10 elements are uniformly drawn in $[-0.5, 0.5]^{10}$. The remaining 40 arms are randomly generated as follows. First, we draw a vector v uniformly in $[-1, 1]^{20}$ and normalize it to have unit norm. Then, if $v^T\theta \leq 0.5$, we add v to the set of arms. Otherwise, we reject the vector and keep repeating this procedure until we reach a total of 50 arms. In this way, we obtain random linear instances where the first arm is optimal with value $\mu_1(\theta) = 1$, the next 9 arms are hard to discriminate from it since they have small gap (either 0.1 or 0.2), and all remaining 40 arms have moderate to large gap (at least 0.5) and are thus easy to eliminate.

For OSI, the true parameter θ has the first ten elements uniformly drawn in $([-0.2, -0.1] \cup [0.1, 0.2])^{10}$ and the second ten elements uniformly drawn in $[-0.5, 0.5]^{10}$. Similarly as before, to generate the remaining 40 arms we first draw a vector v uniformly in $[-1, 1]^{20}$ and normalize it to have unit norm. Then, if $|v^T\theta| \geq 0.5$, we add v to the set of arms. Otherwise, we reject the vector and keep repeating this procedure until we reach a total of 50 arms. We thus obtain random linear instances where the first 10 arms are hard to learn since they have small gap (i.e., the absolute mean, which is between 0.1 and 0.2), and all remaining 40 arms have moderate to large gap (at least 0.5) and are thus easy to eliminate.

Unstructured instances (experiments of Appendix G.3.2) We used $K = 40$ arms. For BAI and Top-m, the mean reward of the first 5 arms is $\mu_1 = 1$, $\mu_2 = 0.9$, $\mu_3 = 0.8$, $\mu_4 = 0.7$, and $\mu_5 = 0.6$. For all remaining arms the mean reward is uniformly drawn in $[0, 0.5]$. For OSI, the mean reward of the first 4 arms is $\mu_1 = 0.1$, $\mu_2 = -0.2$, $\mu_3 = 0.3$, and $\mu_4 = -0.4$. For all remaining arms the mean reward is uniformly drawn in $[-1, -0.5] \cup [0.5, 1]$.

	Algorithm	No elim. (LLR)		Selective elim.		Full elim.	
		Samples	Time	Samples	Time	Samples	Time
BAI	LinGapE	4.51 ± 1.3	0.19	4.49 ± 1.3	0.17	4.49 ± 1.3	0.57
	LinGapE + elim			4.16 ± 1.4	0.15	4.17 ± 1.4	0.58
	LinGame	5.28 ± 1.7	0.21	5.09 ± 1.8	0.19	5.09 ± 1.8	0.6
	LinGame + elim			4.05 ± 1.2	0.17	4.05 ± 1.2	0.65
	FWS	4.68 ± 4.2	0.84	4.68 ± 4.2	0.82	4.68 ± 4.2	1.39
	FWS + elim			4.21 ± 1.4	0.58	4.21 ± 1.4	1.16
	Lazy TaS	9.99 ± 8.8	0.45	9.75 ± 8.9	0.45	9.75 ± 8.9	0.78
	Lazy TaS + elim			8.7 ± 8.9	0.38	8.7 ± 8.9	0.73
	Oracle	6.65 ± 1.8	0.04	6.55 ± 1.9	0.02	6.55 ± 1.9	0.31
	XY-Adaptive					13.89 ± 6.0	2.23
	RAGE					16.28 ± 6.2	0.02
Top-m ($m = 3$)	m-LinGapE	6.26 ± 1.2	0.29	6.21 ± 1.2	0.24	6.21 ± 1.2	1.35
	m-LinGapE + elim			5.77 ± 1.2	0.19	5.77 ± 1.2	1.29
	MisLid	7.06 ± 1.4	0.34	6.81 ± 1.5	0.27	6.81 ± 1.5	1.48
	MisLid + elim			5.89 ± 1.1	0.22	5.89 ± 1.1	1.42
	FWS	5.91 ± 1.7	1.51	5.9 ± 1.7	1.46	5.9 ± 1.7	2.66
	FWS + elim			5.84 ± 1.7	0.83	5.84 ± 1.7	2.02
	Lazy TaS	13.1 ± 6.5	0.71	12.85 ± 6.4	0.67	12.85 ± 6.4	1.57
	Lazy TaS + elim			11.34 ± 6.3	0.56	11.34 ± 6.3	1.47
	Oracle	8.74 ± 1.8	0.1	8.65 ± 1.8	0.04	8.65 ± 1.8	1.02
	LinGIFA	5.58 ± 1.1	1.8	5.57 ± 1.1	1.75	5.57 ± 1.1	2.68

Table 3: Experiments on linear instances with $K = 50$ and $d = 10$. The "Time" columns report average times per iteration in milliseconds (i.e., the total time the algorithm took divided by the number of samples). Each entry reports the mean across 100 runs plus/minus standard deviation (which is omitted for compute times due to space constraints). The "+ elim" variant of some algorithms indicates that the corresponding sampling rule is combined with elimination. Samples are scaled down by a factor 10^3.

G.3 Additional Results

G.3.1 Full versus selective elimination

We report in Table 3 the full results of the experiment comparing elimination rules (full vs selective) from which we extracted Figure 2*(middle)*. We recall that the linear instances for this experiment were generated as explained in the first paragraph of Appendix G.2. We did not compare full and selective elimination rules on OSI since, as explained in Appendix B, they are actually equivalent in such a setting.

While we saw in Figure 2*(middle)* that the full elimination rule allows eliminating some arms earlier than the selective one, we notice from Table 3 that the former rule actually yields no advantage in terms of sample complexity. Moreover, its computational overhead makes it much slower than the selective elimination rule. Therefore, in practice we suggest using the selective elimination rule, which always yields reduced computation times and often improved sample complexity.

G.3.2 Unstructured instances

We report the results on unstructured bandit instances (generated according to the procedure of Appendix G.2) in Table 4. The algorithm k-Learner is the unstructured variant of LinGame proposed by [15]. We note that the results are coherent with those for linear instances presented in the main paper. In particular, we observe a reduction in computation times when combining adaptive algorithms with selective elimation. The reduction is however less evident than in the linear case. This is expected since, in general, eliminations are easier in structured problems than in unstructured ones. We also note that combining sampling rules with selective elimination slightly improves the sample complexity of all algorithms.

	Algorithm	No elim. (LLR)		Elim. stopping		Elim. stopping + sampling	
		Samples	Time	Samples	Time	Samples	Time
BAI	k-Learner	18.76 ± 6.5	0.49	18.12 ± 6.6	0.44	14.82 ± 4.6	0.4
	FWS	14.5 ± 4.6	1.25	14.44 ± 4.7	1.24	13.94 ± 4.4	1.16
	Lazy TaS	26.18 ± 8.0	0.32	24.78 ± 7.7	0.31	20.66 ± 7.4	0.33
	Oracle	27.49 ± 3.7	0.07	27.0 ± 3.7	0.04		
	LUCB	14.2 ± 5.2	0.11	14.18 ± 5.2	0.06	13.57 ± 4.7	0.06
	UGapE	15.13 ± 5.0	0.43	15.13 ± 5.0	0.39		
	Racing					34.55 ± 7.6	0.01
Top-m ($m = 3$)	k-Learner	25.84 ± 6.2	0.67	25.06 ± 6.3	0.57	17.65 ± 4.9	0.51
	FWS	17.68 ± 4.7	2.52	17.67 ± 4.7	2.48	17.63 ± 4.6	2.0
	Lazy TaS	38.89 ± 10.4	0.5	37.84 ± 10.7	0.43	27.74 ± 6.4	0.46
	Oracle	34.17 ± 4.5	0.15	33.68 ± 4.9	0.07		
	LUCB	17.61 ± 4.6	0.24	17.58 ± 4.5	0.13	17.15 ± 5.4	0.13
	UGapE	17.87 ± 4.3	0.54	17.87 ± 4.3	0.43		
	Racing					22.67 ± 3.1	0.01
OSI	k-Learner	8.55 ± 1.7	0.61	8.38 ± 1.8	0.55	5.43 ± 1.2	0.52
	FWS	5.54 ± 1.3	1.48	5.53 ± 1.3	1.47	5.47 ± 1.4	1.4
	Lazy TaS	12.83 ± 3.1	0.72	12.27 ± 3.1	0.7	8.74 ± 1.8	0.79
	Oracle	11.55 ± 1.6	0.07	11.41 ± 1.6	0.04		
	LUCB	5.5 ± 1.4	0.11	5.5 ± 1.4	0.1	5.49 ± 1.4	0.1

Table 4: Experiments on unstructured instances with $K = 40$. The "Time" columns report average times per iteration in milliseconds (i.e., the total time the algorithm took divided by the number of samples). Each entry reports the mean across 100 runs plus/minus standard deviation (which is omitted for compute times due to space constraints). Algorithms for which the third column is missing cannot be combined with elimination at sampling, while algorithms for which the first two columns are missing are natively elimination-based. Samples are scaled down by a factor 10^3.

H Examples

H.1 Running example

Consider BAI in a Gaussian linear bandit instance with unit variance, $d = 2$, and arbitrary number of arms $K \geq 3$ (see Figure 3 for an example with $K = 5$). The arm features are $\phi_1 = (1, 0)^T$, $\phi_2 = (0, 1)^T$, and, for all $i = 3, \dots, K$, $\phi_i = (a_i, b_i)^T$ with a_i, b_i arbitrary values in $(-1, 0)$ such that $\|\phi_i\|_2 = 1$. The true parameter is $\theta = (1, 1 - \varepsilon)^T$, for $\varepsilon \in (0, 1/2)$ a possibly very small value. Arm 1 is optimal with mean $\mu_1(\theta) = 1$, while arm 2 is sub-optimal with mean $\mu_2(\theta) = 1 - \varepsilon$. For all other arms $i = 3, \dots, K$, $\mu_i(\theta) \leq 0$.

Let $\omega \in \Delta_K$ be any allocation. Recall that in BAI each piece index is simply an arm, and $\mathcal{P}(i^\star(\theta)) = \mathcal{P}(1) = \{2, \dots, K\}$. Let $k \in \mathcal{P}(1)$ be any sub-optimal arm. The distance to the k-th alternative piece $H_k(\omega, \theta)$ can be computed in closed form as

$$H_k(\omega, \theta) = \frac{((\phi_1 - \phi_k)^T \theta)^2}{2\|\phi_1 - \phi_k\|^2_{V_\omega^{-1}}}.$$

It can be determined that the optimal allocation, solution to $\arg\max_\omega \min_k H_k(\omega, \theta)$, is $\omega^\star = (1/2, 1/2, 0, \dots, 0)^\top$. We will prove this as a consequence of Lemma H.1 below.

The intuition why this example is interesting is as follows. Any correct strategy is required to discriminate between arm 1 and 2 (i.e., to figure out that arm 1 is optimal), which requires roughly $O(1/\varepsilon^2)$ samples from both. An optimal strategy plays these two arms nearly with the same proportions. Since ϕ_1 and ϕ_2 form the canonical basis of $\mathbb{R}^2$, the samples collected by this strategy are informative for estimating the mean reward of *every* arm, even those than are not played. Then, since arms $3, \dots, K$ have at least a sub-optimality gap of 1, an elimination-based strategy discards them with a number of samples not scaling with $1/\varepsilon^2$. This means that a non-elimination strategy runs for $O(1/\varepsilon^2)$ steps over the original problem with K arms, while an elimination-based one quickly reduces the problem to one with only 2 arms. The main impact is computational: since most algorithms need to compute some statistics for each active arm at each round (e.g., closest

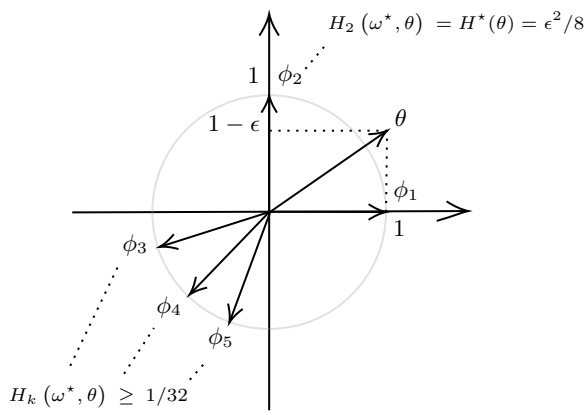

Figure 3: BAI instance with $d = 2$ and $K = 5$. Arm 1 is optimal with a mean reward of 1, arm 2 is ϵ-suboptimal, while all other arms have a large sub-optimality gap. The optimal allocation $\omega^\star = (1/2, 1/2, 0, \dots)^T$ plays only arm 1 and arm 2 with the same proportions. The distance $H_2(\omega^\star, \theta)$ to the alternative piece of the second arm scales as $O(\epsilon^2)$ and fully controls the optimal sample complexity, which in turn is $\Omega(1/\epsilon^2)$. On the other hand, the distance $H_k(\omega^\star, \theta)$ to the alternative piece of every other arm $k > 2$ is large ($\Omega(1)$), which implies that such arms can be eliminated with sample complexity not scaling with $1/\epsilon^2$.

alternatives, confidence intervals, etc.) and that requires at least one loop over the set of active arms, the computational complexity of a non-elimination algorithm is at least $O(K/\varepsilon^2)$, while the one of an elimination-based variant is roughly $O(K + 1/\varepsilon^2)$, a potentially very large improvement.

Remark. *The fact that certain arms are discarded very early is not only due to their large sub-optimality gap but also to the linear structure of the problem (i.e., to the fact that pulling certain arms provides information about others). In fact, if we consider an equivalent bandit problem with the same mean rewards but with K-dimensional features such that $\phi_k = e_k$ (the k-th vector of the canonical basis of $\mathbb{R}^K$) for all k, then pulling an arm provides no information about the others. In this unstructured case, it is known that an optimal allocation $\omega^\star$ makes all the half-spaces at the same distance. That is, $H_k(\omega, \theta)$ is the same for all k and thus no arm is eliminated early.*

The following result formalizes the intuition that arms $3, \dots, K$ can be eliminated much earlier than arm 2. It shows that, for any allocation ω and $k > 2$, the distance $H_k(\omega, \theta)$ to the k-th alternative piece is at least a factor $1/\varepsilon^2$ larger than the distance $H_2(\omega, \theta)$ to the second alternative piece.

Lemma H.1. *For any ω and $k = 3, \dots, K$, $H_k(\omega, \theta) \geq \frac{(1-\varepsilon)^2}{\varepsilon^2} H_2(\omega, \theta)$.*

Proof. For $k > 2$, $\Lambda_k(i^\star) \subseteq \{\eta \mid \eta_1 \leq 0\} \cup \{\eta \mid \eta_2 \leq 0\}$. Hence the distance from θ to $\Lambda_k(i^\star)$ in V_ω-norm is greater than the minimal distance to each of those two half spaces. The closest points to θ in those two half spaces are $(0, 1-\varepsilon)^\top$ and $(1, 0)^\top$, respectively. We get

$$
\begin{aligned}
H_k(\omega, \theta) &\geq \min\left\{\frac{1}{2}\|\theta - (0, 1-\varepsilon)^\top\|_{V_\omega}^2, \frac{1}{2}\|\theta - (1, 0)^\top\|_{V_\omega}^2\right\} \\
&= \frac{1}{2}\min\{\|\phi_1\|_{V_\omega}^2, (1-\varepsilon)^2\|\phi_2\|_{V_\omega}^2\} \geq \frac{1}{2}(1-\varepsilon)^2 \min\{\|\phi_1\|_{V_\omega}^2, \|\phi_2\|_{V_\omega}^2\}.
\end{aligned}
$$

Since we have 2×2 matrices, let us get concrete and write $V_\omega = \begin{pmatrix} a & b \\ b & d \end{pmatrix}$. By our hypotheses on all ϕ_k, $b \geq 0$. . We get that $d = 1 - a$ by computing the trace of V_ω: $\mathrm{tr}(V_\omega) = \mathrm{tr}(\sum_k \omega^k \phi_k \phi_k^\top) = \sum_k \omega^k \mathrm{tr}(\phi_k \phi_k^\top) = \sum_k \omega^k \phi_k^\top \phi_k = 1$.

Then $V_\omega^{-1} = \frac{1}{a(1-a)-b^2}\begin{pmatrix} 1-a & -b \\ -b & a \end{pmatrix}$, $\|\phi_1\|^2_{V_\omega} = a$, $\|\phi_2\|^2_{V_\omega} = 1-a$, $\|\phi_1 - \phi_2\|^2_{V_\omega^{-1}} = \frac{1}{a(1-a)-b^2}(1+2b)$. Thus,

$$H_2(\omega,\theta) = \frac{\varepsilon^2}{2\|\phi_1-\phi_2\|^2_{V_\omega^{-1}}} = \frac{\varepsilon^2(a(1-a)-b^2)}{2(1+2b)} \leq \frac{\varepsilon^2}{2}a(1-a),$$

and

$$H_k(\omega,\theta) \geq \frac{1}{2}(1-\varepsilon)^2 \min\{\|\phi_1\|^2_{V_\omega}, \|\phi_2\|^2_{V_\omega}\} = \frac{1}{2}(1-\varepsilon)^2\min\{1-a,a\} \geq \frac{1}{2}(1-\varepsilon)^2 a(1-a).$$

We finally obtain a comparison between $H_k(\omega,\theta)$ for $k>2$ and $H_2(\omega,\theta)$:

$$H_k(\omega,\theta) \geq \frac{1}{2}(1-\varepsilon)^2 a(1-a) \geq \frac{(1-\varepsilon)^2}{\varepsilon^2} H_2(\omega,\theta)\,.$$

□

Corollary H.2. *The optimal allocation for this problem is* $\omega^\star = (1/2, 1/2, 0, \ldots, 0)^\top$.

Proof. By Lemma H.1, since $\varepsilon < 1/2$ the minimum over H_k is attained by H_2 no matter what ω is, hence the optimal allocation belongs to $\arg\max_\omega H_2(\omega,\theta)$. With the notation of that lemma, $H_2(\omega,\theta) = \frac{\varepsilon^2(a(1-a)-b^2)}{2(1+2b)}$ where $a \in [0,1], b \geq 0$ are coefficients of V_ω. We get that (if that's attainable), the optimal value is reached for $a = 1/2, b = 0$. By construction of our arms ϕ_k, $b = 0$ implies that ω is supported on the two first coordinates. $a = 1/2$ can then be attained by $(1/2, 1/2, 0, \ldots, 0)^\top$. □

H.2 An oracle strategy with elimination at stopping provably reduces the computational complexity w.r.t. LLR stopping

Consider an "oracle" strategy which tracks the optimal proportions $\omega^\star$ from the lower bound, i.e., the arm played at time t is $k_t = \arg\min_k (N^k_{t-1} - t\omega^\star_k)$.

Proposition H.3. *For any $K \geq 3$ and $\varepsilon \in (0, 1/2)$, the exists a Gaussian linear BAI instance with unit variance and $d = 2$ such that, for any $\delta \in (0,1)$, the oracle strategy combined with LLR stopping satisfies*

$$\mathbb{E}[\tau] \geq \Omega\left(\frac{\log(1/\delta)}{\varepsilon^2}\right).$$

On the same instance, suppose we run the oracle strategy with elimination at stopping using a threshold $\beta_{t,\delta} = \log(1/\delta) + O(\log(t))$. Then, for all but 2 arms, the expected elimination time of the corresponding piece is, for full elimination,

$$\mathbb{E}[\tau_k] \leq \widetilde{O}(\log(1/\delta)),$$

while for selective elimination,

$$\mathbb{E}[\tau_k] \leq \widetilde{O}\left(\log(1/\delta) + \frac{1}{\varepsilon^2}\right).$$

Here $\widetilde{O}$ hides constant and logarithmic terms.

Proof. Consider the problem defined in the example of Appendix H.1. It is easy to see that, on such instance, the oracle sampling rule simply alternates between pulling arm 1 and arm 2. The first result is immediate from the lower bound $\mathbb{E}[\tau] \geq \log\left(\frac{1}{2.4\delta}\right)/H^\star(\theta)$. We only need to compute $H^\star(\theta) = \max_\omega \min_k H_k(\omega,\theta)$. By Lemma H.1, since $\varepsilon < 1/2$ the minimum is equal to $H_2(\omega,\theta)$. Since $\omega^\star = (1/2, 1/2, 0, \ldots, 0)^T$, using the closed-form expression of the infimum over each alternative piece as stated above,

$$H^\star(\theta) = \max_\omega H_2(\omega,\theta) = \frac{1}{2}\frac{\varepsilon^2}{\frac{1}{\omega^\star_1} + \frac{1}{\omega^\star_2}} = \frac{\varepsilon^2}{8}\,.$$

This proves the first result. Let us now prove the second one. We use the upper bound on the elimination time of pieces from Theorem 2.6. Note that, for each time t even, we have that $N_t/t = \omega^\star$, which implies that $R(\theta, t) = 0$. Therefore, for each $k = 3, \dots, K$, $\mathbb{E}[\tau_k] \leq \bar{t}_k + 3$, where $\bar{t}_k$ is the first integer such that

$$t \geq \max\left\{ \frac{\left(\sqrt{\beta_{t,\delta}} + \sqrt{\beta_{t,1/t^2}}\right)^2}{H_k(\omega^\star, \theta)}, G(\theta, t) \right\}, \tag{14}$$

with $G(\theta, t) = 0$ for full elimination and $G(\theta, t) = \frac{4\beta_{t,1/t^2}}{H^\star(\theta)}$ for selective elimination. For all $k = 3, \dots, K$, using Lemma H.1,

$$H_k(\omega^\star, \theta) \geq \frac{(1-\varepsilon)^2}{\varepsilon^2} H_2(\omega^\star, \theta) = \frac{(1-\varepsilon)^2}{8} \geq \frac{1}{32}\,.$$

Now, for full elimination, since $\beta_{t,\delta} = \log(1/\delta) + O(\log(t))$ and $H_k(\omega^\star, \theta) \geq 1/32$ as proved above, $\bar{t}_k$ is below the first time t such that

$$t \geq 32 \log(1/\delta) + O(\log(t)),$$

which yields $\bar{t}_k \leq \widetilde{O}(\log(1/\delta))$. The same reasoning can be done for selective elimination, which yields the stated bound. □

An immediate consequence of Proposition H.3 is that the total computational complexity of the oracle strategy can be significantly reduced with elimination, especially in the regime where δ and ε are small.

Proposition H.4. *On the same instance constructed in Proposition H.3, the expected per-round computation time of the oracle strategy with LLR stopping is $\Omega(K)$, while it is at most $O(K^2\varepsilon^2)$ for full elimination and $O(K\varepsilon^2 + K/\log(1/\delta))$ for selective elimination.*

Proof. The per-round complexity of the oracle strategy with LLR stopping is $\Omega(K)$ since checking the stopping rule requires computing the infimum LLR for exactly $K-1$ alternative pieces at each step. Full elimination requires two nested loops over the set of active arms, which take at most $O(K^2)$ complexity (when no arm has been eliminated). By Proposition H.3, this set is of size at most K only until time $\widetilde{O}(\log(1/\delta))$, while it is of cardinality at most 2 after that. Since the total sample complexity is $\widetilde{O}(\log(1/\delta)/\varepsilon^2)$ (the same as LLR stopping), the total computation time is $O(K^2 \log(1/\delta) + \log(1/\delta)/\varepsilon^2)$ and the average one is thus the stated quantity. Similarly, selective elimination requires a single loop over all active arms at each step, which takes at most $O(K)$ complexity. Again, by Proposition H.3, this set is of size at most K only until time $\widetilde{O}(\log(1/\delta) + 1/\varepsilon^2)$, while it is of cardinality at most 2 after that. Therefore, the total computation complexity is $O(K\log(1/\delta) + K/\varepsilon^2 + \log(1/\delta)/\varepsilon^2)$. Dividing by the total number of rounds $\log(1/\delta)/\varepsilon^2$ gives the stated bound. □

H.3 Low information regret with elimination at stopping provably discards many arms early

Proposition H.5. *For any $K \geq 3$ and $\varepsilon \in (0, 1/2)$, there exists a Gaussian linear BAI instance with unit variance and $d = 2$ such that, for any $\delta \in (0, 1)$, any low information regret sampling rule combined with LLR stopping satisfies*

$$\mathbb{E}[\tau] \geq \Omega\left(\frac{\log(1/\delta)}{\varepsilon^2}\right).$$

On the same instance, suppose we run the same strategy with elimination at stopping using a threshold $\beta_{t,\delta} = \log(1/\delta) + O(\log(t))$. Let $T_\varepsilon := \inf_t\{t : R(\theta, t)/t \leq \varepsilon^2/16\}$ (note that this quantity is constant in δ). Then, for all but 2 arms, the expected elimination time of the corresponding piece is, for full elimination,

$$\mathbb{E}[\tau_k] \leq \widetilde{O}(\log(1/\delta) + T_\varepsilon),$$

while for selective elimination,

$$\mathbb{E}[\tau_k] \leq \widetilde{O}\left(\log(1/\delta) + \frac{1}{\varepsilon^2} + T_\varepsilon\right).$$

Here $\widetilde{O}$ hides constant and logarithmic terms.

Remark. *On the instance constructed in Proposition H.5, we actually have $H^\star(\theta) = \varepsilon^2/8$ while, for any $\gamma \in (0, \varepsilon^2/16]$ and $k > 2$, $\min_{\omega \in \Omega_\gamma(\theta)} H_k(\omega, \theta) \geq 1/64$. This implies that, from Theorem 2.6, all but two arms are eliminated way before stopping by any low information regret sampling rule.*

Proof. Consider the problem defined in the example of Appendix H.1. The first result has already been proved in Proposition H.3. Let us prove the second one.

We use the upper bound on the elimination time of pieces from Theorem 2.6. For arm $k > 2$, the denominator scales as

$$\min_{\omega \in \Omega_{R(\theta,t)/t}(\theta)} H_k(\omega, \theta).$$

Suppose that $t \geq T_\varepsilon := \inf_t \{t : R(\theta, t)/t \leq \varepsilon^2/16\}$. Then, by definition of the set Ω, any $\omega \in \Omega_{R(\theta,t)/t}(\theta)$ satisfies

$$H_2(\omega, \theta) \geq \min_{k \neq 1} H_k(\omega, \theta) \geq H^\star(\theta) - \frac{R(\theta, t)}{t} \geq \varepsilon^2/16.$$

Combining this result with Lemma H.1,

$$\min_{\omega \in \Omega_{R(\theta,t)/t}(\theta)} H_k(\omega, \theta) \geq \frac{(1-\varepsilon)^2}{\varepsilon^2} \min_{\omega \in \Omega_{R(\theta,t)/t}(\theta)} H_2(\omega, \theta) \geq \frac{(1-\varepsilon)^2}{16} \geq 1/64.$$

Then, from Theorem 2.6 combined with the condition $t \geq T_\varepsilon$, $\mathbb{E}[\tau_k] \leq \bar{t}_k + 2 + T_\varepsilon$, where $\bar{t}_k$ is the first integer such that

$$t \geq \max \left\{ 64 \left(\sqrt{\beta_{t,\delta}} + \sqrt{\beta_{t,1/t^2}} \right)^2, G(\theta, t) \right\}, \tag{15}$$

with $G(\theta, t) = 0$ for full elimination and $G(\theta, t) = \frac{4\beta_{t,1/t^2}}{H^\star(\theta)}$ for selective elimination. We can now conclude as in the proof of Proposition H.3. □

I Additional details

Algorithm 2 LinGame [16]: vanilla (left) and with selective elimination (right)

while not stopped **do**

  for $i \neq i^\star(\hat{\theta}_t)$ **do** ▷ stopping

    $L_{i,t} = \min_{\lambda \in \Lambda_i(i^\star(\hat{\theta}_t))} \|\hat{\theta}_t - \lambda\|^2_{V_{N_t}}$

  end for

  if $\min_{i \neq i^\star(\hat{\theta}_t)} L_{i,t} > \beta_{t,\delta}$ **then** STOP

  Get w_t from a learner $\mathcal{L}$

  for $i \neq i^\star(\hat{\theta}_t)$ **do** ▷ sampling

    $\lambda_{i,t} = \arg\min_{\lambda \in \Lambda_i(i_t)} \|\hat{\theta}_t - \lambda\|^2_{V_{w_t}}$

  end for

  $\lambda_t = \arg\min_{i \neq i^\star(\hat{\theta}_t)} \|\hat{\theta}_t - \lambda_{i,t}\|^2_{V_{w_t}}$

  Update $\mathcal{L}$ with losses based on λ_t

  Pull arm $k_t = \arg\min_k N_t^k - \sum_{s=1}^t w_s$

end while

while not stopped **do**

  Set $\mathcal{P}_t^{\mathrm{stp}}(i^\star(\hat{\theta}_t)) = \mathcal{P}_{t-1}^{\mathrm{stp}}(i^\star(\hat{\theta}_t))$

  for $i \in \mathcal{P}_{t-1}^{\mathrm{stp}}(i^\star(\hat{\theta}_t))$ **do** ▷ stopping

    $L_{i,t} = \min_{\lambda \in \Lambda_i(i^\star(\hat{\theta}_t))} \|\hat{\theta}_t - \lambda\|^2_{V_{N_t}}$

    if $L_{i,t} > \beta_{t,\delta}$ **delete** i from $\mathcal{P}_t^{\mathrm{stp}}(i^\star(\hat{\theta}_t))$

  end for

  if $\mathcal{P}_t^{\mathrm{stp}}(i^\star(\hat{\theta}_t)) = \emptyset$ **then** STOP

  Get w_t from a learner $\mathcal{L}$

  for $i \in \mathcal{P}_t^{\mathrm{smp}}(i^\star(\hat{\theta}_t))$ **do** ▷ sampling

    $\lambda_{i,t} = \arg\min_{\lambda \in \Lambda_i(i_t)} \|\hat{\theta}_t - \lambda\|^2_{V_{w_t}}$

  end for

  $\lambda_t = \arg\min_{i \in \mathcal{P}_t^{\mathrm{smp}}(i^\star(\hat{\theta}_t))} \|\hat{\theta}_t - \lambda_{i,t}\|^2_{V_{w_t}}$

  Update $\mathcal{L}$ with losses based on λ_t

  Pull arm $k_t = \arg\min_k N_t^k - \sum_{s=1}^t w_s$

  Update $\mathcal{P}_{t+1}^{\mathrm{smp}}(i^\star(\hat{\theta}_t))$ (Algorithm 3)

end while

Algorithm 3 Update active pieces at sampling

Input: statistics at time t, answer i, active pieces $\mathcal{P}_{t-1}^{\mathrm{smp}}(i)$

Stores: current helper set $\tilde{\mathcal{P}}_{t-1}^{\mathrm{smp}}(i)$ (initialized as $\tilde{\mathcal{P}}_0^{\mathrm{smp}}(i) = \mathcal{P}(i)$), helper set at last reset $\tilde{\mathcal{P}}_{\mathrm{last}}^{\mathrm{smp}}(i)$

Output: updated active pieces $\mathcal{P}_t^{\mathrm{smp}}(i)$

if $t = \bar{t}_0^{2^j}$ for some integer $j \geq 0$ **then**

  Reset $\tilde{\mathcal{P}}_t^{\mathrm{smp}}(i) = \mathcal{P}(i)$ and store $\tilde{\mathcal{P}}_{\mathrm{last}}^{\mathrm{smp}}(i) = \tilde{\mathcal{P}}_{t-1}^{\mathrm{smp}}(i)$

else

  Set $\tilde{\mathcal{P}}_t^{\mathrm{smp}}(i) = \tilde{\mathcal{P}}_{t-1}^{\mathrm{smp}}(i)$

end if

for $p \in \tilde{\mathcal{P}}_{t-1}^{\mathrm{smp}}(i)$ **do**

  if $\inf_{\lambda \in \Lambda_p(i)} L_t(\hat{\theta}_t, \lambda) > \alpha_{t,\delta}$ **eliminate** p from $\tilde{\mathcal{P}}_t^{\mathrm{smp}}(i)$

end for

Return $\mathcal{P}_t^{\mathrm{smp}}(i) = \tilde{\mathcal{P}}_t^{\mathrm{smp}}(i) \cap \tilde{\mathcal{P}}_{\mathrm{last}}^{\mathrm{smp}}(i)$