ε-Good Action Identification in Fixed-Budget Monte Carlo Tree Search

Yinan Li
Department of Computer Science
University of Arizona
yinanli@arizona.edu

Tuan Nguyen
Department of Computer Science
University of Arizona
tnguyen9210@arizona.edu

Kwang-Sung Jun
GSAI/CSE
POSTECH
kwangsungjun@postech.ac.kr

Abstract

We study the max–min action identification problem in depth-2 trees, which is an important special case of Monte Carlo Tree Search (MCTS). Departing from prior work, we consider the fixed budget setting where the learner, for T time steps, sequentially and adaptively selects depth-2 nodes to observe stochastic rewards, and then recommends a subtree (i.e., depth-1 node) estimated to achieve the max-min value. Motivated by approximate planning, we focus on *ε-good subtree identification*, where any subtree whose min value is within ε of the optimal maximin value is acceptable. Our main contribution is an *ε-agnostic* algorithm—requiring no knowledge of ε—that nevertheless achieves instance-dependent ε-good error bounds. We show that, for every ε, the misidentification probability decays as $\exp\big(-\tilde{\Theta}(\frac{T}{H_2(\varepsilon)})\big)$, where $H_2(\varepsilon)$ captures both cross-subtree and within-subtree mean reward differences. In the special case where each subtree has a single child, the problem reduces to the standard pure exploration of multi-armed bandits where our bound recovers (up to accelerating factors) the best-known ε-good guarantee. On the lower-bound side, we present complementary positive and negative results. While there is a gap between the upper and lower bounds, we discuss the main technical challenges in obtaining a tighter lower bound, which tells us that the max-min tree problem is quite different from the standard K-armed bandits. To our knowledge, this is the first provable algorithmic guarantee for fixed-budget max-min action identification.

1 Introduction

Monte Carlo Tree Search (MCTS) is a widely used paradigm for planning in large sequential decision problems, especially in game trees where exhaustive search is computationally infeasible [2, 7, 20]. Starting from a root state, MCTS repeatedly performs simulated rollouts, selectively expands or samples parts of the tree, and uses the observed noisy payoffs to decide which root action to play. This viewpoint has been highly influential in game playing and planning such as modern large-scale systems combining tree search with learned value or policy functions [26]. From a statistical perspective, the key question is a pure-exploration one: given a limited simulation budget, how should samples be allocated across the tree so as to identify a good root action?

In a two-player zero-sum game tree, the value of a root action is determined by the opponent's subsequent responses. Thus the value propagated to the root is naturally described by alternating

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

max and min operations. In an arbitrary-depth tree, this max–min value is obtained recursively from the leaves to the root. The depth-2 case is the simplest nontrivial instance of this structure. Each root action i corresponds to a subtree, each leaf (i, j) corresponds to a possible opponent response, and the value of action i is $v_i = \min_{j\in[L]} \mu_{i,j}$. The goal is then to identify an action with largest max–min value, $\max_i v_i$. Although simple, this model already captures the central difficulty absent from standard best-arm identification: the learner must allocate samples both across root actions and within each action to determine its worst child.

This depth-2 abstraction is closely connected to prior theoretical work on MCTS and maximin action identification. Garivier et al. [12] introduced a strategic bandit model motivated by MCTS, in which the learner samples noisy outcomes of action pairs and aims to identify a maximin action in the fixed-confidence setting. Kaufmann and Koolen [16] studied best-action identification in game trees of arbitrary depth, using confidence intervals to summarize deeper levels of the tree and applying best-arm identification at the root. Huang et al. [13] developed a more general structured fixed-confidence framework motivated by minimax game search. More recently, the generic GLR/Tracking analysis of Kaufmann and Koolen [17, Section 5.2.2] implies asymptotic fixed-confidence optimality for depth-2 maxmin game trees under the usual regularity assumptions on the oracle weights. These works provide fixed-confidence guarantees, where the algorithm stops once it can return a correct action with probability at least $1 - \delta$. We discuss fixed-confidence literature in detail in Appendix A.

Departing from prior work, we study the fixed-budget setting. The learner is given a prescribed simulation budget T and must output a recommendation after exactly T samples. This criterion is natural in MCTS applications where a decision must be made after a fixed planning window. Unlike regret minimization, where performance is evaluated during the sampling process, fixed-budget pure exploration separates simulation from execution: the learner is judged only by the quality of the final recommendation [1, 3, 10]. Our objective is to upperbound the probability of recommending a non-optimal, or more generally a non-ε-good, root action after the fixed budget has been exhausted.

Max–min identification differs fundamentally from standard best-arm identification (BAI). In ordinary BAI, each candidate arm has a fixed mean that can be sampled directly. In our setting, the learner recommends a subtree i with value $v_i = \min_{j\in[L]} \mu_{i,j}$, while observations are obtained from individual leaves (i, j). This creates two additional difficulties.

First, directly applying an ordinary elimination algorithm to the KL leaves does not preserve the max–min objective: eliminating the minimizing leaf of a surviving subtree increases its represented min-value and may make a suboptimal subtree appear optimal. Our leaf/subtree elimination rule is designed to prevent this failure (Section 3.1).

Second, a naive bottom-up approach that first solves BAI within every subtree and then compares their minima can be wasteful. For example, applying Successive Accepts and Rejects [4] separately within every subtree may spend many samples distinguishing two nearly tied leaves of a suboptimal subtree, even when both leaves are already well below the optimal subtree. Such comparisons are irrelevant to identifying the optimal root action (see empirical evidence in Appendix G). The relevant difficulty is therefore jointly determined by cross-subtree and within-subtree gaps.

Moreover, in practical planning, identifying the *exact* optimal subtree is often unnecessary and even unrealistic: when multiple subtrees are nearly optimal, it suffices to output any subtree whose value is within a target tolerance ε of the optimum. This motivates the *ε-good subtree identification* objective studied in this work. Concretely, rather than requiring an exactly optimal subtree, we ask the algorithm to return an *ε-good* subtree, meaning a subtree whose min value (the smallest mean reward among its children) is at least $v^\star - \varepsilon$, where $v^\star$ is the optimal max-min value. This relaxation is algorithmically and statistically meaningful: it enlarges the set of acceptable outputs, often yields substantially smaller sample requirements, and better matches MCTS practice where planning is approximate. At the same time, the correct dependence on ε is subtle in max-min trees because the learner must resolve both *cross-subtree* comparisons (which subtree is better) and *within-subtree* structure (which leaf determines the min-value), and the relevant gaps depend on ε.

A key design goal of our algorithm is *ε-agnosticity*. In many deployments, ε is not known in advance, may be chosen downstream, or may vary across instances and operating conditions. We therefore develop an algorithm that *does not take ε as input* yet still enjoys guarantees that scale with the (unknown) target error: for every (meaningful) ε, the same algorithm achieves a fixed-budget error guarantee (or equivalently a sample-complexity guarantee) governed by an instance-dependent

complexity $H_2(\varepsilon)$ that captures the difficulty of identifying an ε-good subtree. In other words, our method automatically adapts to the target tolerance ε without prior knowledge of it.

To summarize this paper's contribution, we provide (i) an ε-agnostic sampling and recommendation procedure, (ii) an upper bound on the probability of recommending a non-ε-good subtree under a fixed sampling budget, and (iii) an instance-dependent complexity characterization $H_2(\varepsilon)$ that explicitly reflects both cross-subtree and within-subtree gaps. These guarantees extend classical fixed-budget best-arm identification analyses [e.g. 1] to max-min trees, and complement fixed-confidence maximin action identification work [e.g. 12, 13, 16] with the fixed-budget counterpart, and address approximate identification with automatic adaptation to ε. To our knowledge, this is the first provable algorithmic guarantee for fixed-budget MCTS. Our guarantees are not directly comparable to fixed-confidence guarantees, since the objectives differ. Nevertheless, the gap quantities appearing in our upper bound have a similar structural form as the inverse-gap terms in prior fixed-confidence analyses. See Appendix A for detailed discussions of the sample complexity.

On the lower-bound side, we prove information-theoretic lower bounds for both exact and ε-good maximin subtree identification. These bounds are governed by a critical-leaf complexity $H_{\mathrm{lb}}(\nu)$ and its ε-dependent extension $H_{\mathrm{lb}}(\nu, \varepsilon)$, respectively. They involve the leaves that determine competitor subtree minima and the leaves needed to certify the optimal subtree, thereby highlighting a structural source of hardness in max-min identification. We also defer to Appendix E a complementary negative result showing that a direct permutation-style lower-bound route, aligning with the idea of the lower bound in Audibert et al. [1] for unstructured BAI, cannot yield an H_2-type lower bound for max–min trees even in the smallest nontrivial $(K = 2, L = 2)$ case.

Our results provide nonasymptotic, instance-dependent error upper bounds and class-wise lower bounds. This differs from the notions of fixed-budget optimality studied by Degenne [8], Komiyama et al. [21], which investigate optimal asymptotic error rates and whether they can be attained across instances. These results show that sharp fixed-budget optimality is subtle even for unstructured BAI. To our knowledge, they do not directly extend to the structured max–min tree setting in a way that yields a matching characterization for our problem. We further show, through reductions from Positivity and ordinary BAI studied by Degenne [8], that for any fixed-budget algorithm, there exists a (K, L) max–min tree instance on which its sample-complexity upper bound differs from an instance-wise lower bound by a factor depending on K or L. We discuss these reductions and the challenges of obtaining tighter upper and lower bounds in Appendix F.

2 Preliminaries

Model and indexing. There are K subtrees (root actions) indexed by $i \in [K] := \{1, \dots, K\}$. Each subtree contains L leaves indexed by $j \in [L] := \{1, \dots, L\}$, and we denote a leaf by $(i, j) \in [K] \times [L]$. (Allowing subtree-dependent leaf counts L_i requires only notational changes; our upper and lower bounds extend verbatim.)

Each leaf (i, j) is associated with an unknown reward distribution $\nu_{i,j}$ on $\mathbb{R}$ with mean $\mu_{i,j} := \mathbb{E}_{X \sim \nu_{i,j}}[X]$. We assume $\nu_{i,j}$ is *sub-Gaussian with variance proxy* 1: for all $\lambda \in \mathbb{R}$,

$$\mathbb{E}_{X \sim \nu_{i,j}}\left[\exp(\lambda(X - \mu_{i,j}))\right] \le \exp\left(\frac{\lambda^2}{2}\right). \tag{1}$$

Max-min values and WLOG ordering. Define the min-value of subtree i by $v_i := \min_{j \in [L]} \mu_{i,j}$ and the optimal max–min value by $v^\star := \max_{i \in [K]} v_i$. Without loss of generality, re-index leaves within each subtree and re-index subtrees so that

$$\mu_{1,1} \ge \mu_{2,1} \ge \cdots \ge \mu_{K,1}, \qquad \mu_{i,1} \le \mu_{i,2} \le \cdots \le \mu_{i,L} \ \ \forall i \in [K]. \tag{2}$$

Then $v_i = \mu_{i,1}$ for all i, and subtree 1 is optimal with $v^\star = v_1 = \mu_{1,1}$. Throughout, we assume the optimal subtree is unique for simplicity.

Fixed-budget interaction. As illustrated in Algorithm 1, a (possibly randomized) learning algorithm interacts with the instance for a fixed budget of T rounds. At each round, it adaptively selects a leaf (i, j) and observes one sample from $\nu_{i,j}$. After T samples, it outputs a recommended subtree index $\widehat{i}_T \in [K]$.

Algorithm 1 Fixed-budget interaction

Require: Budget T
1: **for** $t = 1, 2, \ldots, T$ **do**
2: Select a leaf (i_t, j_t) adaptively (possibly at random)
3: Observe a sample $X_t \sim \nu_{i_t, j_t}$
4: **end for**
5: **Output**: A recommended subtree index $\hat{i}_T \in [K]$

ε-good subtree identification. For $\varepsilon \geq 0$, a subtree i is *ε-good* if its min-value is within ε of optimal:

$$v_i \geq v^\star - \varepsilon \qquad \Longleftrightarrow \qquad \mu_{i,1} \geq \mu_{1,1} - \varepsilon.$$

Let $\mathcal{G}_\varepsilon := \{i \in [K] : v_i \geq v^\star - \varepsilon\}$ denote the set of ε-good subtrees, and $g(\varepsilon) := |\mathcal{G}_\varepsilon|$ be the number of ε-good subtrees, and $\mathcal{B}_\varepsilon := [K] \setminus \mathcal{G}_\varepsilon$ the set of ε-bad subtrees. The learning objective is to output $\widehat{i}_T \in \mathcal{G}_\varepsilon$; the error event is $\{\widehat{i}_T \notin \mathcal{G}_\varepsilon\}$. When $G_\varepsilon = [K]$, the problem is trivial; Throughout the nontrivial analysis we consider ε such that $G_\varepsilon \neq [K]$.

Gap quantities. We define leaf-wise gaps used in our instance-dependent bounds. For the optimal subtree $i = 1$, for each $j \in [L]$ define

$$\Delta_{1,j} := \mu_{1,j} - \mu_{2,1}. \tag{3}$$

For any $i \neq 1$ and $j \in [L]$, define

$$\Delta_{i,j} := \max\Big\{\mu_{1,1} - \mu_{i,1},\ \mu_{i,j} - \mu_{i,1}\Big\}. \tag{4}$$

Under (2) and the unique optimal subtree assumption, all these gaps are positive. Intuitively, $\mu_{1,1} - \mu_{i,1}$ captures the cross-subtree gap in its min value, while $\mu_{i,j} - \mu_{i,1}$ captures the within-subtree difficulty of certifying the minimizing leaf. These quantities will parameterize our upper and lower bounds for ε-good identification.

Sorted gaps, critical gap, and complexity. Let $\{\Delta_{i,j} : (i,j) \in [K] \times [L]\}$ be the multiset of the KL gap values defined above. Let (ℓ) denote the index of the ℓ-th smallest element in this multiset,

$$\Delta_{(1)} \leq \Delta_{(2)} \leq \cdots \leq \Delta_{(KL)},$$

breaking ties arbitrarily. With gaps and sorted gaps, we define the complexity measure as follows:

Definition 1. Define the *critical gap* (the gap level that separates ε-good from non-ε-good subtrees)

$$\Delta^\star := \Delta_{g(\varepsilon)+1,\,1}. \tag{5}$$

Let m be the largest integer such that $\Delta_{(m+1)} = \Delta^\star$. Finally, define the (sorted-gap) complexities

$$H_2(\varepsilon) := \max_{r \geq m+1} r\,\Delta_{(r)}^{-2}, \qquad H_1(\varepsilon) := \sum_{i=1}^{KL} \big(\Delta_{(i)} \vee \varepsilon\big)^{-2}. \tag{6}$$

3 Successive Rejects for MCTS

3.1 From Successive Rejects to subtree identification.

Our algorithm design starts from the celebrated *Successive Rejects* (SR) algorithm for fixed-budget best-arm identification. SR splits the budget into phases; in each phase it samples the remaining arms uniformly and then removes one arm based on empirical evidence. A precise description appears in [1] (Figure 3 therein): at the end of phase k, SR removes the arm with the lowest empirical mean among the active (surviving) ones.

In our max-min subtree setting, the relevant sub-optimality of a leaf (i, j) is not captured by a single mean comparison: a leaf can matter because it separates min values of subtrees (via $\mu_{1,1} - \mu_{i,1}$) or because it separates the minimum within a subtree (via $\mu_{i,j} - \mu_{i,1}$). Accordingly, we run an SR-style phase-based procedure, but with (i) empirical gaps corresponding to our sub-optimality gaps $\Delta_{i,j}$ from Section 2, and (ii) a *subtree elimination* rule that removes whole subtrees when sub-optimality evidence is strong.

Algorithm 2 Successive Rejects for (K, L)-MCTS

Require: Total pull budget T

$\mathcal{A} \leftarrow [K] \times [L]$ // the set of active leaves

$\forall \mathcal{A}' \subseteq [K] \times [L], \mathcal{L}_i(\mathcal{A}') := \{j \in [L] : (i,j) \in \mathcal{A}'\}$ // extracts active leaves of subtree i

$\forall \mathcal{A}' \subseteq [K] \times [L], \mathcal{S}(\mathcal{A}') := \{i \in [K] : \mathcal{L}_i(\mathcal{A}') \neq \emptyset\}$ // extracts active subtrees

$k \leftarrow 1$ // phase index

Note: Throughout, ties for any choices are broken with an arbitrary yet fixed rule.

while $k \leq KL - 1$ **do**

  $n_k \leftarrow \left\lceil \dfrac{(T - KL)/\overline{\log}(KL)}{KL + 1 - k} \right\rceil$, where $\overline{\log}(KL) := \frac{1}{2} + \sum_{r=2}^{KL} \frac{1}{r}$

  $\forall (i,j) \in \mathcal{A}$, pull arm (i,j) until the pull count equals n_k and compute (over a total n_k pulls) the empirical mean $\hat{\mu}_{i,j}$

  $\forall i \in \mathcal{S}(\mathcal{A}),\ \hat{1}(i) \leftarrow \arg\min_{j \in \mathcal{L}_i(\mathcal{A})}\ \hat{\mu}_{i,j}$ // subtree i's empirical worst leaf (i.e., with the smallest value)

  $\hat{a} \leftarrow \arg\max_{i \in \mathcal{S}(\mathcal{A})} \hat{\mu}_{i,\hat{1}(i)}$ // the empirical maximin subtree

  $\hat{\mu}^* \leftarrow \hat{\mu}_{\hat{a},\hat{1}(\hat{a})}$ // the empirical maximin value

  $\forall (i,j) \in \mathcal{A}, \hat{\Delta}_{i,j} \leftarrow \begin{cases} \hat{\mu}_{i,j} - \max_{i' \neq \hat{a}} \hat{\mu}_{i',\hat{1}(i')} & \text{if } i = \hat{a} \\ \max\left(\hat{\mu}^* - \hat{\mu}_{i,\hat{1}(i)},\ \hat{\mu}_{i,j} - \hat{\mu}_{i,\hat{1}(i)}\right) & \text{otherwise} \end{cases}$ // empirical gaps

  $\hat{\Delta}_{\max} \leftarrow \max_{(i,j) \in \mathcal{A}} \hat{\Delta}_{i,j}$

  if $\exists$ a subtree $x \in \mathcal{S}(\mathcal{A}) \backslash \{\hat{a}\}$: $\forall j \in \mathcal{L}_x(\mathcal{A}),\ \hat{\Delta}_{x,j} = \hat{\Delta}_{\max}$ **then**

    Let x be the subtree satisfying the condition in the if statement.

    $\mathcal{E} \leftarrow \{(x,j) : j \in \mathcal{L}_x(\mathcal{A})\}$

  else

    Choose (x,y) among active leaves $\mathcal{A}$ s.t. $\hat{\Delta}_{x,y} = \hat{\Delta}_{\max}$

    $\mathcal{E} \leftarrow \{(x,y)\}$

  end if

  $\mathcal{A} \leftarrow \mathcal{A} \setminus \mathcal{E}$

  if $|\mathcal{S}(\mathcal{A})| = 1$ **then**

    Let $\hat{i}_T$ be the only element in $\mathcal{S}(\mathcal{A})$ // the only remaining subtree.

    break

  end if

  $k \leftarrow k + |\mathcal{E}|$

end while

Output: $\hat{i}_T$

Empirical minimizers, best subtree, and empirical gaps. Fix a phase k and let $\hat{\mu}_{i,j}$ denote the empirical mean of leaf (i,j) computed from all samples collected so far. For each active subtree i, define its empirical minimizer index

$$\hat{1}(i) \in \arg\min_{j \in \mathcal{L}_i(\mathcal{A})}\ \hat{\mu}_{i,j},$$

and the corresponding empirical min-value $\hat{v}_i(t) := \hat{\mu}_{i,\hat{1}(i)}$. Let the empirical best subtree be

$$\hat{a} \in \arg\max_{i \in \mathcal{S}(\mathcal{A})}\ \hat{v}_i, \qquad \hat{\mu}^\star := \hat{v}_{\hat{a}} = \hat{\mu}_{\hat{a},\hat{1}(\hat{a})}.$$

We now define empirical gaps $\hat{\Delta}_{i,j}$ by literally replacing population means in the true gaps (Section 2) with their empirical counterparts. For an active leaf (i,j):

$$\text{if } i = \hat{a}, \qquad \hat{\Delta}_{i,j}\ :=\ \hat{\mu}_{i,j}\ -\ \max_{s \neq i} \hat{\mu}_{s,\hat{1}(s)},$$

$$\text{if } i \neq \hat{a}, \qquad \hat{\Delta}_{i,j}\ :=\ \max\Big(\hat{\mu}^\star - \hat{\mu}_{i,\hat{1}(i)},\ \hat{\mu}_{i,j} - \hat{\mu}_{i,\hat{1}(i)}\Big).$$

These are empirical counterparts of $\Delta_{i,j}$ that either certify the optimal subtree, or simultaneously encode (a) separation from the current best subtree and (b) within-subtree separation from the current empirical minimum.

3.2 Tree-aware fixed-budget elimination: subtree vs. leaf

Our sampling rule belongs to the broad family of elimination/racing algorithms used in fixed-budget BAI and fixed-confidence maximin identification. The novelty is not the use of elimination [e.g. 1] per se. Rather, the issue in a max–min tree is that deleting a single leaf can change the surviving empirical minimum of a subtree and thereby make a suboptimal subtree appear artificially strong. Our subtree-removal condition is designed to avoid this failure mode in a fixed-budget SR-style schedule. The elimination step is where we depart from SR. Recall that $\mathcal{A}$ is the current set of active leaves. At the end of each phase, compute $\hat{\Delta}_{i,j}$ for all $(i,j) \in \mathcal{A}$. Let $\hat{\Delta}_{\max} := \max_{(i,j)\in\mathcal{A}} \hat{\Delta}_{i,j}$ be the empirical largest gap value among active leaves.

Single leaf elimination. In most phases, the algorithm behaves like Successive Rejects: it computes empirical gaps $\hat{\Delta}_{i,j}$ for all active leaves and removes one leaf with the largest empirical gap. Heuristically, a large gap indicates that the corresponding leaf is already well separated at the current resolution and is therefore unlikely to be the bottleneck for identifying an ε-good subtree; eliminating it concentrates future samples on the harder, smaller-gap comparisons.

Subtree elimination. There is, however, an important exception in max–min trees. In contrast to unstructured best-arm identification, eliminating a leaf while keeping its subtree active can change the subtree's surviving minimum. If the eliminated leaf is the true minimizer of a suboptimal subtree, then the minimum among the remaining leaves increases, so the algorithm would subsequently compare other subtrees against a different objective value than the true max–min value of that subtree. This creates a failure mode: a suboptimal subtree can appear artificially stronger after its minimizer is removed, which may corrupt later elimination decisions and ultimately the final recommendation.

Subtree-elimination ideas also appear in prior fixed-confidence MCTS and maximin-identification work. We do not claim that eliminating an entire subtree is new by itself. In Garivier et al. [12] and Teraoka et al. [27], a subtree can be eliminated once it contains a leaf that is confidently smaller than all leaves in another subtree. This condition is natural in the fixed-confidence setting and is closely aligned with the intuition behind α–β pruning [19]: once a subtree's worst-case value is certified to be below that of a competitor, the subtree can be safely discarded.

However, this type of elimination requires a high-confidence certificate of suboptimality, which in turn may require identifying the subtree's true minimizing leaf. In our fixed-budget setting, spending samples to certify the internal minimum of every potentially suboptimal subtree can be wasteful, and is not compatible with the goal of an ε-agnostic ε-good identification. Moreover, removing only a single leaf can be unsafe in a max–min tree: if the removed leaf is the true minimizer of a suboptimal subtree, then the minimum among the surviving leaves is artificially increased, so later phases compare against a distorted value for that subtree.

Our rule addresses this issue differently. We eliminate a subtree only when all of its active leaves attain the largest empirical gap $\widehat{\Delta}_{\max}$ (line 14 of Algorithm 2). In that case, rather than removing one leaf and risking distortion of the subtree's surviving min-value, we remove the entire subtree. Thus, our rule is not an α–β-style certificate of suboptimality; it is a tree-safe fixed-budget elimination mechanism designed to preserve the max–min structure during a Successive-Rejects-style procedure. In fact, our particular subtree elimination rule is essential for obtaining our guarantee, and we were not able to obtain the same guarantee with other straightforward rules.

In K-armed bandits, SR removes exactly one arm per phase, hence it runs a fixed number of phases. In contrast, our subtree elimination can remove *many* leaves in a single phase. Therefore, we implement the procedure with a `while` loop: the next phase is triggered only after updating the active set. The total number of phases is data-dependent, and it adapts to how many leaves were eliminated in the previous phase. Operationally, this places the method between SR and more aggressive halving-style schemes: eliminations can be as mild as SR (one leaf at a time) or much stronger when evidence supports discarding an entire subtree.

Budget reallocation. Our algorithm uses a conservative sampling schedule to keep the algorithm and analysis simple. When an entire subtree is eliminated, some phases may be skipped, leaving part of the budget unused. A natural variant is to redistribute this unused budget over the remaining phases according to the remaining budget and the number of active leaves. As long as every active leaf receives at least the cumulative number of samples prescribed by the original schedule, the same error guarantee can be maintained. This variant may improve empirical performance by making better use of the available budget.

3.3 Instance-dependent performance guarantee

An ε-agnostic guarantee for ε-good identification. A salient feature of Theorem 2 is that the algorithm is *ε-agnostic*: it does not take ε as an input and its sampling/elimination decisions are identical for all target accuracies. Nevertheless, for every meaningful $\varepsilon \geq 0$ (such that not all subtrees are ε-good), the same output $\widehat{i}_T$ enjoys an *ε-good* fixed-budget guarantee: the probability of returning a subtree whose min value is more than ε below that of the optimal subtree decays exponentially in the budget T, with an exponent governed by the ε-dependent complexity $H_2(\varepsilon)$. In other words, without knowing the target resolution in advance, the algorithm automatically adapts to the effective difficulty of identifying an ε-good subtree.

Theorem 2 (ε-agnostic ε-good error bound)**.** *Suppose $KL \geq 2$ and $T > KL$. Recall the definition of $H_2(\varepsilon)$ in Definition 1 and $\overline{\log}(KL)$ defined in line 7 of Algorithm 2. Then the failure probability of Successive Rejects for (K, L)-MCTS (Algorithm 2) in returning an ε-good subtree satisfies*

$$\mathbb{P}\Big(\widehat{i}_T \notin \mathcal{G}_\varepsilon\Big) \leq 2K^2L^2 \exp\left(-\frac{T - KL}{128\,\overline{\log}(KL)\,H_2(\varepsilon)}\right).$$

For comparison with prior fixed-confidence bounds, we relate the sorted-gap complexity $H_2(\varepsilon)$ to the sum-type quantity $H_1(\varepsilon)$. The following lemma shows that $H_2(\varepsilon)$ is always dominated by $H_1(\varepsilon)$; detailed comparisons with fixed-confidence complexities are given in Appendix A.

Lemma 3. *Recall the definition of $H_2(\varepsilon), H_1(\varepsilon)$ in Definition 1. We have,*

$$H_2(\varepsilon) \leq H_1(\varepsilon).$$

For $\varepsilon = 0$, the reverse inequality holds up to a logarithmic factor [1]. However, for $\varepsilon > 0$, such a reverse inequality does not hold in general: $H_1(\varepsilon)/H_2(\varepsilon)$ can be arbitrarily large, even for $K = 3$ and $L = 1$. See Appendix C for the proof of Lemma 3 and a counterexample showing that the reverse inequality does not hold in general for $\varepsilon > 0$.

Special case: K-armed bandits ($L = 1$). When each subtree has a single leaf ($L = 1$), the model reduces to the classical K-armed bandit identification problem: there are arms indexed by $i \in [K]$, each arm i has an unknown 1-sub-Gaussian reward distribution with mean μ_i, and the learner allocates a fixed budget T of samples before outputting a recommendation $\widehat{i}_T$. Without loss of generality assume $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_K$, so that arm 1 is optimal and $\mu^\star = \mu_1$. For $\varepsilon \geq 0$, an arm i is called ε-good if $\mu_i \geq \mu^\star - \varepsilon$, and let $\mathcal{G}_\varepsilon := \{i \in [K] : \mu_i \geq \mu^\star - \varepsilon\}$.

Corollary 4 (K-armed bandits: fixed-budget ε-good identification)**.** *Specialize Theorem 2 to $L = 1$. Then the failure probability of Successive Rejects in outputting an ε-good arm satisfies*

$$\mathbb{P}(\widehat{i}_T \notin \mathcal{G}_\varepsilon) \;\leq\; 2K^2 \exp\left(-\frac{T - K}{128\,\overline{\log}(K)\,H_2(\varepsilon)}\right),$$

where $H_2(\varepsilon)$ is the ε-dependent sorted-gap complexity from Definition 1 (with the gaps reducing to the standard arm-gap quantities when $L = 1$).

Remark 5 (novelty already for K-armed bandits)**.** Even in the classical K-armed bandit setting ($L = 1$), Corollary 4 appears to be new for the *Successive Rejects* (SR) analysis: it provides an ε-agnostic ε-good guarantee, with an explicit ε-dependent complexity $H_2(\varepsilon)$ governing the fixed-budget error exponent. While SR is well studied for exact identification ($\varepsilon = 0$), extending SR-style analyses to approximate identification is nontrivial because the set of acceptable outputs depends on ε and the proof must control the elimination dynamics uniformly over all meaningful accuracy levels.

Related ε-agnostic guarantees are known for Sequential Halving (SH) in ordinary BAI ($L = 1$) [28]. Our $H_2(\varepsilon)$ guarantee matches their result up to accelerating factors. However, directly extending SH to max–min trees by eliminating half of the active leaves according to our empirical structured gaps is not tree-safe: it may eliminate the minimizing leaf of a suboptimal subtree while other leaves of that subtree remain active. A tree-aware SH variant would need to combine halving with our subtree-elimination rule. In this case, the number of surviving leaves becomes data-dependent, so the usual SH budget schedule and analysis do not transfer directly. We discuss this further in Appendix A.

3.4 Proof roadmap

The proof of Theorem 2 follows an inductive "invariant + concentration" strategy. The central invariant is the notion of ε-*soundness* (Definition 6), which formalizes what it means for the active set at the beginning of a phase to remain faithful to the true max–min structure. Recall that $\mathcal{A}$ is the current active leaf set, and for each subtree $i \in [K]$, write $\mathcal{A}_i = \mathcal{A} \cap (\{i\} \times [L])$ as its active leaves. We call subtree i *nonempty* if $\mathcal{A}_i \neq \emptyset$ and *empty* otherwise. We call leaf (i, j) *surviving* if $(i, j) \in \mathcal{A}$.

Definition 6. For any phase k, we say "phase k is ε-sound" if at the beginning of phase k,

If $k \leq KL - m$:

(1) The optimal subtree is nonempty: $\mathcal{A}_1 \neq \emptyset$.

(2) $\forall x \neq 1$ and $\varepsilon/2$-good, $(x, 1)$ is surviving: $\forall x \in \mathcal{G}_{\varepsilon/2} \setminus \{1\},\ (x, 1) \in \mathcal{A}$.

(3) $\forall x$ such that x is $\varepsilon/2$-bad, either $(x, 1)$ is surviving, or x is empty.

If $k > KL - m$:

(4) Either all $\varepsilon/2$-good subtrees are nonempty, or all ε-bad subtrees are empty.

(5) $\forall x$ such that x is ε-bad, either $(x, 1)$ is surviving, or x is empty.

For the ease of exposition, suppose the algorithm terminates after exactly k' phases, we still consider the soundness of $k' + 1$, which implies the termination condition.

Roughly, soundness requires two things: (i) the optimal subtree (and, more generally, near-optimal subtrees) are not prematurely eliminated, and (ii) any not-near-optimal subtree either still contains its true minimizing leaf or has already been completely removed. This invariant is essential because eliminating the true minimizer of a sub-optimal subtree would "inflate" its surviving minimum and thereby distort subsequent max–min comparisons.

Throughout the analysis, we assume without loss of generality that the critical gap $\Delta^\star$ is well-defined, i.e., not all subtrees are ε-good. Indeed, if every subtree is ε-good, then every output is acceptable and the misidentification probability is identically zero, so the theorem holds trivially. Under this nontrivial regime, $\Delta^\star$ and the associated index m (Definition 1) meaningfully separate "acceptable" subtrees from those that must be rejected.

Key idea: protect $\varepsilon/2$-good subtrees from ε-bad ones. At a high level, the algorithm is designed to ensure that any truly ε-bad subtree is eliminated *before* any $\varepsilon/2$-good subtree can be discarded. This is the role of Definition 6: conditions 3 and 5 rule out the possibility that a non-near-optimal subtree appears artificially strong. In the early regime ($k \leq KL - m$), conditions 1–3 guarantee that the optimal subtree remains active and that every $\varepsilon/2$-good subtree stays safely anchored by its true minimizer $(x, 1)$. See Lemma 13 to Lemma 19, in particular Lemma 16 to Lemma 19 in the Appendix for formal statements. Specifically, we include a warmup section in appendix B.1 that analyzes phase 1, to assist understanding of the proof ideas. In the later regime ($k > KL - m$), conditions 4–5 capture the endgame: either all $\varepsilon/2$-good subtrees persist, or all ε-bad subtrees have been eliminated, which in particular implies the termination condition. See Lemma 21 to Lemma 22 in the Appendix for formal statements.

The formal induction argument (Lemma 23 in the Appendix) proceeds by showing that on a suitable high-probability deviation event (uniform concentration of empirical means), every phase is ε-sound. The induction step analyzes the elimination rule at the end of a phase and proves that, assuming phase k is ε-sound (at the beginning), the algorithm will not break any condition in ε-soundness with the elimination happening in phase k. This maintains soundness across phases and yields the desired fixed-budget error bound after union-bounding over phases.

4 Lower bounds for maximin subtree identification

This section investigates lower bounds for fixed-budget maximin subtree identification. We first study exact identification ($\varepsilon = 0$), then extend the lower bound to ε-good identification. We introduce a complexity term tailored to the maximin structure, which will govern the instance-dependent (over instances with bounded complexity) fixed-budget lower bound.

Lower bound complexity. Let ν denote an instance, and recall that the (subtree-level) gaps $\Delta_{i,1}$ for $i \neq 1$ and the (within-optimal-subtree) gaps $\Delta_{1,j}$ for $j \neq 1$ are defined in Section 2. We define the

max-min lower bound complexity

$$H_{\mathrm{lb}}(\nu) \;:=\; \sum_{i\neq 1} \frac{1}{\Delta_{i,1}^2} \;+\; \sum_{j\neq 1} \frac{1}{\Delta_{1,j}^2}. \tag{7}$$

The first term $\sum_{i\neq 1} \Delta_{i,1}^{-2}$ captures the cost (in terms of number of samples needed) of separating the competitor's minimum leaf mean $\{\mu_{i,1} : i \neq 1\}$ (even if the minimizing leaf is known for each non-optimal subtree) from the optimal minimum leaf mean $\mu_{1,1}$. The second term $\sum_{j\neq 1} \Delta_{1,j}^{-2}$ reflects the cost of verifying the subtree 1 is indeed the optimal subtree: even if the min value of the strongest competitor subtree ($\mu_{2,1}$) is known, the learner must gather enough evidence to at least validate that all non-minimizer leaves inside the optimal subtree 1, $\{\mu_{1,j} : j \neq 1\}$, are above $\mu_{2,1}$.

Recall the definition of $H_1(\varepsilon)$ in Definition 1. Note that with $\varepsilon = 0$, $H_2(0)$ and $H_1(0)$ are well known to be equal up to logarithmic factors, i.e., $H_2(0) \leq H_1(0) \leq \log(2KL)\, H_2(0)$ [e.g., 1]. For any max-min tree instance ν, to introduce the explicit dependence on ν, we write

$$H(\nu) \;\equiv\; H_1(0) \;=\; \sum_{(i,j)\in[K]\times[L]} \frac{1}{\Delta_{i,j}^2}.$$

By construction, the summands defining $H_{\mathrm{lb}}(\nu)$ form a subset of those in $H(\nu)$, hence $H_{\mathrm{lb}}(\nu) \leq H(\nu)$ for every instance ν. For $\mathcal{H} > 0$, define the class of instances with bounded complexity:

$$\mathcal{P}_{\mathcal{H}} \;:=\; \big\{\nu :\; H(\nu) \leq \mathcal{H}\big\}.$$

Theorem 7 (Lower bound for maximin trees)**.** *Let ν be a maximin-tree instance with a unique optimal subtree, and assume its reward distributions on all leaves are Gaussian with variance 1. Consider the instance class $\mathcal{P}_{4H(\nu)}$. For any fixed-budget algorithm π that collects at most T samples in total, there exists an instance $\nu' \in \mathcal{P}_{4H(\nu)}$ such that*

$$\mathbb{P}_{\nu'}(\mathrm{error}) \geq \frac{1}{4} \exp\!\left(-\frac{8T}{H_{\mathrm{lb}}(\nu)}\right),$$

where $\mathbb{P}_{\nu'}(\mathrm{error})$ denotes the probability that π outputs an incorrect optimal subtree on ν'.

The proof of Theorem 7 is given in Appendix D.2. Theorem 7 is a local worst-case statement around a reference instance ν: even when restricting attention to instances whose leafwise complexity $H(\cdot)$ is within a constant factor of $H(\nu)$, there exists a hard instance ν' on which the error cannot decay faster than $\exp(-\Theta(T/H_{\mathrm{lb}}(\nu)))$. The appearance of $H_{\mathrm{lb}}(\nu)$ reflects that, in maximin trees, the difficulty is driven by at least a subset of "critical" leaves (those determining subtree minima and certifying the optimal subtree). This "non-uniform importance" phenomenon is also visible in the fixed-confidence literature, e.g., in the optimal allocation behavior discussed by Garivier et al. [12] in their Section 6, certain arms receive asymptotically vanishing sampling effort depending on the instance, indicating that they become effectively irrelevant as the confidence level increases. We discuss this phenomenon further in Appendix F.2.

Dependence on (K, L). For exact identification, $\Delta_{i,j} \geq \Delta_{i,1}$ for every $i \geq 2$, and $\Delta_{1,1} = \Delta_{2,1}$. Consequently, $H_2(0) \leq H_1(0) \leq (L+1)H_{\mathrm{lb}}(\nu)$. Thus the complexity in our upper bound, $\overline{\log}(KL)H_2(0)$, is at most an $O(L\overline{\log}(KL))$ factor larger than $H_{\mathrm{lb}}(\nu)$. To investigate whether a gap depending on (K, L) is unavoidable, we consider reductions from Positivity and ordinary best-arm identification. These reductions show that, for any fixed-budget algorithm, there exists a max–min instance with an $\Omega(\max\{L, \log K\})$ gap in the asymptotic difficulty ratio [8]. Thus, some dependence on (K, L) is unavoidable, although neither our upper nor lower complexity bound is known to be tight. We provide details in Appendix F.1.

Connection to classical BAI lower bounds ($L = 1$). When $L = 1$, the maximin tree reduces to the standard K-armed best-arm identification (BAI) problem. In this degenerate case, Theorem 7 recovers the familiar fixed-budget H_1 lower bound established in Kaufmann et al. [18, Theorem 16] and Carpentier and Locatelli [5, Theorem 1, first bound]. Indeed, $H(\nu) = H_1(0)$ becomes the usual leafwise complexity $\sum_{i\neq 1} \Delta_i^{-2}$, while $H_{\mathrm{lb}}(\nu)$ collapses to the same quantity (since the within-optimal-subtree term vanishes), and the statement reduces to an $\exp\big(-\Theta(T/H(\nu))\big)$ lower bound on the misidentification probability for some instance of comparable complexity.

Before proving Theorem 7, we provide a warm-up result for the K-armed case, stated as Theorem 25. There we present a clean proof under Gaussian arms with unit variance, using a flipping-style construction as in Carpentier and Locatelli [5] (see also Kaufmann et al. [18]). Our proof does not rely on the classical fixed-budget change-of-measure route based on the empirical KL divergence [1, 5]. Instead, it follows a proof recipe common in regret lower bounds and also in Kaufmann et al. [18]: construct a collection of nearby alternatives, decompose the KL divergence of the induced transcript distributions via the expected number of pulls, and convert KL divergence into an unavoidable testing error via standard inequalities (e.g., Bretagnolle–Huber / Le Cam). This perspective aligns with the information-theoretic tools developed for regret minimization and is in the same spirit as the techniques surveyed in standard bandit references [e.g. 24].

Proof strategy for maximin trees. The proof of Theorem 7 extends the warm-up recipe to the tree-structured setting. At a high level, we again build a finite family of alternative instances by perturbing as few leaves as possible at a time. We then argue, via a contradiction-style argument over this family, that any fixed-budget algorithm must either allocate substantial samples to each leaf (or a group of leaves), or incur a non-negligible probability of misidentification on at least one alternative. The resulting lower bound is governed by $H_{\mathrm{lb}}(\nu)$, which isolates the structural source of hardness in maximin identification: separating subtree minima across competitors and certifying the minimum within the optimal subtree. The full proof is deferred to Appendix D.2.

Extension to ε-good identification. For $\varepsilon > 0$, define

$$H_{\mathrm{lb}}(\nu, \varepsilon) := \sum_{i=2}^{K} \frac{1}{(\Delta_{i,1} \vee \varepsilon)^2} + \sum_{j=2}^{L} \frac{1}{(\Delta_{1,j} \vee \varepsilon)^2}.$$

Theorem 8 (Lower bound for ε-good identification)**.** *Under the same assumptions as Theorem 7, for any $\varepsilon > 0$ and any fixed-budget algorithm, there exists an instance ν' with $H_1(\varepsilon; \nu') \le 4H_1(\varepsilon; \nu)$ such that*

$$\mathbb{P}_{\nu'}\left(\widehat{i}_T \notin \mathcal{G}_\varepsilon(\nu')\right) \ge \frac{1}{4} \exp\left(-\frac{18T}{H_{\mathrm{lb}}(\nu, \varepsilon)}\right).$$

Unlike exact identification, changing the optimal subtree does not necessarily make the original optimum ε-bad. We address this by constructing a base instance whose only ε-good subtree is 1, together with alternatives on which subtree 1 is ε-bad. The proof is given in Appendix D.3.

A limitation of permutation-style lower bounds. A complementary negative result is given in Appendix E, where we show that a permutation-style lower-bound route, in the spirit of Audibert et al. [1] and based on the idea that the knowledge (just except for the indices) of the instance is available to the algorithm, cannot by itself establish an H_2-type lower bound for max–min trees. This does not rule out lower bounds using other alternative families or different quantifiers. Together with Theorem 7, this suggests that fixed-budget max–min lower bounds are qualitatively subtler than their unstructured BAI counterparts.

5 Conclusions

We studied the fixed-budget depth-2 max–min action identification motivated by Monte Carlo Tree Search (MCTS), focusing on ε-good subtree identification, where any subtree whose min value is within ε of optimal is acceptable. As our main contribution, we present a Successive-Rejects-style *ε-agnostic* method that requires no knowledge of ε yet achieves instance-dependent guarantees with explicit ε-dependence governed by $H_2(\varepsilon)$. To the best of our knowledge, this work gives the first provable fixed-budget guarantees for an MCTS max–min action identification problem. On the lower-bound side, we establish lower bounds for both exact and ε-good identification, governed by a critical-leaf complexity that captures the difficulty of distinguishing subtree minima and certifying the optimal subtree.

Important open problems include closing the upper–lower gap, extending the analysis to deeper trees, and developing fixed-budget guarantees for more complex and general tree-search problems, such as stochastic game trees with chance nodes [22], non-max–min backup rules [7, 23], and general structured pure-exploration objectives [9, 13].

Acknowledgments

Yinan Li, Tuan Ngo Nguyen, Kwang-Sung Jun were supported in part by the National Science Foundation under grant CCF-2327013 and Meta Platforms, Inc. This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) in 2026 (Grant No. RS-2026-25612350, Project Title: Development of an End-to-End Design and Validation Framework for Mathematically Principled Next-Generation AI Architectures and Learning Algorithms, contribution 45%). This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (RS-2019-II191906, Artificial Intelligence Graduate School Program(POSTECH), contribution 5%; RS-2026-25616313, AI Star Fellowship(POSTECH, Interdisciplinary Program), contribution 50%).

References

[1] J.-Y. Audibert, S. Bubeck, and R. Munos. Best Arm Identification in Multi-Armed Bandits. In *Proceedings of the Conference on Learning Theory (COLT)*, 2010.

[2] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

[3] S. Bubeck, R. Munos, and G. Stoltz. Pure Exploration in Multi-armed Bandits Problems. In *Proceedings of the International Conference on Algorithmic Learning Theory (ALT)*, pages 23–37, 2009.

[4] S. Bubeck, T. Wang, and N. Viswanathan. Multiple Identifications in Multi-Armed Bandits. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 258–265, 2013.

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

[6] T. Cazenave. Sequential halving applied to trees. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(1):102–105, 2014.

[7] R. Coulom. Efficient selectivity and backup operators in monte-carlo tree search. In *International conference on computers and games*, pages 72–83. Springer, 2006.

[8] R. Degenne. On the existence of a complexity in fixed budget bandit identification. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 1131–1154. PMLR, 2023.

[9] R. Degenne and W. M. Koolen. Pure exploration with multiple correct answers. *Advances in Neural Information Processing Systems*, 32, 2019.

[10] E. Even-Dar, S. Mannor, and Y. Mansour. Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems. *Journal of Machine Learning Research*, 7:1079–1105, 2006.

[11] V. Gabillon, M. Ghavamzadeh, A. Lazaric, and S. Bubeck. Multi-bandit best arm identification. *Advances in Neural Information Processing Systems*, 24, 2011.

[12] A. Garivier, E. Kaufmann, and W. M. Koolen. Maximin action identification: A new bandit framework for games. In *Conference on Learning Theory*, pages 1028–1050. PMLR, 2016.

[13] R. Huang, M. M. Ajallooeian, C. Szepesvári, and M. Müller. Structured best arm identification with fixed confidence. In *International Conference on Algorithmic Learning Theory*, pages 593–616. PMLR, 2017.

[14] Z. Karnin, T. Koren, and O. Somekh. Almost optimal exploration in multi-armed bandits. In *International conference on machine learning*, pages 1238–1246. PMLR, 2013.

[15] J. Katz-Samuels and K. Jamieson. The true sample complexity of identifying good arms. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1781–1791, 2020.

[16] E. Kaufmann and W. M. Koolen. Monte-carlo tree search by best arm identification. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4897–4906, 2017.

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

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

[19] D. E. Knuth and R. W. Moore. An analysis of alpha-beta pruning. *Artificial intelligence*, 6(4): 293–326, 1975.

[20] L. Kocsis and C. Szepesvári. Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer, 2006.

[21] J. Komiyama, T. Tsuchiya, and J. Honda. Minimax optimal algorithms for fixed-budget best arm identification. *Advances in Neural Information Processing Systems*, 35:10393–10404, 2022.

[22] M. Lanctot, A. Saffidine, J. Veness, C. Archibald, and M. H. Winands. Monte carlo*-minimax search. *arXiv preprint arXiv:1304.6057*, 2013.

[23] M. Lanctot, M. H. Winands, T. Pepels, and N. R. Sturtevant. Monte carlo tree search with heuristic evaluations using implicit minimax backups. In *2014 IEEE Conference on Computational Intelligence and Games*, pages 1–8. IEEE, 2014.

[24] T. Lattimore and C. Szepesv\'ari. Bandit Algorithms. 2018. URL `http://downloads.tor-lattimore.com/book.pdf`.

[25] R. Munos. From bandits to monte-carlo tree search: The optimistic principle applied to optimization and planning. *Foundations and Trends® in Finance*, 7(1):1–129, 2014.

[26] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[27] K. Teraoka, K. Hatano, and E. Takimoto. Efficient sampling method for monte carlo tree search problem. *IEICE TRANSACTIONS on Information and Systems*, 97(3):392–398, 2014.

[28] Y. Zhao, C. Stephens, C. Szepesvári, and K.-S. Jun. Revisiting simple regret: Fast rates for returning a good arm. In *International Conference on Machine Learning*, pages 42110–42158. PMLR, 2023.

Appendix

Table of Contents

A Related Work **13**

B Proof of Theorem 2 **17**
B.1 Warmup: Analysis of phase 1 17
B.2 Analysis of a generic phase k in early regime ($k \leq KL - m$) 19
B.3 Analysis of a generic phase k in late regime ($k > KL - m$) 24
B.4 Proof of Theorem 2 25

C Proof of Lemma 3 **28**

D Proof of Theorem 7 **29**
D.1 Warmup: A fixed-budget lower bound for BAI with Gaussian arms 29
D.2 Proof of Theorem 7 30
D.3 Proof of Theorem 8 33

E A negative result for permutation-style lower bounds **36**

F Further discussion of the upper-lower-bound gap **39**
F.1 Dependence on (K, L) 39
F.2 Critical leaves and fixed-budget optimality 40

G Experiments **42**

A Related Work

MCTS, bandit planning, and fixed-budget tree search. Monte Carlo Tree Search (MCTS) was developed as a way to combine selective tree search with Monte Carlo evaluation in large game trees [2, 7]. A central algorithmic development is UCT, which applies upper-confidence bandit ideas to Monte Carlo planning [20]. This connection between bandit allocation and planning has also been studied more broadly through the optimistic principle for large-scale optimization and planning [25].

Our work follows the pure-exploration interpretation of MCTS: during planning, the learner spends a finite simulation budget and is evaluated only through the quality of the final root action. This is distinct from regret minimization during sampling. Several practical MCTS algorithms exploit this fixed-budget viewpoint. For example, Cazenave [6] proposed SHOT, which applies Sequential Halving to trees, and Teraoka et al. [27] studied subtree deletion rules for Monte Carlo tree search. These works are algorithmically motivating, but they do not provide the same type of instance-dependent fixed-budget error guarantee for noisy depth-2 max–min identification that we study here.

Fixed-budget BAI, elimination algorithms, and bottom-up baselines. Our algorithm is inspired by the fixed-budget best-arm identification literature. In ordinary BAI, the learner receives a budget T and must recommend an arm after the budget is exhausted. This setting is closely related to simple-regret minimization in pure exploration [3]. The Successive Rejects algorithm of Audibert et al. [1] is a classical fixed-budget elimination algorithm: it samples surviving arms in phases and removes one empirically suboptimal arm at the end of each phase. Karnin et al. [14] later proposed Sequential Halving, another elimination-based fixed-budget method with strong guarantees.

A natural bottom-up baseline for our problem would be to first solve the within-subtree identification problem for every subtree, and then compare the estimated subtree minima. For example, one could

apply a best-arm identification algorithm to the negated rewards within each subtree in order to identify its minimizing leaf. This idea is closely related to the multi-bandit BAI setting of Gabillon et al. [11], where the goal is to identify the best arm in each of several bandit problems, and to the Successive Accepts and Rejects framework of Bubeck et al. [4], which also applies to multi-bandit best-arm identification. Such a bottom-up approach, however, is potentially wasteful for max–min action identification. The final decision only requires identifying a subtree with large minimum value; it does not require solving all within-subtree minimization problems to high accuracy. In particular, for a subtree that is clearly suboptimal, many non-minimizing leaves may be irrelevant to the final recommendation.

Our setting reduces to standard fixed-budget BAI when $L = 1$, but the max–min case $L > 1$ introduces an additional structural difficulty. A leaf may matter because it separates the min-values of two subtrees, or because it identifies the minimizing leaf within a subtree. Therefore, directly applying ordinary SR to all KL leaves would ignore the max–min semantics, while a fully bottom-up multi-bandit approach may oversolve irrelevant within-subtree problems. Our algorithm keeps the phase-based spirit of SR, but modifies the empirical gaps and elimination rule so that the active set remains faithful to the max–min structure while avoiding the need to certify every subtree minimum.

Approximate identification and multiple correct answers. The ε-good objective is also related to pure-exploration problems with multiple acceptable answers. In ordinary bandits, Katz-Samuels and Jamieson [15] study the problem of identifying an arm whose mean is within ε of the best mean, and characterize how the number of good arms can change the sample complexity. Zhao et al. [28] give an ε-good analysis of Sequential Halving that holds for every ε although ε is not an input to the algorithm. In the fixed-confidence setting, Degenne and Koolen [9] study pure exploration with multiple correct answers and show that extending Track-and-Stop-style methods to this setting requires care because the answer map can be discontinuous.

Our problem has a related but different multiple-answer structure. The acceptable set is

$$G_\varepsilon = \{i : v_i \geq v^\star - \varepsilon\},$$

where each value v_i is itself a minimum over leaves. Thus, ε-good max–min identification combines approximate root-action identification with within-subtree minimization. Our contribution is an ε-agnostic fixed-budget guarantee for this structured objective.

Relation to Sequential Halving. A natural attempt to extend Sequential Halving (SH) to max–min trees would be to rank active leaves according to their empirical structured gaps $\hat{\Delta}_{i,j}$ and eliminate half of them in each phase. However, this may remove the minimizing leaf of a suboptimal subtree while other leaves of that subtree remain active, thereby inflating its represented min-value. Thus, the ε-agnostic SH analysis of Zhao et al. [28] does not directly extend to max–min identification.

A plausible SH-MCTS variant would combine halving with our tree-safe subtree-elimination condition: a surviving suboptimal subtree should retain its minimizing leaf or be removed entirely. However, the number of leaves eliminated in each phase would then become instance- and data-dependent. Consequently, the number of surviving leaves after each phase is no longer known in advance, so the standard SH budget schedule, which relies on eliminating exactly half of the candidates, does not apply directly.

Proving the $g(\varepsilon/2)$ acceleration of Zhao et al. [28] creates an additional difficulty. Their argument tracks the increasing fraction of good arms among surviving candidates. In our setting, elimination acts on leaves while acceptable answers are subtrees. Thus, their counting argument does not transfer directly. Establishing a similar acceleration would require a new subtree-level survival and counting argument.

Developing and analyzing a tree-aware SH variant is an interesting direction for future work.

Asymptotically optimal fixed-confidence methods. The generic GLR/Tracking framework of Kaufmann and Koolen [17] also applies to best-action identification in maxmin game trees. In their Section 5.2.2, the goal is to identify the root action with largest value in a maxmin game tree by querying noisy samples from the leaves. For depth-two trees, they note that the problem has rank $L + 1$, where L is the maximum number of second-player actions, and that a GLR stopping rule with the corresponding rank-based threshold is asymptotically optimal when combined with Tracking, under the required regularity assumptions on the oracle weights.

Table 1: Comparison with closely related fixed-confidence work.

Work	Setting	Tree/model	Algorithmic style
Garivier et al. [12]	Fixed confidence	Depth-2 maximin trees	M-LUCB; Maximin-Racing
Kaufmann and Koolen [16]	Fixed confidence	Arbitrary-depth game trees	BAI-MCTS with propagated confidence intervals
Huang et al. [13]	Fixed confidence	General structured BAI / minimax search	LUCB-micro
Kaufmann and Koolen [17, Section 5.2.2]	Fixed confidence	Depth-2 maxmin game trees as a generic identification problem	GLR stopping + Tracking
This paper	**Fixed budget**	**Depth-2 max–min trees**	**SR-style tree-aware elimination**

Their optimal allocation is characterized through the variational quantity

$$T^\star(\mu)^{-1} = \sup_{w\in\Sigma} \inf_{\lambda\in \mathsf{Alt}(\mu)} \sum_a w_a d(\mu_a, \lambda_a),$$

and, for maxmin game trees, this characterization does not generally yield a simple closed-form sampling rule. Even in the depth-two case, the oracle weights are described as computable through disciplined convex optimization tools and as only numerically computable. In contrast, our result studies the fixed-budget criterion and gives an explicit finite-budget error-probability bound for ε-good identification using an ε-agnostic Successive-Rejects-style procedure.

Complexity comparison with fixed-confidence guarantees. Fixed-confidence and fixed-budget criteria optimize different quantities, so the following comparison should be interpreted at the level of instance-dependent gap structure rather than as a formal dominance statement. Fixed-confidence bounds control quantities such as $\mathbb{E}_\mu[\tau_\delta]$ or high-probability stopping times, typically in the regime $\delta \to 0$, whereas our result controls

$$\mathbb{P}_\mu\Big(\widehat{i}_T \notin G_\varepsilon\Big)$$

for a deterministic budget T. Thus, translating a fixed-budget error exponent of the form

$$\mathbb{P}_\mu\Big(\widehat{i}_T \notin G_\varepsilon\Big) \lesssim \exp\Big(-\frac{T}{H}\Big)$$

into the heuristic relation $\delta \approx \exp(-T/H)$ is useful only for comparing the leading inverse-gap terms. It should not be interpreted as saying that one setting strictly dominates the other. In what follows, we compare the complexity measures only to clarify how the same max–min gap structure appears across fixed-budget and fixed-confidence analyses.

To the best of our knowledge, our work provides the first *fixed-budget* guarantees for maximin action identification. Prior theoretical results for MCTS identification are primarily in the fixed-confidence setting, where the goal is to return the optimal subtree with probability at least $1-\delta$ and to minimize the (random) stopping time. The closest fixed-confidence references are Garivier et al. [12], Huang et al. [13], Kaufmann and Koolen [16].

Garivier et al. [12] study depth-2 maximin trees under $\varepsilon = 0$ (exact identification) in the fixed-confidence setting. They propose two algorithms whose sample complexities are characterized by instance-dependent terms $H_1^\star(\mu)$ and $H_2^\star(\mu)$. In our notation (with indices defined in Equation 2), their complexities can be written as

$$H_1^\star(\mu) = \sum_{j\in[L]} \frac{1}{\Big(\mu_{1,j} - \frac{\mu_{1,1}+\mu_{2,1}}{2}\Big)^2} + \sum_{(i,j):i\neq 1} \frac{1}{\Big(\frac{\mu_{1,1}+\mu_{2,1}}{2} - \mu_{i,1}\Big)^2 \vee \Big(\mu_{i,j} - \mu_{i,1}\Big)^2},$$

and

$$H_2^\star(\mu) := \sum_{j\in[L]} \frac{1}{\big(\mu_{1,j} - \mu_{1,1}\big)^2 \vee \big(\mu_{1,1} - \mu_{2,1}\big)^2} + \sum_{(i,j):i\neq 1} \frac{1}{\big(\mu_{1,1} - \mu_{i,1}\big)^2 \vee \big(\mu_{i,j} - \mu_{i,1}\big)^2}.$$

Our fixed-budget sample complexity (up to logarithmic factors) is

$$H_2(\varepsilon) := \max_{r\geq m+1} r\,\Delta_{(r)}^{-2}, \qquad \text{with} \qquad H_2(\varepsilon) \leq H_1(\varepsilon)$$

by Lemma 3. At $\varepsilon = 0$,

$$H_1(0) = \sum_j \frac{1}{\big(\mu_{1,j} - \mu_{2,1}\big)^2} \; + \; \sum_{i \neq 1} \sum_j \frac{1}{\Big(\big(\mu_{i,j} - \mu_{i,1}\big) \vee \big(\mu_{1,1} - \mu_{i,1}\big)\Big)^2}. \tag{8}$$

A direct comparison shows that both complexities in Garivier et al. [12] match ours up to constant factors:

$$H_1(0) \;\leq\; H_1^\star(\mu) \;\leq\; 4H_1(0), \qquad H_1(0) \;\leq\; H_2^\star(\mu) \;\leq\; 4H_1(0).$$

In the special case $\varepsilon = 0$, our fixed-budget sample complexity thus matches their fixed-confidence sample complexity up to logarithmic factors.

Kaufmann and Koolen [16] study MCTS as best-arm identification in trees of arbitrary depth in the fixed-confidence setting, with an ε-relaxation. Specializing to depth-2 maximin trees, their sample complexity is

$$H_\varepsilon^\star(\mu) = \sum_j \frac{1}{\big((\mu_{1,1} - \mu_{2,1}) \vee \varepsilon\big)^2} \; + \; \sum_{i \neq 1} \sum_j \frac{1}{\big((\mu_{1,1} - \mu_{i,1}) \vee \varepsilon\big)^2}.$$

In contrast, our leafwise sample complexity (for ε-agnostic identification) is

$$H_1(\varepsilon) = \sum_j \frac{1}{\big((\mu_{1,j} - \mu_{2,1}) \vee \varepsilon\big)^2} \; + \; \sum_{i \neq 1} \sum_j \frac{1}{\Big(\big((\mu_{i,j} - \mu_{i,1}) \vee (\mu_{1,1} - \mu_{i,1})\big) \vee \varepsilon\Big)^2},$$

The key difference is that $H_\varepsilon^\star(\mu)$ uses the same tree-level separation $(\mu_{1,1} - \mu_{i,1})$ for every leaf in subtree i, and thus repeatedly "pays" for leaves that are not determining the min value of a sub-optimal subtree. Their $H_\varepsilon^\star(\mu)$ can be much larger than our $H_1(\varepsilon)$ (and hence than our $H_2(\varepsilon)$), because $H_1(\varepsilon)$ depends on the leaf-specific gap quantity

$$(\mu_{i,j} - \mu_{i,1}) \vee (\mu_{1,1} - \mu_{i,1}),$$

which can be arbitrarily large for most leaves even when $(\mu_{1,1} - \mu_{i,1})$ is small.

Huang et al. [13] generalizes the MCTS identification problem to arbitrary structure in the fixed-confidence setting. When specialized to depth-2 maximin trees, their instance-dependent complexity reduces to the same form as in Garivier et al. [12], and therefore the same order with our $H_1(0)$.

Fixed-budget complexity and optimality. A recent line of work shows that optimal sample-complexity characterizations in the fixed-budget setting are substantially more delicate than in the fixed-confidence setting, even for ordinary best-arm identification. Komiyama et al. [21] study minimax optimality for fixed-budget BAI through normalized error exponents. They emphasize that, unlike in fixed confidence, a fixed-budget algorithm cannot in general fully adapt to every instance: improving the probability of error on one instance may worsen it on another. Their optimal rate R^{go} is defined through an oracle allocation that depends on the final empirical distribution after T rounds, and the corresponding tracking problem is nontrivial. They also introduce a second rate R_∞^{go} and a conceptual delayed optimal tracking algorithm achieving it, but the resulting procedure is computationally almost infeasible.

This perspective is reinforced by Degenne [8], who studies whether fixed-budget identification tasks admit a "complexity" analogous to the fixed-confidence characteristic time: a lower bound on the error exponent that is attained by a single algorithm on all instances. He shows that such a complexity need not exist, even in simple best-arm identification problems including Bernoulli BAI with two arms and Gaussian BAI for sufficiently large K. These results suggest that, in fixed-budget problems, there may be no single universally optimal instance-dependent complexity attained uniformly by a practical adaptive algorithm.

Our work is consistent with this viewpoint. We do not claim a minimax-optimal fixed-budget exponent for max–min trees. Instead, we give an explicit finite-budget, instance-dependent upper bound for an ε-agnostic Successive-Rejects-style algorithm. The remaining gap between our upper and lower bounds should therefore be interpreted in light of the broader difficulty of fixed-budget optimality: even in unstructured BAI, the existence and attainability of a sharp universal complexity is subtle.

B Proof of Theorem 2

Throughout the analysis, we assume without loss of generality that the critical gap $\Delta^\star$ is well-defined, i.e., not all subtrees are ε-good. Indeed, if every subtree is ε-good, then every output is acceptable and the misidentification probability is identically zero, so the theorem holds trivially.

B.1 Warmup: Analysis of phase 1

Recap on the empirical minimizers and empirical gaps. Recall from Section 3.1, for a fixed phase k, let $\hat{\mu}_{i,j}$ denote the empirical mean of leaf (i,j) computed from all samples collected so far. Recall that $\mathcal{A}$ is the current set of active leaves. and, for each subtree i, let $\mathcal{A}_i := \mathcal{A} \cap (\{i\} \times [L])$ be its active leaves. For each active subtree i (i.e., $\mathcal{A}_i \neq \emptyset$), define its empirical minimizer index

$$\hat{1}(i) \in \arg\min_{(i,j)\in\mathcal{A}_i} \hat{\mu}_{i,j},$$

and the corresponding empirical min-value $\hat{v}_i := \hat{\mu}_{i,\hat{1}(i)}$. Define the empirically best subtree

$$\hat{a} \in \arg\max_{i:\ \mathcal{A}_i\neq\emptyset} \hat{v}_i.$$

We further define the empirically second-best subtree

$$\hat{b} \in \arg\max_{i:\ \mathcal{A}_i\neq\emptyset,\ i\neq\hat{a}} \hat{v}_i,$$

breaking ties arbitrarily. Recall as well the definition of empirical gaps $\hat{\Delta}_{i,j}$ from Section 3.1 and the definion of the empirically largest gap $\hat{\Delta}_{\max} = \max_{(i,j)\in\mathcal{A}} \hat{\Delta}_{i,j}$ from line 13 in Algorithm 2. Note that in phase 1, $\mathcal{A} = [K] \times [L]$.

Lemma 9. *At phase 1, if all leaves satisfy that* $|\hat{\mu} - \mu| < \frac{\Delta_{(KL)}}{8}$*, then* $\hat{\Delta}_{\max} > \frac{3}{4}\Delta_{(KL)}$.

Proof. 1. If $\Delta_{(KL)}$ is given by $\mu_{x,L} - \mu_{x,1}$ for some $x \neq 1$.
If x is not the empirically best subtree,

$$\hat{\Delta}_{x,L} = \max(\hat{\mu}_{x,L} - \hat{\mu}_{x,\hat{1}(x)}, \hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{x,\hat{1}(x)}) \geq \hat{\mu}_{x,L} - \hat{\mu}_{x,\hat{1}(x)} \geq \hat{\mu}_{x,L} - \hat{\mu}_{x,1} > \frac{3}{4}\Delta_{(KL)}$$

If x is the empirically best subtree,

$$\hat{\Delta}_{x,L} = \hat{\mu}_{x,L} - \hat{\mu}_{\hat{b},\hat{1}(\hat{b})} \geq \hat{\mu}_{x,L} - \hat{\mu}_{x,\hat{1}(x)} \geq \hat{\mu}_{x,L} - \hat{\mu}_{x,1} > \frac{3}{4}\Delta_{(KL)}$$

Hence $\hat{\Delta}_{\max} \geq \hat{\Delta}_{x,L} > \frac{3}{4}\Delta_{(KL)}$.

2. If $\Delta_{(KL)}$ is given by $\mu_{1,L} - \mu_{2,1}$. In this case the condition that $|\hat{\mu} - \mu| < \frac{\Delta_{(KL)}}{8}$ for all leaves ensures that $\hat{\mu}_{1,L} - \hat{\mu}_{x,1} > \frac{3}{4}\Delta_{(KL)}, \forall x \neq 1$.
If 1 is not the empirically best subtree, then there $\exists x \neq 1$ such that x is the empirically best subtree,

$$\hat{\Delta}_{1,L} = \max(\hat{\mu}_{1,L} - \hat{\mu}_{1,\hat{1}(1)}, \hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{1,\hat{1}(1)}) \geq \hat{\mu}_{1,L} - \hat{\mu}_{1,\hat{1}(1)} \geq \hat{\mu}_{1,L} - \hat{\mu}_{x,\hat{1}(x)} \geq \hat{\mu}_{1,L} - \hat{\mu}_{x,1} > \frac{3}{4}\Delta_{(KL)}$$

If 1 is the empirically best subtree,

$$\hat{\Delta}_{1,L} = \hat{\mu}_{1,L} - \hat{\mu}_{\hat{b},\hat{1}(\hat{b})} \geq \hat{\mu}_{1,L} - \hat{\mu}_{\hat{b},1} > \frac{3}{4}\Delta_{(KL)}$$

Hence $\hat{\Delta}_{\max} \geq \hat{\Delta}_{1,L} > \frac{3}{4}\Delta_{(KL)}$.

3. If $\Delta_{(KL)}$ is given by $\mu_{1,1} - \mu_{K,1}$. In this case the condition that $|\hat{\mu} - \mu| < \frac{\Delta_{(KL)}}{8}$ for all leaves ensures that $\hat{\mu}_{1,j} - \hat{\mu}_{K,1} > \frac{3}{4}\Delta_{(KL)}, \forall j \in [L]$. Hence, K is not the empirically best subtree, and

$$\begin{aligned}\hat{\Delta}_{K,1} &= \max(\hat{\mu}_{K,1} - \hat{\mu}_{K,\hat{1}(K)}, \hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{K,\hat{1}(K)}) \\ &\geq \hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{K,\hat{1}(K)} \geq \hat{\mu}_{1,\hat{1}(1)} - \hat{\mu}_{K,\hat{1}(K)} \geq \hat{\mu}_{1,\hat{1}(1)} - \hat{\mu}_{K,1} > \frac{3}{4}\Delta_{(KL)}\end{aligned}$$

□

Lemma 10. *At phase 1, if all leaves satisfy that* $|\hat{\mu} - \mu| < \frac{\Delta_{(KL)}}{8}$*, then the optimal leaves in suboptimal subtrees, i.e.,* $\{(y, 1) : y \neq 1\}$*, will not be eliminated by the single leaf elimination rule.*

Proof. Let $y \neq 1$, we will show that if $|\hat{\mu} - \mu| < \frac{\Delta_{(KL)}}{8}$ for all leaves, then $(y, 1)$ will not be eliminated by the single-leaf elimination rule.

1. If $\hat{\Delta}_{y,1}$ is given by $\hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{y,\hat{1}(y)}$,
   (a) If there exists a leaf (y, j) such that $\hat{\mu}_{y,j} > \hat{\mu}_{\hat{a},\hat{1}(\hat{a})}$, then

   $$\hat{\Delta}_{y,j} = \max(\hat{\mu}_{y,j} - \hat{\mu}_{y,\hat{1}(y)}, \hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{y,\hat{1}(y)}) = \hat{\mu}_{y,j} - \hat{\mu}_{y,\hat{1}(y)} > \hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{y,\hat{1}(y)} = \hat{\Delta}_{y,1}$$

   hence $(y, 1)$ will not be eliminated by the single-leaf elimination rule.
   (b) If all leaves (y, j)'s are such that $\hat{\mu}_{y,j} \leq \hat{\mu}_{\hat{a},\hat{1}(\hat{a})}$, then $\forall j$,

   $$\hat{\Delta}_{y,j} = \max(\hat{\mu}_{y,j} - \hat{\mu}_{y,\hat{1}(y)}, \hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{y,\hat{1}(y)}) = \hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{y,\hat{1}(y)} = \hat{\Delta}_{y,1}$$

   hence either subtree y will be eliminated entirely, or one leaf not in y will be eliminated, or leaf/leaves in another subtree not y will be eliminated.
2. If $\hat{\Delta}_{y,1}$ is given by $\hat{\mu}_{y,1} - \hat{\mu}_{y,\hat{1}(y)}$, then

   $$\hat{\mu}_{y,1} - \hat{\mu}_{y,\hat{1}(y)} = \hat{\mu}_{y,1} - \mu_{y,1} + \mu_{y,1} - \mu_{y,\hat{1}(y)} + \mu_{y,\hat{1}(y)} - \hat{\mu}_{y,\hat{1}(y)} < \frac{\Delta_{(KL)}}{4}$$

   By Lemma 9, $\hat{\Delta}_{\max} > \frac{3}{4}\Delta_{(KL)}$, hence $\hat{\Delta}_{y,1} < \hat{\Delta}_{\max}$, and $(y, 1)$ will not be eliminated by the single-leaf elimination rule.
3. If y is the empirically best subtree and $\hat{\Delta}_{y,1}$ is given by $\hat{\mu}_{y,1} - \hat{\mu}_{\hat{b},\hat{1}(\hat{b})}$, then

   $$\hat{\mu}_{y,1} - \hat{\mu}_{\hat{b},\hat{1}(\hat{b})} \leq \hat{\mu}_{y,1} - \hat{\mu}_{1,\hat{1}(1)} = \hat{\mu}_{y,1} - \mu_{y,1} + \mu_{y,1} - \mu_{1,\hat{1}(1)} + \mu_{1,\hat{1}(1)} - \hat{\mu}_{1,\hat{1}(1)} < \frac{\Delta_{(KL)}}{4}$$

   where the first inequality is because y is the empirically best subtree, and hence $\hat{\mu}_{\hat{b},\hat{1}(\hat{b})} \geq \hat{\mu}_{1,\hat{1}(1)}$.
   By Lemma 9, $\hat{\Delta}_{\max} > \frac{3}{4}\Delta_{(KL)}$, hence $\hat{\Delta}_{y,1} < \hat{\Delta}_{\max}$, and $(y, 1)$ will not be eliminated by the single-leaf elimination rule.

□

Lemma 11. *At phase 1, if all leaves satisfy that* $|\hat{\mu} - \mu| < \frac{\Delta_{(KL)}}{8}$*, then subtree* 1 *will not become empty.*

Proof. We note that it suffices to show that subtree 1 will not be eliminated by the subtree elimination rule since the single-leaf elimination rule for a single-leaf subtree coincides with the subtree elimination rule.

We prove this lemma by contradiction in each of the following cases.

1. If $\Delta_{(KL)}$ is given by $\mu_{x,L} - \mu_{x,1}$ for some $x \neq 1$.
   Since $|\hat{\mu} - \mu| < \frac{\Delta_{(KL)}}{8}$ for all leaves, we have that $\hat{\mu}_{x,L} - \hat{\mu}_{x,1} > \frac{3}{4}\Delta_{(KL)}$. If it were the case that the subtree x is the empirically best subtree, and subtree 1 gets eliminated, then $\hat{\Delta}_{x,L} \leq \hat{\Delta}_{1,1}$. However, $\hat{\Delta}_{x,L} = \hat{\mu}_{x,L} - \hat{\mu}_{\hat{b},\hat{1}(\hat{b})} \geq \hat{\mu}_{x,L} - \hat{\mu}_{x,\hat{1}(x)} \geq \hat{\mu}_{x,L} - \hat{\mu}_{x,1} > \frac{3}{4}\Delta_{(KL)}$, but $\hat{\Delta}_{1,1} = \hat{\mu}_{x,\hat{1}(x)} - \hat{\mu}_{1,1} \leq \hat{\mu}_{x,1} - \hat{\mu}_{1,1} = \hat{\mu}_{x,1} - \mu_{x,1} + \mu_{x,1} - \mu_{1,1} + \mu_{1,1} - \hat{\mu}_{1,1} < \frac{1}{4}\Delta_{(KL)}$. Contradiction.
   If it were the case that the subtree y, with $y \neq 1,\ y \neq x$ is the empirically best subtree, and subtree 1 gets eliminated, then $\hat{\Delta}_{x,L} \leq \hat{\Delta}_{1,1}$. However, $\hat{\Delta}_{x,L} \geq \hat{\mu}_{x,L} - \hat{\mu}_{x,\hat{1}(x)} \geq \hat{\mu}_{x,L} - \hat{\mu}_{x,1} > \frac{3}{4}\Delta_{(KL)}$, but $\hat{\Delta}_{1,1} = \hat{\mu}_{y,\hat{1}(y)} - \hat{\mu}_{1,1} \leq \hat{\mu}_{y,1} - \hat{\mu}_{1,1} = \hat{\mu}_{y,1} - \mu_{y,1} + \mu_{y,1} - \mu_{1,1} + \mu_{1,1} - \hat{\mu}_{1,1} < \frac{1}{4}\Delta_{(KL)}$. Contradiction.

2. If $\Delta_{(KL)}$ is given by $\mu_{1,L} - \mu_{2,1}$.

   In this case the condition that $|\hat{\mu} - \mu| < \frac{\Delta_{(KL)}}{8}$ for all leaves ensures that $\hat{\mu}_{1,L} - \hat{\mu}_{x,1} > \frac{3}{4}\Delta_{(KL)}, \forall x \neq a$. If subtree 1 were eliminated, then there $\exists x \neq 1$ such that x is the empirically best subtree and $\hat{\mu}_{x,1} \geq \hat{\mu}_{x,\hat{1}(x)} \geq \hat{\mu}_{1,L}$, which leads to a contradiction.

3. If $\Delta_{(KL)}$ is given by $\mu_{1,1} - \mu_{K,1}$.

   In this case the condition that $|\hat{\mu} - \mu| < \frac{\Delta_{(KL)}}{8}$ for all leaves ensures that $\hat{\mu}_{1,j} - \hat{\mu}_{K,1} > \frac{3}{4}\Delta_{(KL)}, \forall j \in [L]$. If subtree 1 were eliminated, then there $\exists x \neq 1$ such that x is the empirically best subtree and $\forall j, \hat{\Delta}_{1,j} = \hat{\mu}_{x,\hat{1}(x)} - \hat{\mu}_{1,j} = \hat{\mu}_{x,\hat{1}(x)} - \mu_{x,\hat{1}(x)} + \mu_{x,\hat{1}(x)} - \mu_{1,j} + \mu_{1,j} - \hat{\mu}_{1,j} \leq \frac{1}{4}\Delta_{(KL)}$, which leads to a contradiction with that, $\forall j, \hat{\Delta}_{1,j} = \hat{\Delta}_{\max}$, since $\hat{\Delta}_{\max} > \frac{3}{4}\Delta_{(KL)}$ by Lemma 9.

□

Lemma 12. *Let subtree $s \neq 1$ be $\varepsilon/2$-good. At phase 1, if all leaves satisfy that $|\hat{\mu} - \mu| < \frac{\Delta_{(KL)}}{8}$, then $(s, 1)$ will not be eliminated.*

Proof. In the proof of Lemma 10, in cases 2 and 3, we've shown that $\widehat{\Delta}_{s,1} < \frac{\Delta_{(KL)}}{4}$. We will now upper bound $\widehat{\Delta}_{s,1}$ in case 1 where $\hat{\Delta}_{s,1}$ is given by $\hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{s,\hat{1}(s)}$.

$$\begin{aligned}
\hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{s,\hat{1}(s)} &\leq \hat{\mu}_{\hat{a},1} - \hat{\mu}_{s,\hat{1}(s)} \\
&< \mu_{\hat{a},1} + \frac{\Delta_{(KL)}}{8} - \mu_{s,\hat{1}(s)} + \frac{\Delta_{(KL)}}{8} \\
&\leq \mu_{1,1} - \mu_{s,\hat{1}(s)} + \frac{\Delta_{(KL)}}{4} \\
&\leq \mu_{1,1} - \mu_{s,1} + \frac{\Delta_{(KL)}}{4} \\
&\leq \frac{\varepsilon}{2} + \frac{\Delta_{(KL)}}{4} && (s \text{ is } \varepsilon/2\text{-good}) \\
&< \frac{3\Delta_{(KL)}}{4} && (\varepsilon < \Delta^* \leq \Delta_{(KL)})
\end{aligned}$$

By Lemma 9, $\hat{\Delta}_{\max} > \frac{3}{4}\Delta_{(KL)}$, hence $\hat{\Delta}_{s,1} < \hat{\Delta}_{\max}$, and $(s, 1)$ will not be eliminated □

B.2 Analysis of a generic phase k in early regime ($k \leq KL - m$)

Phase-k (restricted) gaps. Recall that $\nu = \{\nu_{i,j}\}_{(i,j)\in[K]\times[L]}$ is the original instance. At the beginning of phase k, the algorithm maintains an active leaf set $\mathcal{A}^{(k)} \subseteq [K] \times [L]$. For each subtree i, let $\mathcal{A}_i^{(k)} := \mathcal{A}^{(k)} \cap (\{i\} \times [L])$ denote its active leaves; we call subtree i *active* if $\mathcal{A}_i^{(k)} \neq \emptyset$. The eliminations performed in earlier phases effectively restrict the instance to the remaining leaves. Accordingly, we define *phase-k gaps* by applying the same max-min gap construction as in (3)–(4), but with all minima and competitor comparisons taken *within the remaining tree*.

Formally, define the (mean) min-value of subtree i restricted to active leaves as

$$v_i^{(k)} := \min\{\mu_{i,j} : (i,j) \in \mathcal{A}^{(k)}\}, \qquad i \in [K] \text{ with } \mathcal{A}_i^{(k)} \neq \emptyset,$$

and let $i_*^{(k)} \in \arg\max_{i:\mathcal{A}_i^{(k)}\neq\emptyset} v_i^{(k)}$ be a best active subtree under this restriction. Let $v_{(2)}^{(k)} := \max\{v_i^{(k)} : i \neq i_*^{(k)},\ \mathcal{A}_i^{(k)} \neq \emptyset\}$ denote the second-best active min-value (breaking ties arbitrarily).

Equivalently, $\Delta_{i,j}^{(k)}$ is the "gap" that would be assigned to leaf (i,j) if one considered the induced instance consisting only of active leaves at phase k.

With this interpretation, we define

$$\Delta_{i,j}^{(k)} := \begin{cases} \mu_{i,j} - v_{(2)}^{(k)}, & \text{if } i = i_*^{(k)}, \\ \max\left\{ v_{i_*^{(k)}}^{(k)} - v_i^{(k)},\ \mu_{i,j} - v_i^{(k)} \right\}, & \text{if } i \neq i_*^{(k)}. \end{cases}$$

Furthermore, define

$$\Delta_{\max}^{(k)} := \max_{(i,j)\in\mathcal{A}^{(k)}} \Delta_{i,j}^{(k)}. \tag{9}$$

Phase-k residual instance. We define the *phase-k residual instance* as the restriction of ν to active leaves,

$$\nu^{(k)} := \{\nu_{i,j} : (i,j) \in \mathcal{A}^{(k)}\}.$$

In the rest of this section, for the simplicity of the notation, and to focus on the logic through phases of elimination, we denote by:

- a, the optimal subtree within $\nu^{(k)}$, i.e., $a := i_*^{(k)}$.
- b, the second optimal subtree within $\nu^{(k)}$, i.e., $b \in \arg\max_{i:i\neq i_*^{(k)}, \mathcal{A}_i^{(k)}\neq\emptyset} v_i^{(k)}$.
- z, the least optimal subtree within $\nu^{(k)}$, i.e., $z \in \arg\min_{i:\mathcal{A}_i^{(k)}\neq\emptyset} v_i^{(k)}$.

Recall that in Appendix B.1, with the same spirit, we denoted by $\hat{a}$ the empirically optimal subtree within the reduced instance, and $\hat{b}$ the empirically second optimal subtree within the reduced instance. Define the active-maximum index

$$h_i \in \arg\max_{(i,j)\in\mathcal{A}_x^{(k)}} \mu_{i,j}. \tag{10}$$

Note that we omit the phase index k on $a, b, z, \hat{a}, \hat{b}, h_i$, as the phase index is clear from the context in all of the following lemmas.

The next immediate lemma establishes the key invariants maintained by "ε-soundness":

1. Subtree 1 remains to be the optimal subtree in $\nu^{(k)}$;
2. The min leaf (which defines the sub-optimality of the subtree) is surviving, if the subtree is still active.

These are the key structures of a tree that our algorithm maintains with high probability with ε-soundness, so that the residual instance $\nu^{(k)}$ does not become "confusing" in terms of the max-min subtree compared to the original tree instance.

Lemma 13. *Let $k \leq KL - m$. Suppose after $k-1$ phases, the algorithm has not terminated, and phase k is ε-sound. Then,*

1. *$a = 1$.*
2. *For any $i \geq 2, j \in [L]$, if $(i,j) \in \mathcal{A}^{(k)}$, then $(i,1) \in \mathcal{A}^{(k)}$.*

Proof. The second statement follows directly from items 2 and 3 in Definition 6.

The first statement then follows from items 1–3. Indeed, subtree 1 is nonempty, and every other active subtree i retains its minimizing leaf $(i,1)$. Thus,

$$v_1^{(k)} \geq \mu_{1,1} > \mu_{i,1} = v_i^{(k)}$$

for every other active subtree i, which implies $a = 1$. □

Lemma 14. *Let $k \leq KL - m$. Suppose after $k-1$ phases, the algorithm has not terminated, and phase k is ε-sound. Then, for each $(i,j) \in \mathcal{A}^{(k)}$,*

$$\Delta_{i,j}^{(k)} \geq \Delta_{i,j}$$

Proof. Fix any $(i,j) \in \mathcal{A}$.

- Case 1: $i = 1$. By Lemma 13, $\Delta_{i,j}^{(k)}$ is given by $\mu_{i,j} - \mu_{x,1}$, for some subtree $x \neq 1$. Thus,

$$\Delta_{i,j}^{(k)} = \mu_{i,j} - \mu_{x,1} \geq \mu_{i,j} - \mu_{2,1} = \Delta_{i,j}.$$

- Case 2: $i \neq 1$. By the first item in Lemma 13, subtree i is a subtimal subtree; moreover,

$$\begin{aligned}
\Delta_{i,j}^{(k)} &= \max\left\{v_1^{(k)} - v_i^{(k)},\ \mu_{i,j} - v_i^{(k)}\right\} \\
&= \max\left\{v_1^{(k)} - \mu_{i,1},\ \mu_{i,j} - \mu_{i,1}\right\} && \text{(Second item in Lemma 13)} \\
&\geq \max\left\{\mu_{1,1} - \mu_{i,1},\ \mu_{i,j} - \mu_{i,1}\right\} && (v_1^{(k)} \geq \mu_{1,1}) \\
&= \Delta_{i,j}
\end{aligned}$$

□

Lemma 15. *Let $k \leq KL - m$. Suppose after $k-1$ phases, the algorithm has not terminated, and phase k is ε-sound. Let $\Delta_{\max}^{(k)}$ denote the max phase-k gap at the beginning of phase k (see Equation 9 for the definition), then $\Delta_{\max}^{(k)} \geq \Delta_{(KL+1-k)}$.*

Proof. At the beginning of phase k, there are $k-1$ leaves already eliminated and $KL-k+1$ leaves surviving, that is, $|\mathcal{A}^{(k)}| = KL+1-k$. We have,

$$\begin{aligned}
\Delta_{\max}^{(k)} &= \max_{(i,j)\in\mathcal{A}^{(k)}} \Delta_{i,j}^{(k)} \\
&\geq \max_{(i,j)\in\mathcal{A}^{(k)}} \Delta_{i,j} && \text{(Lemma 14)} \\
&\geq \Delta_{(KL+1-k)}
\end{aligned}$$

where the last inequality is because, for any $\mathcal{S} \subseteq [K]\times[L]$ with $|\mathcal{S}| = KL+1-k$, we have $\max_{(i,j)\in\mathcal{S}} \Delta_{i,j} \geq \Delta_{(KL+1-k)}$. □

The following lemmas are direct extensions of Lemma 9 to 11 and Lemma 12 to any phase $k \leq KL - m$. A similar proof idea follows with replacing the original instance ν (in Lemma 9 to 12) with phase-k residual instance $\nu^{(k)}$. Thanks to Lemma 15, in order to control the deviation ($|\hat{\mu} - \mu|$) with respect to $\Delta_{\max}^{(k)}$, it suffices to control it w.r.t. $\Delta_{(KL+1-k)}$.

Despite the similarity of the proofs for the following lemmas in this subsection compared to Lemmas 9 to 12, we include the proofs for completeness.

Lemma 16. *Let $k \leq KL - m$. Suppose after $k-1$ phases, the algorithm has not terminated, and phase k is ε-sound. If all leaves satisfy that $|\hat{\mu} - \mu| < \frac{\Delta_{(KL+1-k)}}{8}$, then $\hat{\Delta}_{\max} > \frac{3}{4}\Delta_{(KL+1-k)}$.*

Proof. Recall $\Delta_{\max}^{(k)}$ defined in Equation (9). Recall the active-maximum index h_i defined in Equation (10). By Lemma 15, all leaves satisfy that $|\hat{\mu} - \mu| < \frac{\Delta_{(KL+1-k)}}{8} \leq \frac{\Delta_{\max}^{(k)}}{8}$.

By the ε-soundness of phase k and Lemma 13, there are following three cases:

1. If $\Delta_{\max}^{(k)}$ is given by $\mu_{x,h_x} - \mu_{x,1}$ for some $x \neq a$.
   If x is not the empirically best subtree,

   $$\hat{\Delta}_{x,h_x} = \max(\hat{\mu}_{x,h_x} - \hat{\mu}_{x,\hat{1}(x)}, \hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{x,\hat{1}(x)}) \geq \hat{\mu}_{x,h_x} - \hat{\mu}_{x,\hat{1}(x)} \geq \hat{\mu}_{x,h_x} - \hat{\mu}_{x,1} > \frac{3}{4}\Delta_{\max}^{(k)}$$

   If x is the empirically best subtree,

   $$\hat{\Delta}_{x,h_x} = \hat{\mu}_{x,h_x} - \hat{\mu}_{\hat{b},\hat{1}(\hat{b})} \geq \hat{\mu}_{x,h_x} - \hat{\mu}_{x,\hat{1}(x)} \geq \hat{\mu}_{x,h_x} - \hat{\mu}_{x,1} > \frac{3}{4}\Delta_{\max}^{(k)}$$

   Hence $\hat{\Delta}_{\max} \geq \hat{\Delta}_{x,L} > \frac{3}{4}\Delta_{\max}^{(k)}$.

2. If $\Delta_{\max}^{(k)}$ is given by $\mu_{a,h_a} - \mu_{b,1}$. In this case the condition that $|\hat{\mu} - \mu| < \frac{\Delta_{\max}^{(k)}}{8}$ for all leaves ensures that $\hat{\mu}_{a,h_a} - \hat{\mu}_{x,1} > \frac{3}{4}\Delta_{\max}^{(k)}, \forall x \neq a$.
If a is not the empirically best subtree, then there $\exists x \neq a$ such that x is the empirically best subtree,

$$\hat{\Delta}_{a,h_a} = \max(\hat{\mu}_{a,h_a} - \hat{\mu}_{a,\hat{1}(a)}, \hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{a,\hat{1}(a)}) \geq \hat{\mu}_{a,h_a} - \hat{\mu}_{a,\hat{1}(a)} \geq \hat{\mu}_{a,h_a} - \hat{\mu}_{x,\hat{1}(x)} \geq \hat{\mu}_{a,h_a} - \hat{\mu}_{x,1} > \frac{3}{4}\Delta_{\max}^{(k)}$$

If a is the empirically best subtree,

$$\hat{\Delta}_{a,h_a} = \hat{\mu}_{a,h_a} - \hat{\mu}_{\hat{b},\hat{1}(\hat{b})} \geq \hat{\mu}_{a,h_a} - \hat{\mu}_{\hat{b},1} > \frac{3}{4}\Delta_{\max}^{(k)}$$

Hence $\hat{\Delta}_{\max} \geq \hat{\Delta}_{a,h_a} > \frac{3}{4}\Delta_{\max}^{(k)}$.

3. If $\Delta_{\max}^{(k)}$ is given by $\mu_{a,1} - \mu_{z,1}$. In this case the condition that $|\hat{\mu} - \mu| < \frac{\Delta_{\max}^{(k)}}{8}$ for all leaves ensures that $\hat{\mu}_{a,j} - \hat{\mu}_{z,1} > \frac{3}{4}\Delta_{\max}^{(k)}, \forall j$ such that $(a,j) \in \mathcal{A}^{(k)}$. Hence, z is not the empirically best subtree, and

$$\begin{aligned}\hat{\Delta}_{z,1} = \max(\hat{\mu}_{z,1} - \hat{\mu}_{z,\hat{1}(z)}, \hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{z,\hat{1}(z)}) \geq \hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{z,\hat{1}(z)} &\geq \hat{\mu}_{a,\hat{1}(a)} - \hat{\mu}_{z,\hat{1}(z)} \\ &\geq \hat{\mu}_{a,\hat{1}(a)} - \hat{\mu}_{z,1} \\ &> \frac{3}{4}\Delta_{\max}^{(k)}\end{aligned}$$

Therefore, in all the 3 cases, $\hat{\Delta}_{\max} > \frac{3}{4}\Delta_{\max}^{(k)} \geq \frac{3}{4}\Delta_{(KL+1-k)}$, where the last inequality is by Lemma 15. □

Lemma 17. *Let $k \leq KL - m$. Suppose after $k-1$ phases, the algorithm has not terminated, and phase k is ε-sound. If all leaves satisfy that $|\hat{\mu} - \mu| < \frac{\Delta_{(KL+1-k)}}{8}$, then the optimal leaves in suboptimal subtrees, i.e., $\{(y,1) : y \neq a\}$, will not be eliminated by the single leaf elimination rule.*

Proof. By Lemma 15, all leaves satisfy that $|\hat{\mu} - \mu| < \frac{\Delta_{(KL+1-k)}}{8} \leq \frac{\Delta_{\max}^{(k)}}{8}$.

Let $y \neq a$, we will show that if $|\hat{\mu} - \mu| < \frac{\Delta_{\max}^{(k)}}{8}$ for all leaves, then $(y,1)$ will not be eliminated by the single-leaf elimination rule.

1. If $\hat{\Delta}_{y,1}$ is given by $\hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{y,\hat{1}(y)}$,
   (a) If there exists a leaf (y,j) such that $\hat{\mu}_{y,j} > \hat{\mu}_{\hat{a},\hat{1}(\hat{a})}$, then

   $$\hat{\Delta}_{y,j} = \max(\hat{\mu}_{y,j} - \hat{\mu}_{y,\hat{1}(y)}, \hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{y,\hat{1}(y)}) = \hat{\mu}_{y,j} - \hat{\mu}_{y,\hat{1}(y)} > \hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{y,\hat{1}(y)} = \hat{\Delta}_{y,1}$$

   hence $(y,1)$ will not be eliminated by the single-leaf elimination rule.
   (b) If all leaves (y,j)'s are such that $\hat{\mu}_{y,j} \leq \hat{\mu}_{\hat{a},\hat{1}(\hat{a})}$, then $\forall j$,

   $$\hat{\Delta}_{y,j} = \max(\hat{\mu}_{y,j} - \hat{\mu}_{y,\hat{1}(y)}, \hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{y,\hat{1}(y)}) = \hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{y,\hat{1}(y)} = \hat{\Delta}_{y,1}$$

   hence either subtree y will be eliminated entirely, or one leaf not in y will be eliminated, or another subtree not y will be eliminated.
2. If $\hat{\Delta}_{y,1}$ is given by $\hat{\mu}_{y,1} - \hat{\mu}_{y,\hat{1}(y)}$, then

   $$\hat{\mu}_{y,1} - \hat{\mu}_{y,\hat{1}(y)} = \hat{\mu}_{y,1} - \mu_{y,1} + \mu_{y,1} - \mu_{y,\hat{1}(y)} + \mu_{y,\hat{1}(y)} - \hat{\mu}_{y,\hat{1}(y)} < \frac{\Delta_{\max}^{(k)}}{4}$$

   By Lemma 16, $\hat{\Delta}_{\max} > \frac{3}{4}\Delta_{\max}^{(k)}$, hence $\hat{\Delta}_{y,1} < \hat{\Delta}_{\max}$, and $(y,1)$ will not be eliminated by the single-leaf elimination rule.
3. If y is the empirically best subtree and $\hat{\Delta}_{y,1}$ is given by $\hat{\mu}_{y,1} - \hat{\mu}_{\hat{b},\hat{1}(\hat{b})}$, then

   $$\hat{\mu}_{y,1} - \hat{\mu}_{\hat{b},\hat{1}(\hat{b})} \leq \hat{\mu}_{y,1} - \hat{\mu}_{a,\hat{1}(a)} = \hat{\mu}_{y,1} - \mu_{y,1} + \mu_{y,1} - \mu_{a,\hat{1}(a)} + \mu_{a,\hat{1}(a)} - \hat{\mu}_{a,\hat{1}(a)} < \frac{\Delta_{\max}^{(k)}}{4}$$

where the first inequality is because y is the empirically best subtree, and hence $\hat{\mu}_{\hat{b},\hat{1}(\hat{b})} \geq \hat{\mu}_{a,\hat{1}(a)}$.

By Lemma 16, $\hat{\Delta}_{\max} > \frac{3}{4}\Delta^{(k)}_{\max}$, hence $\hat{\Delta}_{y,1} < \hat{\Delta}_{\max}$, and $(y,1)$ will not be eliminated by the single-leaf elimination rule.

□

Lemma 18. *Let $k \leq KL - m$. Suppose after $k-1$ phases, the algorithm has not terminated, and phase k is ε-sound. If all leaves satisfy that $|\hat{\mu} - \mu| < \frac{\Delta_{(KL+1-k)}}{8}$, then subtree a will not become empty.*

Proof. By Lemma 15, all leaves satisfy that $|\hat{\mu} - \mu| < \frac{\Delta_{(KL+1-k)}}{8} \leq \frac{\Delta^{(k)}_{\max}}{8}$.

We note that it suffices to show that subtree a will not be eliminated by the subtree elimination rule since the single-leaf elimination rule for a single-leaf subtree coincides with the subtree elimination rule.

We prove this lemma by contradiction in each of the following cases.

1. If $\Delta^{(k)}_{\max}$ is given by $\mu_{x,h_x} - \mu_{x,1}$ for some $x \neq a$.

   Since $|\hat{\mu} - \mu| < \frac{\Delta^{(k)}_{\max}}{8}$ for all leaves, we have that $\hat{\mu}_{x,h_x} - \hat{\mu}_{x,1} > \frac{3}{4}\Delta^{(k)}_{\max}$. If it were the case that the subtree x is the empirically best subtree, and subtree a gets eliminated, then, for any (a,j) such that $(a,j) \in \mathcal{A}^{(k)}_a$, $\hat{\Delta}_{x,h_x} \leq \hat{\Delta}_{a,j}$. However, $\hat{\Delta}_{x,h_x} = \hat{\mu}_{x,h_x} - \hat{\mu}_{\hat{b},\hat{1}(\hat{b})} \geq \hat{\mu}_{x,h_x} - \hat{\mu}_{x,\hat{1}(x)} \geq \hat{\mu}_{x,h_x} - \hat{\mu}_{x,1} > \frac{3}{4}\Delta^{(k)}_{\max}$, but $\hat{\Delta}_{a,j} = \hat{\mu}_{x,\hat{1}(x)} - \hat{\mu}_{a,j} \leq \hat{\mu}_{x,1} - \hat{\mu}_{a,j} = \hat{\mu}_{x,1} - \mu_{x,1} + \mu_{x,1} - \mu_{a,j} + \mu_{a,j} - \hat{\mu}_{a,j} < \frac{1}{4}\Delta^{(k)}_{\max}$. Contradiction.
   If it were the case that the subtree y with $y \neq a, x$ is the empirically best subtree, and subtree a gets eliminated, then, for any (a,j) such that $(a,j) \in \mathcal{A}^{(k)}_a$, $\hat{\Delta}_{x,h_x} \leq \hat{\Delta}_{a,j}$. However, $\hat{\Delta}_{x,h_x} \geq \hat{\mu}_{x,h_x} - \hat{\mu}_{x,\hat{1}(x)} \geq \hat{\mu}_{x,h_x} - \hat{\mu}_{x,1} > \frac{3}{4}\Delta^{(k)}_{\max}$, but $\hat{\Delta}_{a,j} = \hat{\mu}_{y,\hat{1}(y)} - \hat{\mu}_{a,j} \leq \hat{\mu}_{y,1} - \hat{\mu}_{a,j} = \hat{\mu}_{y,1} - \mu_{y,1} + \mu_{y,1} - \mu_{a,j} + \mu_{a,j} - \hat{\mu}_{a,j} < \frac{1}{4}\Delta^{(k)}_{\max}$. Contradiction.
2. If $\Delta^{(k)}_{\max}$ is given by $\mu_{a,h_a} - \mu_{b,1}$.

   In this case the condition that $|\hat{\mu} - \mu| < \frac{\Delta^{(k)}_{\max}}{8}$ for all leaves ensures that $\hat{\mu}_{a,h_a} - \hat{\mu}_{x,1} > \frac{3}{4}\Delta^{(k)}_{\max}, \forall$ active $x \neq a$. If subtree a were eliminated, then there $\exists$ active $x \neq a$ such that x is the empirically best subtree and $\hat{\mu}_{x,1} \geq \hat{\mu}_{x,\hat{1}(x)} \geq \hat{\mu}_{a,h_a}$, which leads to a contradiction.
3. If $\Delta^{(k)}_{\max}$ is given by $\mu_{a,1} - \mu_{z,1}$.

   In this case the condition that $|\hat{\mu} - \mu| < \frac{\Delta^{(k)}_{\max}}{8}$ for all leaves ensures that $\hat{\mu}_{a,j} - \hat{\mu}_{z,1} > \frac{3}{4}\Delta^{(k)}_{\max}, \forall j$ such that $(a,j) \in \mathcal{A}^{(k)}_a$. If subtree a were eliminated, then there $\exists x \neq a$ such that x is the empirically best subtree and $\forall j$ such that $(a,j) \in \mathcal{A}^{(k)}_a$, $\hat{\Delta}_{a,j} = \hat{\mu}_{x,\hat{1}(x)} - \hat{\mu}_{a,j} = \hat{\mu}_{x,\hat{1}(x)} - \mu_{x,\hat{1}(x)} + \mu_{x,\hat{1}(x)} - \mu_{a,j} + \mu_{a,j} - \hat{\mu}_{a,j} \leq \frac{1}{4}\Delta^{(k)}_{\max}$, which leads to a contradiction with $\forall j$ such that $(a,j) \in \mathcal{A}^{(k)}_a$, $\hat{\Delta}_{a,j} = \hat{\Delta}_{\max}$, since $\hat{\Delta}_{\max} > \frac{3}{4}\Delta^{(k)}_{\max}$ by Lemma 16.

□

Lemma 19. *Let $k \leq KL - m$. Suppose after $k-1$ phases, the algorithm has not terminated, and phase k is ε-sound. Let subtree $s \neq a$ be $\varepsilon/2$-good and surviving after $k-1$ phases. If all leaves satisfy that $|\hat{\mu} - \mu| < \frac{\Delta_{(KL+1-k)}}{8}$, then $(s,1)$ will not be eliminated in phase k.*

Proof. By Lemma 15, all leaves satisfy that $|\hat{\mu} - \mu| < \frac{\Delta_{(KL+1-k)}}{8} \leq \frac{\Delta^{(k)}_{\max}}{8}$.

In the proof of Lemma 17, in cases 2 and 3, we've shown that $\widehat{\Delta}_{s,1} < \frac{\Delta^{(k)}_{\max}}{4}$. We will now upper bound $\widehat{\Delta}_{s,1}$ in case 1 where $\hat{\Delta}_{s,1}$ is given by $\hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{s,\hat{1}(s)}$.

$$\begin{aligned}
\hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{s,\hat{1}(s)} &\le \hat{\mu}_{\hat{a},1} - \hat{\mu}_{s,\hat{1}(s)} \\
&< \mu_{\hat{a},1} + \frac{\Delta^{(k)}_{\max}}{8} - \mu_{s,\hat{1}(s)} + \frac{\Delta^{(k)}_{\max}}{8} \\
&\le \mu_{a,1} - \mu_{s,\hat{1}(s)} + \frac{\Delta^{(k)}_{\max}}{4} \\
&\le \mu_{a,1} - \mu_{s,1} + \frac{\Delta^{(k)}_{\max}}{4} \\
&\le \frac{\varepsilon}{2} + \frac{\Delta^{(k)}_{\max}}{4} && (s \text{ is } \varepsilon/2\text{-good}) \\
&< \frac{\Delta_{(KL+1-k)}}{2} + \frac{\Delta^{(k)}_{\max}}{4} && (\text{Lemma 20: } \varepsilon < \Delta^* \le \Delta_{(KL+1-k)}) \\
&\le \frac{3\Delta^{(k)}_{\max}}{4} && (\text{Lemma 15: } \Delta_{(KL+1-k)} \le \Delta^{(k)}_{\max})
\end{aligned}$$

By Lemma 16, $\hat{\Delta}_{\max} > \frac{3}{4}\Delta^{(k)}_{\max}$, hence $\hat{\Delta}_{s,1} < \hat{\Delta}_{\max}$, and $(s,1)$ will not be eliminated □

Lemma 20. *Recall the definition of Δ^* and m in Definition 1, we have, for any $k \in [KL-m]$,*

$$\varepsilon < \Delta^* \le \Delta_{(KL+1-k)}.$$

Proof. The first inequality, $\varepsilon < \Delta^*$, follows directly the definition of Δ^*.

For the second inequality,

$$\begin{aligned}
\Delta_{(KL+1-k)} &\ge \Delta_{(KL+1-(KL-m))} && (\Delta_{(\ell)} \text{ is non-decreasing in } \ell) \\
&= \Delta_{(m+1)} \\
&= \Delta^* && (\text{Definition of } m)
\end{aligned}$$

□

B.3 Analysis of a generic phase k in late regime ($k > KL - m$)

Lemma 21. *Let $k > KL - m$. Suppose after $k-1$ phases, the algorithm has not terminated, and phase k is ε-sound. Suppose t is an active nonempty ε-bad subtree at the beginning of phase k. If all leaves satisfy that $|\hat{\mu} - \mu| < \frac{\Delta^*}{8}$, then, $(t,1)$ is not eliminated by the single leaf elimination rule by the end of phase k.*

Proof. First, by item 5 in the soundness definition (Definition 6), $(t,1)$ is surviving at the beginning of phase k. Next, we note that t cannot be the empirically best subtree. This is because, by the item 4 of soundness, "either all $\varepsilon/2$-good subtrees are nonempty, or all ε-bad subtrees are empty", the existence of t implies that "all ε-bad subtrees are empty" cannot be true, hence it must be that "all $\varepsilon/2$-good subtrees are nonempty". Specifically, let s be an $\varepsilon/2$-good subtree, and for any surviving leaf (s,j),

$$\begin{aligned}
\hat{\mu}_{s,j} - \hat{\mu}_{t,\hat{1}(t)} &\ge \mu_{s,j} - \frac{\Delta^*}{8} - \hat{\mu}_{t,1} \\
&\ge \mu_{s,1} - \mu_{t,1} - \frac{\Delta^*}{4} \\
&= (\mu_{a,1} - \mu_{t,1}) - (\mu_{a,1} - \mu_{s,1}) - \frac{\Delta^*}{4} \\
&\ge \Delta^* - \varepsilon/2 - \frac{\Delta^*}{4} && (\mu_{a,1} - \mu_{t,1} \ge \Delta^*,\ \mu_{a,1} - \mu_{s,1} \le \varepsilon/2) \\
&> \frac{\Delta^*}{4} && (\text{Lemma 20: } \Delta^* > \varepsilon)
\end{aligned}$$

Hence, t is not the empirically best subtree, and $\hat{\Delta}_{t,1}$ is given by $\hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{t,\hat{1}(t)}$.

By the same argument as in case 1 in Lemma 10's proof, we have that $(t,1)$ is not eliminated by the single leaf elimination rule. □

Lemma 22. *Let $k > KL - m$. Suppose after $k-1$ phases, the algorithm has not terminated, and phase k is ε-sound. Suppose t is an active nonempty ε-bad subtree and s is an active nonempty $\varepsilon/2$-good subtree, at the beginning of phase k. If all leaves satisfy that $|\hat{\mu} - \mu| < \frac{\Delta^*}{8}$, then, s does not become empty in phase k.*

Proof. We note that it suffices to show that subtree s will not be eliminated by the subtree elimination rule since the single-leaf elimination rule for a single-leaf subtree coincides with the subtree elimination rule, which further implies that we only need to consider the case where s is not the empirically best.

By the reasoning in the previous lemma's proof, $(t,1)$ is surviving at the beginning of phase k. t is not the empirically best subtree, and $\hat{\Delta}_{t,1}$ is given by $\hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{t,\hat{1}(t)}$.

As a result, to show s does not become empty in phase k, it suffices to show $\hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{t,\hat{1}(t)} > \hat{\mu}_{\hat{a},\hat{1}(\hat{a})} - \hat{\mu}_{s,\hat{1}(s)}$, that is $\hat{\mu}_{t,\hat{1}(t)} < \hat{\mu}_{s,\hat{1}(s)}$. Indeed,

$$\begin{aligned}
\hat{\mu}_{s,\hat{1}(s)} - \hat{\mu}_{t,\hat{1}(t)} &\geq \mu_{s,\hat{1}(s)} - \frac{\Delta^*}{8} - \hat{\mu}_{t,1} \\
&\geq \mu_{s,1} - \mu_{t,1} - \frac{\Delta^*}{4} \\
&= (\mu_{a,1} - \mu_{t,1}) - (\mu_{a,1} - \mu_{s,1}) - \frac{\Delta^*}{4} \\
&\geq \Delta^* - \varepsilon/2 - \frac{\Delta^*}{4} && (\mu_{a,1} - \mu_{t,1} \geq \Delta^*,\ \mu_{a,1} - \mu_{s,1} \leq \varepsilon/2) \\
&> \frac{\Delta^*}{4} && (\text{Lemma 20: } \Delta^* > \varepsilon)
\end{aligned}$$

Therefore, s will not be eliminated by the subtree elimination rule, and s will not become empty in phase k. □

B.4 Proof of Theorem 2

Lemma 23. *A sufficient event for the SR algorithm to succeed in finding an ε-good subtree is (without the subscript, $\hat{\mu}, \mu$ refer to all leaves)*

$$A_1 := \begin{cases} \forall k \in [KL-m], \left|\hat{\mu}^{(k)} - \mu\right| < \frac{\Delta_{(KL+1-k)}}{8} \\ \forall k \in [KL-m+1, KL-1], \left|\hat{\mu}^{(k)} - \mu\right| < \frac{\Delta_{(m+1)}}{8} \end{cases}$$

Proof. Note that not all phases are executed. We first show by induction that under event A_1, for any phase k, phase k is sound.

- $k \leq KL - m$.
  - Base case: if $k = 1$, phase 1 is apparently sound since nothing has been operated on any leaves.
  - Induction case: Assume phase $k-1$ is sound.
    1. If phase $k-1$ is not executed, then phase k is automatically sound since no operation is done in phase $k-1$.
    2. If phase $k-1$ is executed, then by Lemma 17, Lemma 18 and Lemma 19, at phase $k-1$, the optimal(min) leaf in any suboptimal subtree is not eliminated alone, the optimal subtree does not become empty, and $\forall s \neq 1$ and $\varepsilon/2$-good, $(s,1)$ is not eliminated. It follows that phase k is sound, completing the induction for $k \leq KL - m$.

- $k > KL - m$.
  - Base case: $k = KL - m + 1$.
    1. If phase $KL - m$ is not executed, then by the induction on phases $k \le KL - m$, phase $KL - m$ is sound. Note that if a tree satisfies the soundness condition (Definition 6) for the earlier phases $k \le KL-m$, then it also satisfies the soundness condition for the later phases $k > KL - m + 1$. It follows that phase $KL - m + 1$ is automatically sound since no operation is done in phase $KL - m$.
    2. If phase $KL - m$ is executed, then by Lemma 17, Lemma 18 and Lemma 19, at phase $KL - m$, the optimal leaf in any suboptimal subtree is not eliminated alone, the optimal subtree does not become empty, and $\forall s \neq 1$ and $\varepsilon/2$-good, $(s, 1)$ is not eliminated. It follows that phase $KL - m + 1$ is sound.
  - Induction case: Assume phase $k - 1 \in [KL - m + 1, KL - 1]$ is sound.
    1. If phase $k - 1$ is not executed, then phase k is automatically sound since no operation is done in phase $k - 1$.
    2. If phase $k - 1$ is executed, then by Lemma 21 and Lemma 22, at phase $k - 1$, if there exists nonempty ε-bad subtree, then the optimal leaf in any ε-bad subtree is not eliminated alone, and any $\varepsilon/2$-good subtree does not become empty. It follows that phase k is sound, completing the induction for $k > KL - m$.

Therefore, for any phase k, phase k is sound. Suppose the algorithm terminates after exactly k' phases, so the termination situation satisfies the soundness condition in phase $k' + 1$.

- If $k'+1 \le KL-m$, then by the first item of the soundness definition for phase $k \le KL-m$, the best subtree is not empty. In conjunction with that the algorithm terminates when there is only one surviving subtree, we conclude that the algorithm succeeds in finding an ε-good subtree. (Actually, it outputs the best subtree.)
- If $k'+1 > KL-m$, then by the first item of the soundness definition for phase $k > KL-m$, "either all $\varepsilon/2$-good subtrees are nonempty, or all ε-bad subtrees are empty".
  - If the number of $\varepsilon/2$-good subtrees is 1, then either all ε-bad subtrees are empty, or the $\varepsilon/2$-good subtree is nonempty. In both cases, the only surviving subtree is ε-good.
  - Else if the number of $\varepsilon/2$-good subtrees is > 1, then since "all $\varepsilon/2$-good subtrees are nonempty" cannot be true, it must be that all ε-bad subtrees are empty, which implies that the only surviving subtree is ε-good.

□

Theorem 24 (Restatement of Theorem 2)**.** *Let K, L satisfy $KL \ge 2$, and suppose $T > KL$. Run Algorithm 2 with the phase schedule*

$$n_k = \left\lceil \frac{(T - KL)/\overline{\log}(KL)}{KL + 1 - k} \right\rceil, \qquad \overline{\log}(KL) := \frac{1}{2} + \sum_{r=2}^{KL} \frac{1}{r}.$$

Then, for every $\varepsilon \ge 0$ such that not all subtrees are ε-good, the failure probability of Successive Rejects for (K, L)-MCTS satisfies

$$\mathbb{P}\Big(\widehat{i}_T \notin G_\varepsilon\Big) \le 2K^2L^2 \exp\left(-\frac{T - KL}{128\,\overline{\log}(KL)\,H_2(\varepsilon)}\right).$$

If all subtrees are ε-good, then $\mathbb{P}(\widehat{i}_T \notin G_\varepsilon) = 0$.

Proof. If all subtrees are ε-good, then $G_\varepsilon = [K]$, and hence

$$\mathbb{P}\Big(\widehat{i}_T \notin G_\varepsilon\Big) = 0.$$

Thus the theorem is trivial in this case. In the rest of the proof, we assume that not all subtrees are ε-good, so that $G_\varepsilon \neq [K]$ and the critical gap $\Delta^\star$ and the index m in Definition 1 are well defined.

We first show that our budget allocation n_k is feasible, i.e., the total number of samples does not exceed the budget T.

Budget feasibility. Let $N := KL$ and define

$$\overline{\log}(N) := \frac{1}{2} + \sum_{r=2}^{N} \frac{1}{r}.$$

We use the phase schedule

$$n_k = \left\lceil \frac{(T-N)/\overline{\log}(N)}{N+1-k} \right\rceil, \qquad k = 1, \dots, N-1.$$

The worst case for the budget is when exactly one leaf is eliminated in each phase. In that case, phase k has $N+1-k$ active leaves, and the total number of pulls is

$$B = Nn_1 + \sum_{k=2}^{N-1} (N+1-k)(n_k - n_{k-1}).$$

By summation by parts,

$$B = \sum_{k=1}^{N-2} n_k + 2n_{N-1}.$$

Let

$$a := \frac{T-N}{\overline{\log}(N)}.$$

Since $\lceil x \rceil \le x + 1$,

$$\begin{aligned}
B &\le \sum_{k=1}^{N-2} \left(\frac{a}{N+1-k} + 1 \right) + 2\left(\frac{a}{2} + 1 \right) \\
&= a \left(\sum_{k=1}^{N-2} \frac{1}{N+1-k} + 1 \right) + N \\
&= a \left(\sum_{r=3}^{N} \frac{1}{r} + 1 \right) + N \\
&= a \left(\frac{1}{2} + \sum_{r=2}^{N} \frac{1}{r} \right) + N \\
&= a\,\overline{\log}(N) + N \\
&= T.
\end{aligned}$$

Thus the total number of samples is at most T in the single-elimination case.

If a phase eliminates more than one leaf, the algorithm skips some intermediate single-elimination phases. This cannot increase the budget. Indeed, if the phase index jumps from a to $b > a$, then the actual additional cost is

$$(N+1-b)(n_b - n_a),$$

whereas eliminating the same leaves one at a time would cost

$$\sum_{\ell=a+1}^{b} (N+1-\ell)(n_\ell - n_{\ell-1}) \ge (N+1-b) \sum_{\ell=a+1}^{b} (n_\ell - n_{\ell-1}) = (N+1-b)(n_b - n_a).$$

Hence subtree eliminations cannot increase the total budget, and the algorithm uses at most T samples.

One-phase concentration. By the definition of n_k,

$$n_k \ge \frac{a}{N+1-k}.$$

Therefore, by the sub-Gaussian concentration inequality,

$$\mathbb{P}\left(|\widehat{\mu}_i^{(k)} - \mu_i^{(k)}| \ge \frac{\Delta}{8} \right) \le 2\exp\left(-\frac{n_k \Delta^2}{128} \right) \le 2\exp\left(-\frac{a\Delta^2}{128(N+1-k)} \right).$$

Union bound over phases. Recall the event A_1 defined in Lemma 23. By that lemma, A_1 is sufficient for success, i.e.,

$$A_1 \subseteq \{\widehat{i}_T \in G_\varepsilon\}.$$

Therefore,

$$\{\widehat{i}_T \notin G_\varepsilon\} \subseteq \overline{A}_1.$$

It remains to bound $\mathbb{P}(\overline{A}_1)$. Using the one-phase concentration bound above and taking a union bound over leaves and phases,

$$\begin{aligned}
\mathbb{P}(\overline{A}_1) &\le \sum_{k=1}^{N-m}\sum_{i=1}^{N}\mathbb{P}\left(|\widehat{\mu}_i^{(k)} - \mu_i^{(k)}| \ge \frac{\Delta_{(N-k+1)}}{8}\right) \\
&\quad + \sum_{k=N-m+1}^{N-1}\sum_{i=1}^{N}\mathbb{P}\left(|\widehat{\mu}_i^{(k)} - \mu_i^{(k)}| \ge \frac{\Delta_{(m+1)}}{8}\right) && \text{(Union bound)} \\
&\le \sum_{k=1}^{N-m} 2N\exp\left(-\frac{a\,\Delta_{(N-k+1)}^2}{128(N-k+1)}\right) \\
&\quad + \sum_{k=N-m+1}^{N-1} 2N\exp\left(-\frac{a\,\Delta_{(m+1)}^2}{128(m+1)}\right) && \text{(Hoeffding's inequality)}
\end{aligned}$$

By the definition

$$H_2(\varepsilon) := \max_{r \ge m+1} r\Delta_{(r)}^{-2},$$

each exponent above is at most

$$-\frac{a}{128H_2(\varepsilon)}.$$

Since there are at most N phases and at most N leaves in each phase,

$$\begin{aligned}
\mathbb{P}(\overline{A}_1) &\le 2N^2\exp\left(-\frac{a}{128H_2(\varepsilon)}\right) \\
&= 2K^2L^2\exp\left(-\frac{T-KL}{128\,\overline{\log}(KL)\,H_2(\varepsilon)}\right).
\end{aligned}$$

This proves the stated error bound with complexity $H_2(\varepsilon) = \max_{r\ge m+1} r\Delta_{(r)}^{-2}$.

□

C Proof of Lemma 3

Proof. Let i be such that $i \ge m+1$. For any $j \le i$, $(\Delta_{(j)} \vee \varepsilon) \le \Delta_{(i)}$, since $\Delta_{(j)} \le \Delta_{(i)}$ and $\varepsilon \le \Delta_{(i)}$. Thus, for all $j \le i$, $(\Delta_{(j)} \vee \varepsilon)^{-2} \ge \Delta_{(i)}^{-2}$. Summing up, we have,

$$i\Delta_{(i)}^{-2} \le \sum_{j=1}^{i}(\Delta_{(j)} \vee \varepsilon)^{-2} \le \sum_{j=1}^{KL}(\Delta_{(j)} \vee \varepsilon)^{-2}$$

Since this holds for any $i \ge m+1$, we can take the maximum over $i \ge m+1$:

$$\max_{i\ge m+1} i\Delta_{(i)}^{-2} \le \sum_{j=1}^{KL}(\Delta_{(j)} \vee \varepsilon)^{-2}$$

□

Reverse inequality. For $\varepsilon = 0$, when all gaps are positive, we have

$$H_1(0) = \sum_{r=1}^{KL} \frac{1}{\Delta_{(r)}^2} \le H_2(0) \sum_{r=1}^{KL} \frac{1}{r} \le (1 + \log(KL)) H_2(0).$$

For $\varepsilon > 0$, the reverse inequality does not hold in general. Consider ordinary BAI with $K = 3$, $L = 1$, and means

$$\mu_1 = 0, \qquad \mu_2 = -\varepsilon/2, \qquad \mu_3 = -1,$$

where $0 < \varepsilon < 1$. The first two arms are ε-good, and

$$H_2(\varepsilon) = 3, \qquad H_1(\varepsilon) = \frac{2}{\varepsilon^2} + 1.$$

Thus, $H_1(\varepsilon)/H_2(\varepsilon)$ can be arbitrarily large as $\varepsilon \to 0$.

D Proof of Theorem 7

D.1 Warmup: A fixed-budget lower bound for BAI with Gaussian arms

We present a version of a fixed-budget lower bound for best arm identification with flipping construction using Gaussian arms with unit variance. The construction follows the spirit of Carpentier and Locatelli [5], while the proof below is new and self-contained.

Instance family. Fix $K \ge 2$ and parameters $(d_i)_{i=2}^K$ with $d_i > 0$. Define K instances $\{\nu_i\}_{i=1}^K$ over K arms, where each arm yields i.i.d. rewards distributed as $\mathcal{N}(\mu_k, 1)$.

Base instance ν_1:

$$\mu_1 = \tfrac{1}{2}, \qquad \mu_i = \tfrac{1}{2} - d_i \quad (i = 2, \dots, K).$$

For each $i \in \{2, \dots, K\}$, alternative instance ν_i:

$$\mu_i = \tfrac{1}{2} + d_i, \qquad \mu_1 = \tfrac{1}{2}, \qquad \mu_k = \tfrac{1}{2} - d_k \quad (k \in \{2, \dots, K\} \setminus \{i\}).$$

Thus ν_1 has unique best arm 1, whereas ν_i has unique best arm i.

As in the standard BAI setting, define the instance-dependent complexity

$$H(\nu) \;:=\; \sum_{k \neq k^\star(\nu)} \frac{1}{\Delta_k(\nu)^2}, \qquad \Delta_k(\nu) := \mu_{k^\star(\nu)} - \mu_k,$$

where $k^\star(\nu)$ is the unique best arm. Then for every $i \in [K]$,

$$H(\nu_i) \;\le\; H(\nu_1), \tag{11}$$

since for $i = 1$ the gaps are $\Delta_k(\nu_1) = d_k$ ($k \ge 2$), and for $i \ge 2$ the gaps satisfy $\Delta_1(\nu_i) = d_i$ and $\Delta_k(\nu_i) = d_i + d_k \ge d_k$ for $k \neq 1, i$.

Define the bounded-complexity class $\mathcal{P}_H := \{\nu : \; H(\nu) \le H\}$.

Theorem 25 (Fixed-budget lower bound for Gaussian BAI)**.** *Consider the instance ν_1 as described above and write $H := H(\nu_1)$. Let $\mathcal{A}$ be any (possibly adaptive) fixed-budget best-arm identification algorithm that collects exactly T samples and outputs a recommendation $J_T \in [K]$. There exists an instance $\nu \in \mathcal{P}_H$ such that*

$$\mathbb{P}_\nu\big(J_T \neq k^\star(\nu)\big) \;\ge\; \frac{1}{4} \exp\!\left(-\frac{2T}{H}\right).$$

Proof. Fix the family $\{\nu_i\}_{i=1}^K$ above. By Eq. (11), each $\nu_i \in \mathcal{P}_H$.

Let $\mathbb{P}_\nu$ denote the distribution of the full interaction transcript generated by running $\mathcal{A}$ on instance ν. Let N_k be the (random) number of pulls of arm k up to time T, and write $\mathbb{E}_1$ for expectation under ν_1.

Step 1: KL decomposition. For any $i \in \{2, \dots, K\}$, the instances ν_1 and ν_i differ only in the mean of arm i, namely $\frac{1}{2} - d_i$ versus $\frac{1}{2} + d_i$. A standard divergence decomposition (chain rule for KL under adaptive sampling) yields

$$\mathrm{KL}(\mathbb{P}_{\nu_1}, \mathbb{P}_{\nu_i}) = \mathbb{E}_1[N_i] \cdot \mathrm{KL}\big(\mathcal{N}(\tfrac{1}{2} - d_i, 1), \mathcal{N}(\tfrac{1}{2} + d_i, 1)\big). \tag{12}$$

For unit-variance Gaussians,

$$\mathrm{KL}\big(\mathcal{N}(m, 1), \mathcal{N}(m', 1)\big) = \frac{(m - m')^2}{2}.$$

Here $(m - m')^2 = (2d_i)^2 = 4d_i^2$, hence

$$\mathrm{KL}\big(\mathcal{N}(\tfrac{1}{2} - d_i, 1), \mathcal{N}(\tfrac{1}{2} + d_i, 1)\big) = 2d_i^2. \tag{13}$$

Combining (12) and (13) gives

$$\mathrm{KL}(\mathbb{P}_{\nu_1}, \mathbb{P}_{\nu_i}) = 2\,\mathbb{E}_1[N_i]\,d_i^2. \tag{14}$$

Step 2: Two-point testing inequality. We use the Bretagnolle–Huber inequality: for any distributions P, Q and any event E,

$$P(E^c) + Q(E) \;\geq\; \frac{1}{2}\exp\big(-\mathrm{KL}(P, Q)\big). \tag{15}$$

Apply (15) with $P = \mathbb{P}_{\nu_1}$, $Q = \mathbb{P}_{\nu_i}$, and $E = \{J_T = 1\}$. Since arm 1 is optimal under ν_1 and suboptimal under ν_i,

$$\mathbb{P}_{\nu_1}(E^c) = \mathbb{P}_{\nu_1}(J_T \neq 1) = \mathbb{P}_{\nu_1}(\text{error}), \qquad \mathbb{P}_{\nu_i}(E) = \mathbb{P}_{\nu_i}(J_T = 1) \leq \mathbb{P}_{\nu_i}(\text{error}).$$

Therefore,

$$\mathbb{P}_{\nu_1}(\text{error}) + \mathbb{P}_{\nu_i}(\text{error}) \;\geq\; \frac{1}{2}\exp\big(-\mathrm{KL}(\mathbb{P}_{\nu_1}, \mathbb{P}_{\nu_i})\big). \tag{16}$$

Step 3: Choose an index i with small $\mathbb{E}_1[N_i]d_i^2$. We claim there exists $i \in \{2, \dots, K\}$ such that

$$\mathbb{E}_1[N_i]\,d_i^2 \;\leq\; \frac{T}{H}. \tag{17}$$

Indeed, if (17) failed for all $i \geq 2$, then

$$\sum_{i=2}^{K} \mathbb{E}_1[N_i] \;>\; \sum_{i=2}^{K} \frac{T}{H} \cdot \frac{1}{d_i^2} \;=\; \frac{T}{H} \sum_{i=2}^{K} \frac{1}{d_i^2} \;=\; \frac{T}{H} \cdot H \;=\; T,$$

contradicting $\sum_{i=2}^{K} \mathbb{E}_1[N_i] \leq \sum_{k=1}^{K} \mathbb{E}_1[N_k] = T$.

Fix such an i. Then by (14) and (17),

$$\mathrm{KL}(\mathbb{P}_{\nu_1}, \mathbb{P}_{\nu_i}) = 2\,\mathbb{E}_1[N_i]\,d_i^2 \;\leq\; \frac{2T}{H}.$$

Plugging into (16) yields

$$\mathbb{P}_{\nu_1}(\text{error}) + \mathbb{P}_{\nu_i}(\text{error}) \;\geq\; \frac{1}{2}\exp\left(-\frac{2T}{H}\right).$$

Hence at least one of the two terms is at least half of the right-hand side:

$$\max\Big\{\mathbb{P}_{\nu_1}(\text{error}),\; \mathbb{P}_{\nu_i}(\text{error})\Big\} \;\geq\; \frac{1}{4}\exp\left(-\frac{2T}{H}\right).$$

□

D.2 Proof of Theorem 7

Proof. We first construct an instance family.

Instance family. For the reference instance ν, since its reward distributions on all leaves are Gaussian with variance 1, ν can be specified by its mean matrix $\mu = (\mu_{a,b})_{a\in[K],\, b\in[L]}$.

Let $\gamma := \mu_{1,1} - \mu_{2,1} = \Delta_{2,1} > 0$. We define a modified base instance $\bar{\nu}$ by keeping subtrees 1 and 2 unchanged and shifting every leaf in each subtree $i \geq 3$ downward by γ. Specifically, its mean matrix is

$$\bar{\mu}_{a,b} = \begin{cases} \mu_{a,b}, & a \in \{1, 2\}, \\ \mu_{a,b} - \gamma, & a \geq 3. \end{cases}$$

Let $\bar{\Delta}_{i,j}$ denote the leafwise gaps of $\bar{\nu}$. Since the within-subtree gaps are unchanged and the separation from subtree 1 only increases, $\bar{\Delta}_{i,j} \geq \Delta_{i,j}$ for every leaf. Thus $H(\bar{\nu}) \leq H(\nu)$. Moreover,

$$\bar{\Delta}_{1,j} = \Delta_{1,j}, \qquad \bar{\Delta}_{2,1} = \Delta_{2,1} = \gamma, \qquad \bar{\Delta}_{i,1} = \Delta_{i,1} + \gamma \leq 2\Delta_{i,1} \quad (i \geq 3).$$

Consequently,

$$H_{\mathrm{lb}}(\bar{\nu}) \geq \frac{1}{4} H_{\mathrm{lb}}(\nu). \tag{18}$$

Define the index set of "critical" leaves

$$I \;=\; \{(i,1) : i \neq 1\} \;\cup\; \{(1,j) : j \neq 1\}.$$

For each $(i,j) \in I$, we construct an alternative instance $\bar{\nu}^{i,j}$ whose reward distributions are still Gaussian with variance 1, but whose means differ from $\bar{\nu}$ at a subset of leaves.

For any $(i,1)$ with $i \neq 1$, define $\bar{\nu}^{i,1}$ by the mean matrix $\bar{\mu}^{i,1} = (\bar{\mu}^{i,1}_{a,b})_{a,b}$:

$$\bar{\mu}^{i,1}_{a,b} = \begin{cases} \bar{\mu}_{a,b} + 2\bar{\Delta}_{i,1}, & a = i, \\ \bar{\mu}_{a,b}, & \text{otherwise.} \end{cases}$$

Equivalently, $\bar{\nu}^{i,1}$ is identical to $\bar{\nu}$ except that leaves $\{(i,j) : j \in [L]\}$ are shifted upward by $2\bar{\Delta}_{i,1}$.

For any $(1,j)$ with $j \neq 1$, define $\bar{\nu}^{1,j}$ by

$$\bar{\mu}^{1,j}_{a,b} = \begin{cases} \bar{\mu}_{a,b} - 2\bar{\Delta}_{1,j}, & (a,b) = (1,j), \\ \bar{\mu}_{a,b}, & \text{otherwise.} \end{cases}$$

Equivalently, $\bar{\nu}^{1,j}$ is identical to $\bar{\nu}$ except that leaf $(1,j)$ is shifted downward by $2\bar{\Delta}_{1,j}$.

Collect these alternatives and the modified base instance as

$$\mathcal{V}(\bar{\nu}) := \{\bar{\nu}^{i,j} : (i,j) \in I\} \cup \{\bar{\nu}\}.$$

It is easy to see that the optimal subtree in any $\bar{\nu}^{i,j}$ is not subtree 1. Specifically, for any $(i,1)$ with $i \neq 1$, the optimal subtree in $\bar{\nu}^{i,1}$ is subtree i, and the second optimal subtree is subtree 1; for any $(1,j)$ with $j \neq 1$, the optimal subtree in $\bar{\nu}^{1,j}$ is subtree 2.

We verify that $\mathcal{V}(\bar{\nu}) \subseteq \mathcal{P}_{4H(\nu)}$.

For $\bar{\nu}^{i,1}$ with $i \geq 2$, subtree i becomes optimal and subtree 1 becomes second best. The new gaps of leaves in subtree i are

$$\bar{\Delta}_{i,1} + \mu_{i,b} - \mu_{i,1} \geq \Delta_{i,b}.$$

For leaves in subtrees $k \notin \{1, i\}$, the new gaps are

$$\max\{\bar{\Delta}_{i,1} + \bar{\Delta}_{k,1}, \mu_{k,b} - \mu_{k,1}\} \geq \Delta_{k,b}.$$

For leaves in subtree 1, the new gaps satisfy

$$\max\{\bar{\Delta}_{i,1}, \mu_{1,b} - \mu_{1,1}\} \geq \frac{1}{2}(\gamma + \mu_{1,b} - \mu_{1,1}) = \frac{1}{2}\Delta_{1,b}.$$

Thus every new gap is at least half its corresponding gap in ν, and $H(\bar{\nu}^{i,1}) \leq 4H(\nu)$.

For $\bar{\nu}^{1,j}$ with $j \geq 2$, subtree 2 becomes optimal, and every other subtree has value at most $\mu_{1,1} - 2\gamma$. In particular, for every $k \geq 3$, its new gap to the optimal subtree is

$$\bar{\mu}_{2,1} - \bar{\mu}_{k,1} = \mu_{1,1} - \mu_{k,1} = \Delta_{k,1}.$$

Since its within-subtree gaps are unchanged, all its leafwise gaps equal those in ν. For subtree 2, the new leafwise gaps are at least

$$\gamma + \mu_{2,b} - \mu_{2,1} \geq \Delta_{2,b}.$$

For subtree 1, the modified leaf $(1, j)$ has gap $\Delta_{1,j}$, while every other leaf $(1, b)$ has gap

$$2\gamma + (\mu_{1,j} - \mu_{1,1}) + (\mu_{1,b} - \mu_{1,1}) \geq \Delta_{1,b}.$$

Therefore $H(\bar{\nu}^{1,j}) \leq H(\nu)$. Together with $H(\bar{\nu}) \leq H(\nu)$, this proves $\mathcal{V}(\bar{\nu}) \subseteq \mathcal{P}_{4H(\nu)}$.

For any ν', let $\mathbb{P}_{\nu'}$ denote the distribution of the full interaction transcript generated by running π on instance ν'. Let $N_{i,j}$ be the (random) number of pulls of leaf (i, j) up to time T, and write $\mathbb{E}_0$ for expectation under $\mathbb{P}_{\bar{\nu}}$ (here $\bar{\nu}$ is the common base instance).

Step 1: KL decomposition. A standard divergence decomposition (chain rule for KL under adaptive sampling) yields, for any $\nu' \in \mathcal{V}(\bar{\nu})$,

$$\mathrm{KL}(\mathbb{P}_{\bar{\nu}}, \mathbb{P}_{\nu'}) = \sum_{i,j} \mathbb{E}_0[N_{i,j}] \cdot \mathrm{KL}\Big(\mathcal{N}(\bar{\mu}_{i,j}, 1), \mathcal{N}(\mu'_{i,j}, 1)\Big).$$

For unit-variance Gaussians,

$$\mathrm{KL}\big(\mathcal{N}(m, 1), \mathcal{N}(m', 1)\big) = \frac{(m - m')^2}{2}.$$

For any $i \in \{2, \ldots, K\}$, the instances $\bar{\nu}^{i,1}$ and $\bar{\nu}$ differ only in the means of leaves in subtree i. Hence,

$$\begin{aligned}\mathrm{KL}(\mathbb{P}_{\bar{\nu}}, \mathbb{P}_{\bar{\nu}^{i,1}}) &= \sum_j \mathbb{E}_0[N_{i,j}] \cdot \mathrm{KL}\Big(\mathcal{N}(\bar{\mu}_{i,j}, 1), \mathcal{N}(\bar{\mu}^{i,1}_{i,j}, 1)\Big) \\ &= \sum_j \mathbb{E}_0[N_{i,j}] \cdot (2\bar{\Delta}^2_{i,1}). \end{aligned} \tag{19}$$

For any $j \in \{2, \ldots, L\}$, the instances $\bar{\nu}^{1,j}$ and $\bar{\nu}$ differ only in the mean of leaf $(1, j)$. Hence,

$$\begin{aligned}\mathrm{KL}(\mathbb{P}_{\bar{\nu}}, \mathbb{P}_{\bar{\nu}^{1,j}}) &= \mathbb{E}_0[N_{1,j}] \cdot \mathrm{KL}\Big(\mathcal{N}(\bar{\mu}_{1,j}, 1), \mathcal{N}(\bar{\mu}^{1,j}_{1,j}, 1)\Big) \\ &= \mathbb{E}_0[N_{1,j}] \cdot (2\bar{\Delta}^2_{1,j}). \end{aligned} \tag{20}$$

Step 2: Two-point testing inequality. We use the Bretagnolle–Huber inequality: for any distributions P, Q and any event E,

$$P(E^c) + Q(E) \;\geq\; \frac{1}{2} \exp\big(-\mathrm{KL}(P, Q)\big). \tag{21}$$

Let $(i, j) \in I$ and apply (21) with $P = \mathbb{P}_{\bar{\nu}}$, $Q = \mathbb{P}_{\bar{\nu}^{i,j}}$, and $E = \{\text{output subtree } 1\}$. Since subtree 1 is optimal under $\bar{\nu}$ and suboptimal under $\bar{\nu}^{i,j}$,

$$\mathbb{P}_{\bar{\nu}}(E^c) = \mathbb{P}_{\bar{\nu}}(\text{error}), \qquad \mathbb{P}_{\bar{\nu}^{i,j}}(E) \leq \mathbb{P}_{\bar{\nu}^{i,j}}(\text{error}).$$

Thus,

$$\mathbb{P}_{\bar{\nu}}(\text{error}) + \mathbb{P}_{\bar{\nu}^{i,j}}(\text{error}) \;\geq\; \frac{1}{2} \exp\big(-\mathrm{KL}(\mathbb{P}_{\bar{\nu}}, \mathbb{P}_{\bar{\nu}^{i,j}})\big),$$

and

$$\begin{aligned}&\max\Big\{\mathbb{P}_{\bar{\nu}}(\text{error}), \mathbb{P}_{\bar{\nu}^{i,j}}(\text{error})\Big\} \\ &\geq \frac{1}{2}\Big(\mathbb{P}_{\bar{\nu}}(\text{error}) + \mathbb{P}_{\bar{\nu}^{i,j}}(\text{error})\Big) \\ &\geq \frac{1}{4} \exp\big(-\mathrm{KL}(\mathbb{P}_{\bar{\nu}}, \mathbb{P}_{\bar{\nu}^{i,j}})\big). \end{aligned} \tag{22}$$

Step 3: Proof by contradiction. We assume, for contradiction, that the algorithm has a small error on all instances:

$$\forall \nu' \in \mathcal{P}_{4H(\nu)}, \quad \mathbb{P}_{\nu'}(\text{error}) < \frac{1}{4}\exp\left(-2\frac{T}{H_{\text{lb}}(\bar{\nu})}\right).$$

Since $\bar{\nu}$ and all its alternatives belong to $\mathcal{P}_{4H(\nu)}$, this implies that for all $(i,j) \in I$,

$$\max\left\{\mathbb{P}_{\bar{\nu}}(\text{error}), \mathbb{P}_{\bar{\nu}^{i,j}}(\text{error})\right\} < \frac{1}{4}\exp\left(-2\frac{T}{H_{\text{lb}}(\bar{\nu})}\right).$$

In conjunction with Equation (22), we have, for all $(i,j) \in I$,

$$\text{KL}(\mathbb{P}_{\bar{\nu}}, \mathbb{P}_{\bar{\nu}^{i,j}}) > \frac{2T}{H_{\text{lb}}(\bar{\nu})}.$$

By Equations (19) and (20), that is, for all $(i,1)$ with $i \neq 1$,

$$\sum_{j\in[L]} \mathbb{E}_0[N_{i,j}]\bar{\Delta}_{i,1}^2 > \frac{T}{H_{\text{lb}}(\bar{\nu})}$$

and also, for all $(1,j)$ with $j \neq 1$,

$$\mathbb{E}_0[N_{1,j}]\bar{\Delta}_{1,j}^2 > \frac{T}{H_{\text{lb}}(\bar{\nu})}.$$

However,

$$\begin{aligned}
T &\geq \sum_{i\neq 1}\sum_{j\in[L]} \mathbb{E}_0[N_{i,j}] + \sum_{j\neq 1}\mathbb{E}_0[N_{1,j}] \\
&> \sum_{i\neq 1}\frac{T}{H_{\text{lb}}(\bar{\nu})}\frac{1}{\bar{\Delta}_{i,1}^2} + \sum_{j\neq 1}\frac{T}{H_{\text{lb}}(\bar{\nu})}\frac{1}{\bar{\Delta}_{1,j}^2} \\
&= \frac{T}{H_{\text{lb}}(\bar{\nu})}\left(\sum_{i\neq 1}\frac{1}{\bar{\Delta}_{i,1}^2} + \sum_{j\neq 1}\frac{1}{\bar{\Delta}_{1,j}^2}\right) \\
&= T,
\end{aligned}$$

leading to a contradiction. Therefore, it cannot be true that

$$\forall \nu' \in \mathcal{P}_{4H(\nu)}, \quad \mathbb{P}_{\nu'}(\text{error}) < \frac{1}{4}\exp\left(-2\frac{T}{H_{\text{lb}}(\bar{\nu})}\right).$$

Hence there exists $\nu' \in \mathcal{P}_{4H(\nu)}$ such that

$$\mathbb{P}_{\nu'}(\text{error}) \geq \frac{1}{4}\exp\left(-2\frac{T}{H_{\text{lb}}(\bar{\nu})}\right) \geq \frac{1}{4}\exp\left(-8\frac{T}{H_{\text{lb}}(\nu)}\right),$$

where the last inequality follows from Equation (18). This completes the proof. □

D.3 Proof of Theorem 8

Unlike exact identification, changing the optimal subtree does not necessarily make subtree 1 ε-bad. The base and alternative instances may have overlapping sets of ε-good subtrees, in which case a learner can return a common acceptable subtree and be correct on both instances. Thus, directly replacing each gap Δ by $\Delta \vee \varepsilon$ in the exact-identification proof is insufficient. We instead construct a modified base instance whose only ε-good subtree is subtree 1, together with alternatives on which subtree 1 is ε-bad.

For $\mathcal{H} > 0$, define

$$\mathcal{P}_{\mathcal{H}}^{\varepsilon} := \{\nu' : H_1(\varepsilon; \nu') \leq \mathcal{H}\},$$

where the instances are (K, L) maximin trees with unit-variance Gaussian rewards and a unique optimal subtree.

Proof of Theorem 8. We first construct an instance family.

Instance family. Let $\mu = (\mu_{a,b})_{a\in[K],\, b\in[L]}$ be the mean matrix of the reference instance ν, indexed according to (2). Set

$$\gamma := \mu_{1,1} - \mu_{2,1} = \Delta_{2,1} > 0, \qquad \beta := \gamma \vee 2\varepsilon.$$

We define a modified base instance $\bar{\nu}$ by keeping subtree 1 unchanged, shifting every leaf of subtree 2 downward by $\beta - \gamma$, and shifting every leaf in each subtree $i \geq 3$ downward by β. Specifically, its mean matrix is

$$\bar{\mu}_{a,b} = \begin{cases} \mu_{a,b}, & a = 1, \\ \mu_{a,b} - (\beta - \gamma), & a = 2, \\ \mu_{a,b} - \beta, & a \geq 3. \end{cases}$$

Subtree 2 remains second best, with value $\bar{\mu}_{2,1} = \mu_{1,1} - \beta$. Since $\beta \geq 2\varepsilon > \varepsilon$, subtree 1 is the only ε-good subtree of $\bar{\nu}$.

Let $\bar{\Delta}_{i,j}$ denote the leafwise gaps of $\bar{\nu}$. The within-subtree gaps are unchanged, and the separation from subtree 1 only increases. Hence $\bar{\Delta}_{i,j} \geq \Delta_{i,j}$ for every leaf, so

$$H_1(\varepsilon; \bar{\nu}) \leq H_1(\varepsilon; \nu).$$

Moreover,

$$\begin{aligned} \bar{\Delta}_{1,j} &= \Delta_{1,j} + \beta - \gamma \leq 3(\Delta_{1,j} \vee \varepsilon), \\ \bar{\Delta}_{2,1} &= \beta \leq 2(\Delta_{2,1} \vee \varepsilon), \\ \bar{\Delta}_{i,1} &= \Delta_{i,1} + \beta \leq 3(\Delta_{i,1} \vee \varepsilon) \qquad (i \geq 3). \end{aligned}$$

Here we used $\Delta_{1,j} \geq \gamma$ and $\Delta_{i,1} \geq \gamma$ for $i \geq 2$. All these modified gaps are at least $\beta \geq 2\varepsilon$. Consequently,

$$H_{\mathrm{lb}}(\bar{\nu}, \varepsilon) = \sum_{i=2}^{K} \frac{1}{\bar{\Delta}_{i,1}^2} + \sum_{j=2}^{L} \frac{1}{\bar{\Delta}_{1,j}^2} \geq \frac{1}{9} H_{\mathrm{lb}}(\nu, \varepsilon). \tag{23}$$

Define the index set of "critical" leaves

$$I = \{(i,1) : i \in \{2, \ldots, K\}\} \cup \{(1,j) : j \in \{2, \ldots, L\}\}.$$

For each $(i,j) \in I$, we construct an alternative instance $\bar{\nu}^{i,j}$ whose reward distributions remain Gaussian with variance 1. For any $(i,1)$ with $i \geq 2$, define

$$\bar{\mu}_{a,b}^{i,1} = \begin{cases} \bar{\mu}_{a,b} + 2\bar{\Delta}_{i,1}, & a = i, \\ \bar{\mu}_{a,b}, & \text{otherwise}. \end{cases}$$

Equivalently, $\bar{\nu}^{i,1}$ is identical to $\bar{\nu}$ except that leaves $\{(i,j) : j \in [L]\}$ are shifted upward by $2\bar{\Delta}_{i,1}$. Subtree i becomes uniquely optimal, with value $\mu_{1,1} + \bar{\Delta}_{i,1}$, while subtree 1 is second best with value $\mu_{1,1}$. Since $\bar{\Delta}_{i,1} \geq \beta > \varepsilon$, subtree 1 is ε-bad on this alternative.

For any $(1,j)$ with $j \geq 2$, define

$$\bar{\mu}_{a,b}^{1,j} = \begin{cases} \bar{\mu}_{a,b} - 2\bar{\Delta}_{1,j}, & (a,b) = (1,j), \\ \bar{\mu}_{a,b}, & \text{otherwise}. \end{cases}$$

Equivalently, $\bar{\nu}^{1,j}$ is identical to $\bar{\nu}$ except that leaf $(1,j)$ is shifted downward by $2\bar{\Delta}_{1,j}$. This leaf becomes the minimum of subtree 1, with mean

$$\bar{\mu}_{1,j}^{1,j} = \mu_{1,1} - 2\beta - (\mu_{1,j} - \mu_{1,1}).$$

Subtree 2 becomes uniquely optimal, and its advantage over subtree 1 is

$$\bar{\mu}_{2,1} - \bar{\mu}_{1,j}^{1,j} = \beta + \mu_{1,j} - \mu_{1,1} = \bar{\Delta}_{1,j} \geq \beta > \varepsilon.$$

Thus subtree 1 is also ε-bad on this alternative.

Collect these alternatives and the modified base instance as

$$\mathcal{V}(\bar{\nu}) := \{\bar{\nu}\} \cup \{\bar{\nu}^{i,j} : (i,j) \in I\}.$$

We have shown that

$$\mathcal{G}_\varepsilon(\bar{\nu}) = \{1\}, \qquad 1 \notin \mathcal{G}_\varepsilon(\bar{\nu}^{i,j}) \quad \text{for every } (i,j) \in I. \tag{24}$$

We next verify that $\mathcal{V}(\bar{\nu}) \subseteq \mathcal{P}^{\varepsilon}_{4H_1(\varepsilon;\nu)}$. Throughout these comparisons, we keep the original leaf indices and compute each new gap using the optimal subtree and subtree minima of the alternative instance.

For $\bar{\nu}^{i,1}$ with $i \geq 2$, subtree i becomes optimal and subtree 1 becomes second best. The new gaps of leaves in subtree i are

$$\bar{\Delta}_{i,1} + \mu_{i,b} - \mu_{i,1} \geq \Delta_{i,b}.$$

For leaves in subtrees $k \notin \{1, i\}$, the new gaps are

$$\max\{\bar{\Delta}_{i,1} + \bar{\Delta}_{k,1}, \mu_{k,b} - \mu_{k,1}\} \geq \Delta_{k,b}.$$

For leaves in subtree 1, the new gaps satisfy

$$\max\{\bar{\Delta}_{i,1}, \mu_{1,b} - \mu_{1,1}\} \geq \frac{1}{2}(\gamma + \mu_{1,b} - \mu_{1,1}) = \frac{1}{2}\Delta_{1,b}.$$

Thus every new gap is at least half its corresponding gap in ν. Taking the maximum with ε preserves this comparison, and therefore

$$H_1(\varepsilon; \bar{\nu}^{i,1}) \leq 4H_1(\varepsilon; \nu).$$

For $\bar{\nu}^{1,j}$ with $j \geq 2$, subtree 2 becomes optimal. Every other subtree has value at most $\mu_{1,1} - \beta - \gamma$: for subtree 1 this follows from $\beta \geq \gamma$, and for $k \geq 3$ it follows from $\mu_{k,1} \leq \mu_{2,1} = \mu_{1,1} - \gamma$. For every $k \geq 3$, its new gap to the optimal subtree is

$$\bar{\mu}_{2,1} - \bar{\mu}_{k,1} = \mu_{1,1} - \mu_{k,1} = \Delta_{k,1}.$$

Since its within-subtree gaps are unchanged, all its leafwise gaps equal those in ν. For subtree 2, the new leafwise gaps are at least

$$\gamma + \mu_{2,b} - \mu_{2,1} \geq \Delta_{2,b}.$$

For subtree 1, the modified leaf $(1, j)$ has gap

$$\bar{\Delta}_{1,j} = \beta + \mu_{1,j} - \mu_{1,1} \geq \Delta_{1,j},$$

while every other leaf $(1, b)$ has gap

$$2\beta + (\mu_{1,j} - \mu_{1,1}) + (\mu_{1,b} - \mu_{1,1}) \geq \Delta_{1,b}.$$

Therefore $H_1(\varepsilon; \bar{\nu}^{1,j}) \leq H_1(\varepsilon; \nu)$. Together with $H_1(\varepsilon; \bar{\nu}) \leq H_1(\varepsilon; \nu)$, this proves the claimed inclusion.

For any instance ν', let $\mathbb{P}_{\nu'}$ denote the distribution of the full interaction transcript, including the recommendation, generated by running π on ν'. Let $N_{i,j}$ be the number of pulls of leaf (i, j), and write $\mathbb{E}_0$ for expectation under $\mathbb{P}_{\bar{\nu}}$. Throughout the rest of the proof, $\bar{\nu}$ is the common base instance, and $\mathrm{error}_\varepsilon$ denotes recommending a subtree that is not ε-good on the instance in question.

Step 1: KL decomposition. The chain rule for KL under adaptive sampling gives

$$\mathrm{KL}(\mathbb{P}_{\bar{\nu}}, \mathbb{P}_{\nu'}) = \sum_{a=1}^{K} \sum_{b=1}^{L} \mathbb{E}_0[N_{a,b}]\, \mathrm{KL}\Big(\mathcal{N}(\bar{\mu}_{a,b}, 1), \mathcal{N}(\mu'_{a,b}, 1)\Big).$$

For unit-variance Gaussians,

$$\mathrm{KL}\big(\mathcal{N}(m, 1), \mathcal{N}(m', 1)\big) = \frac{(m - m')^2}{2}.$$

For $i \geq 2$, the instances $\bar{\nu}$ and $\bar{\nu}^{i,1}$ differ only in the means of the leaves in subtree i. Hence

$$\mathrm{KL}(\mathbb{P}_{\bar{\nu}}, \mathbb{P}_{\bar{\nu}^{i,1}}) = 2\bar{\Delta}^2_{i,1} \sum_{j=1}^{L} \mathbb{E}_0[N_{i,j}]. \tag{25}$$

For $j \geq 2$, the instances $\bar{\nu}$ and $\bar{\nu}^{1,j}$ differ only in the mean of leaf $(1, j)$. Hence

$$\mathrm{KL}(\mathbb{P}_{\bar{\nu}}, \mathbb{P}_{\bar{\nu}^{1,j}}) = 2\bar{\Delta}^2_{1,j} \mathbb{E}_0[N_{1,j}]. \tag{26}$$

Step 2: Two-point testing inequality. Let $(i,j) \in I$. Apply the Bretagnolle–Huber inequality with $P = \mathbb{P}_{\bar{\nu}}$, $Q = \mathbb{P}_{\bar{\nu}^{i,j}}$, and $E = \{\widehat{i}_T = 1\}$:

$$\mathbb{P}_{\bar{\nu}}(E^c) + \mathbb{P}_{\bar{\nu}^{i,j}}(E) \geq \frac{1}{2}\exp\bigl(-\mathrm{KL}(\mathbb{P}_{\bar{\nu}}, \mathbb{P}_{\bar{\nu}^{i,j}})\bigr).$$

By (24),

$$\mathbb{P}_{\bar{\nu}}(E^c) = \mathbb{P}_{\bar{\nu}}(\mathrm{error}_\varepsilon), \qquad \mathbb{P}_{\bar{\nu}^{i,j}}(E) \leq \mathbb{P}_{\bar{\nu}^{i,j}}(\mathrm{error}_\varepsilon).$$

Consequently,

$$\begin{aligned} &\max\bigl\{\mathbb{P}_{\bar{\nu}}(\mathrm{error}_\varepsilon), \mathbb{P}_{\bar{\nu}^{i,j}}(\mathrm{error}_\varepsilon)\bigr\} \\ &\quad \geq \frac{1}{4}\exp\bigl(-\mathrm{KL}(\mathbb{P}_{\bar{\nu}}, \mathbb{P}_{\bar{\nu}^{i,j}})\bigr). \end{aligned} \tag{27}$$

Step 3: Proof by contradiction. Suppose, for contradiction, that

$$\forall \nu' \in \mathcal{P}^{\varepsilon}_{4H_1(\varepsilon;\nu)}, \qquad \mathbb{P}_{\nu'}(\mathrm{error}_\varepsilon) < \frac{1}{4}\exp\left(-\frac{2T}{H_{\mathrm{lb}}(\bar{\nu},\varepsilon)}\right).$$

Since $\bar{\nu}$ and all its alternatives belong to this class, Equation (27) implies that, for every $(i,j) \in I$,

$$\mathrm{KL}(\mathbb{P}_{\bar{\nu}}, \mathbb{P}_{\bar{\nu}^{i,j}}) > \frac{2T}{H_{\mathrm{lb}}(\bar{\nu},\varepsilon)}.$$

By Equations (25) and (26),

$$\sum_{j=1}^{L} \mathbb{E}_0[N_{i,j}]\bar{\Delta}_{i,1}^2 > \frac{T}{H_{\mathrm{lb}}(\bar{\nu},\varepsilon)} \qquad (i \geq 2),$$

and

$$\mathbb{E}_0[N_{1,j}]\bar{\Delta}_{1,j}^2 > \frac{T}{H_{\mathrm{lb}}(\bar{\nu},\varepsilon)} \qquad (j \geq 2).$$

However, the sampling budget gives

$$\begin{aligned} T &\geq \sum_{i=2}^{K}\sum_{j=1}^{L} \mathbb{E}_0[N_{i,j}] + \sum_{j=2}^{L} \mathbb{E}_0[N_{1,j}] \\ &> \frac{T}{H_{\mathrm{lb}}(\bar{\nu},\varepsilon)}\left(\sum_{i=2}^{K}\frac{1}{\bar{\Delta}_{i,1}^2} + \sum_{j=2}^{L}\frac{1}{\bar{\Delta}_{1,j}^2}\right) \\ &= T, \end{aligned}$$

where the last equality uses the fact that all critical gaps of $\bar{\nu}$ exceed ε. This is a contradiction. Hence there exists $\nu' \in \mathcal{P}^{\varepsilon}_{4H_1(\varepsilon;\nu)}$ such that

$$\mathbb{P}_{\nu'}(\mathrm{error}_\varepsilon) \geq \frac{1}{4}\exp\left(-\frac{2T}{H_{\mathrm{lb}}(\bar{\nu},\varepsilon)}\right) \geq \frac{1}{4}\exp\left(-\frac{18T}{H_{\mathrm{lb}}(\nu,\varepsilon)}\right),$$

where the last inequality follows from (23). This completes the proof. □

E A negative result for permutation-style lower bounds

The lower bound in Section 4 is governed by $H_{\mathrm{lb}}(\nu)$, which shows a sparsity pattern compared to the upper-bound complexity $H_2(0)$. A natural question is whether one can prove a stronger H_2-type lower bound by adapting the permutation construction used in fixed-budget best-arm identification. This appendix shows that a direct permutation-style statement does not hold in general for max–min trees, even in the smallest nontrivial $(2,2)$ case.

The obstruction is that, once the grouping of leaves into subtrees is known, permutation uncertainty alone does not capture the full statistical difficulty of max–min identification measured by $H_2(0)$. We specify below the precise quantifiers of the statement being refuted.

Lower bounds in fixed-budget best-arm identification. Fixed-budget best-arm identification (BAI) can be viewed as a multi-hypothesis testing problem: each bandit instance $\nu = (\nu_1, \ldots, \nu_K)$ induces a unique optimal arm $i^\star(\nu) \in [K]$, and an algorithm π adaptively collects T samples and outputs an estimate $\hat{i} \in [K]$. The misidentification event is $\mathsf{err}(\pi, \nu) := \{\hat{i} \neq i^\star(\nu)\}$ with probability $p_T(\pi, \nu) := \mathbb{P}_\nu^\pi(\mathsf{err})$. A central consequence of this testing viewpoint is that meaningful lower bounds cannot be pointwise in the sense of bounding $p_T(\pi, \nu)$ from below for a fixed instance ν uniformly over algorithms: such a statement is impossible, since the degenerate algorithm that always outputs $i^\star(\nu)$ achieves $p_T(\pi, \nu) = 0$ on that particular instance. Therefore, fixed-budget lower bounds are formulated as local minimax guarantees over alternative instances that induce different optimal arms, and take an adversarial quantifier structure of the form

$$\forall \pi,\ \exists \nu' \text{ “close” to } \nu,\ \text{s.t. } p_T(\pi, \nu') \text{ is large.}$$

where “closeness” can take different definitions, depending on the specific theorem statements and constructions.

Such fixed-budget BAI lower bounds certify that no algorithm can be simultaneously accurate on ν and on a suitably chosen alternative ν' that induces a different correct decision, ruling out “lucky guess” strategies.

Following this spirit, one of the earliest results on fixed-budget BAI lower bounds presented in Audibert et al. [1] adopts a permutation construction, and thus an alternative instance ν' being close to ν means $\nu'_i = \nu_{\sigma(i)}$, for all $i \in [K]$, where $\sigma : [K] \to [K]$ is a permutation. The lower bound (Theorem 4) in Audibert et al. [1] can be interpreted as follows: for any instance ν, and any algorithm π, even if the knowledge (just except for the indices) of the instance is available to the algorithm, on at least one of the close instance ν' (a permutation of ν), π has a misidentification probability at least $\exp(-\tilde{\Theta}(T/H_2(\nu)))$.

One natural question we asked was, can a similar lower bound be shown for max-min subtree identification (with $\varepsilon = 0$), i.e., *is it possible to show that, even if the knowledge (just except for the indices) of the instance ν is available to the algorithm, any algorithm still has a high probability of error of* $\exp(-\tilde{\Theta}(T/H_2(\nu)))$*?*

Subsequently, we show that, perhaps surprisingly, the above-described lower bound for max-min trees cannot be shown. This indicates that the learning difficulty in max-min identification is not merely a consequence of “label uncertainty” (as in unstructured K-armed BAI), but is instead deeply rooted in the internal maximin structure. In particular, unlike the unstructured setting where all arms play symmetric roles under permutations, max-min trees inherently break this symmetry, and not every leaf contributes “equally” to the max-min identification difficulty.

Subtree-preserving permutations. Fix $K, L \geq 1$. A subtree-preserving permutation is a tuple $\sigma = \big(\sigma_0, \sigma_1, \ldots, \sigma_K\big)$, where $\sigma_0 : [K] \to [K]$ is a permutation of subtrees and, for each $i \in [K]$, $\sigma_i : [L] \to [L]$ is a permutation of the leaves within subtree i. Let Σ_{tree} denote the set of all such subtree-preserving permutations. Given an instance ν with reward distributions $\{(i, j, \nu_{i,j})\}_{i\in[K],\, j\in[L]}$, we define the subtree-preserving permuted instance ν^σ to be the instance with reward distributions

$$\{(i, j, \nu_{\sigma_0(i),\sigma_i(j)})\}_{i\in[K],\, j\in[L]} \tag{28}$$

Equivalently, ν^σ is obtained by first inside each subtree i, permutating its leaves by σ_i, and then, permuting subtrees by σ_0.

The candidate permutation-style lower bound. A direct analogue of the permutation lower bound for BAI in Audibert et al. [1] would assert that for any $K, L \geq 2$ and budget $T \geq KL$, there exist universal constants $c_0, C_0 > 0$ such that

$$\forall \nu,\ \forall \pi,\ \exists \sigma \in \Sigma_{\mathrm{tree}} :\quad p_T(\pi, \nu^\sigma) \geq c_0 \exp\!\left(-\frac{C_0 T}{H_2(0; \nu)}\right). \tag{29}$$

Here ν ranges over all (K, L) max–min tree instances with unit-variance Gaussian rewards and a unique optimal subtree, and $H_2(0; \nu)$ denotes the complexity $H_2(0)$ of the base instance. In particular, the quantifier order allows the algorithm π to know ν (including its grouping), with only the permutation σ unknown.

A $(K = 2, L = 2)$ counterexample to the candidate lower bound. The following proposition shows that (29) is false by constructing a procedure whose error probability decays faster than the proposed lower bound for every subtree-preserving permutation of a particular base instance.

Proposition 26 (Known grouping can make $(2, 2)$ maximin identification easier)**.** *Fix $\rho \in (0, 1)$ and consider the base $(2, 2)$ Gaussian maximin-tree instance $\bar{\nu}_\rho$ with unit variances and grouped leaf means*

$$\big\{\{\bar{\mu}_{1,1}, \bar{\mu}_{1,2}\}, \{\bar{\mu}_{2,1}, \bar{\mu}_{2,2}\}\big\} = \big\{\{1, \rho\}, \{\rho^2, 0\}\big\}.$$

Let $\mathcal{I}_\rho^{\mathrm{grp}}$ be the class of instances obtained from $\bar{\nu}_\rho$ by an arbitrary subtree-preserving permutation, i.e., by possibly swapping the two subtrees and by possibly permuting the two leaves inside each subtree. Suppose that the algorithm is given this grouped base instance $\bar{\nu}_\rho$, but not the realized permutation.

Then there exists a single fixed-budget identification procedure π such that, uniformly over all $\nu \in \mathcal{I}_\rho^{\mathrm{grp}}$,

$$\mathbb{P}_\nu\Big(\widehat{i}_T \neq i^\star(\nu)\Big) \leq C \exp\left(-c \frac{T}{H_1^{\mathrm{flatten}}(\rho)}\right),$$

where $c, C > 0$ are universal constants and

$$H_1^{\mathrm{flatten}}(\rho) := (1 - \rho)^{-2} + (1 - \rho^2)^{-2} + 1.$$

Moreover, let $H_1^{\mathrm{tree}}(\rho)$ denote the tree complexity $H_1(0)$ from Definition 1, evaluated on this grouped instance family. Then

$$H_1^{\mathrm{flatten}}(\rho) = o\big(H_1^{\mathrm{tree}}(\rho)\big) \qquad \text{as } \rho \to 0^+.$$

Proof. For every instance $\nu \in \mathcal{I}_\rho^{\mathrm{grp}}$, one subtree has leaf means $\{1, \rho\}$ and the other subtree has leaf means $\{\rho^2, 0\}$. Therefore the subtree containing the leaf with mean 1 is the unique optimal subtree, since its min-value is ρ, whereas the other subtree has min-value 0.

Consider the following fixed-budget procedure. Ignore the tree structure during sampling and run a standard fixed-budget best-arm identification procedure on the four leaves, treating them as four arms. After the budget is exhausted, let $\widehat{\ell}_T$ be the recommended best leaf, and output the subtree containing $\widehat{\ell}_T$. This is a single algorithm; it does not depend on which subtree-preserving permutation was realized.

If the procedure correctly identifies the unique leaf with mean 1, then it outputs the optimal subtree. Hence

$$\Big\{\widehat{i}_T \neq i^\star(\nu)\Big\} \subseteq \Big\{\widehat{\ell}_T \neq \ell^\star(\nu)\Big\},$$

where $\ell^\star(\nu)$ denotes the unique leaf with mean 1. The ordinary four-arm best-arm identification problem has gaps

$$1 - \rho, \qquad 1 - \rho^2, \qquad 1.$$

Thus, by the standard fixed-budget best-arm identification guarantee [1], there exist universal constants $c, C > 0$ such that

$$\mathbb{P}_\nu\Big(\widehat{\ell}_T \neq \ell^\star(\nu)\Big) \leq C \exp\left(-c \frac{T}{H_1^{\mathrm{flatten}}(\rho)}\right),$$

where

$$H_1^{\mathrm{flatten}}(\rho) = (1 - \rho)^{-2} + (1 - \rho^2)^{-2} + 1.$$

This proves the claimed upper bound.

It remains to compare this flattened complexity with the maximin-tree complexity. In the notation of Definition 1, after ordering leaves within subtrees and ordering subtrees by min-value, the same instance has

$$\mu_{1,1} = \rho, \qquad \mu_{1,2} = 1, \qquad \mu_{2,1} = 0, \qquad \mu_{2,2} = \rho^2.$$

Therefore the leafwise maximin gaps are

$$\Delta_{1,1} = \mu_{1,1} - \mu_{2,1} = \rho, \qquad \Delta_{1,2} = \mu_{1,2} - \mu_{2,1} = 1,$$

and

$$\Delta_{2,1} = \max\{\mu_{1,1} - \mu_{2,1},\, \mu_{2,1} - \mu_{2,1}\} = \rho,$$

$$\Delta_{2,2} = \max\{\mu_{1,1} - \mu_{2,1},\, \mu_{2,2} - \mu_{2,1}\} = \max\{\rho, \rho^2\} = \rho.$$

Consequently,

$$H_1^{\mathrm{tree}}(\rho) = \sum_{i,j} \Delta_{i,j}^{-2} = \rho^{-2} + 1 + \rho^{-2} + \rho^{-2} = 1 + 3\rho^{-2}.$$

On the other hand,

$$H_1^{\mathrm{flatten}}(\rho) = (1-\rho)^{-2} + (1-\rho^2)^{-2} + 1 \longrightarrow 3 \qquad \text{as } \rho \to 0^+.$$

Hence

$$\frac{H_1^{\mathrm{flatten}}(\rho)}{H_1^{\mathrm{tree}}(\rho)} \longrightarrow 0 \qquad \text{as } \rho \to 0^+,$$

which proves $H_1^{\mathrm{flatten}}(\rho) = o(H_1^{\mathrm{tree}}(\rho))$. □

Implication and scope. For this family, the sorted maximin gaps are $\rho, \rho, \rho, 1$, so the H_2 complexity of the base instance is

$$H_2(0; \bar{\nu}_\rho) = \max\{3\rho^{-2}, 4\} = \Theta(\rho^{-2}) \qquad \text{as } \rho \to 0^+.$$

By Proposition 26, $H_1^{\mathrm{flatten}}(\rho) \to 3$ while $H_2(0; \bar{\nu}_\rho) \to \infty$. For any proposed universal constants $c_0, C_0 > 0$ in (29), choose ρ sufficiently small that

$$\frac{c}{H_1^{\mathrm{flatten}}(\rho)} > \frac{C_0}{H_2(0; \bar{\nu}_\rho)},$$

where $c > 0$ is the constant in Proposition 26. Then, for sufficiently large T, the procedure in that proposition satisfies

$$\forall \sigma \in \Sigma_{\mathrm{tree}}, \qquad p_T(\pi, \bar{\nu}_\rho^\sigma) < c_0 \exp\left(-\frac{C_0 T}{H_2(0; \bar{\nu}_\rho)}\right).$$

This contradicts the existential permutation in (29), disproving the claimed universal permutation-style lower bound. The same conclusion holds with $H_1(0)$ in place of $H_2(0)$.

For this instance family, the subtree containing the globally largest leaf is also the optimal max–min subtree under every subtree-preserving permutation. Therefore, correctly identifying the largest leaf is sufficient to identify the optimal subtree. This result shows that the permutation-style lower bound in (29) does not hold in general. It does not rule out H_2-type lower bounds obtained using other alternative families or different quantifiers. Closing the sparsity gap between our upper and lower bounds remains.

F Further discussion of the upper-lower-bound gap

Our upper and lower bounds leave a gap when $\varepsilon = 0$ that is not yet fully characterized.

F.1 Dependence on (K, L)

We first compare the complexity terms for exact identification. For every suboptimal subtree $i \geq 2$, our gap definition gives $\Delta_{i,j} \geq \Delta_{i,1}$, while $\Delta_{1,1} = \Delta_{2,1}$. Therefore,

$$\begin{aligned} H_2(0) \leq H_1(0) = H(\nu) &\leq L \sum_{i=2}^{K} \frac{1}{\Delta_{i,1}^2} + \frac{1}{\Delta_{2,1}^2} + \sum_{j=2}^{L} \frac{1}{\Delta_{1,j}^2} \\ &\leq (L+1) H_{\mathrm{lb}}(\nu). \end{aligned} \tag{30}$$

Including the SR phase factor $\overline{\log}(KL)$, our upper-bound complexity is at most $O(L\overline{\log}(KL))$ times $H_{\mathrm{lb}}(\nu)$.

We next investigate whether a gap that grows with (K, L) is unavoidable. For ordinary Gaussian best-arm identification (BAI), Degenne [8, Theorem 8] show that, for any algorithm, there exists an instance with an $\Omega(\log K)$ gap in the asymptotic difficulty ratio. For Positivity, which asks whether all arm means exceed a given threshold, Degenne [8, Theorem 6] establish an analogous $\Omega(K)$ gap. These results show that adapting to an unknown instance can incur an unavoidable cost that grows with the number of arms. We show that both problems can be embedded into max–min identification, yielding an unavoidable $\Omega(\max\{L, \log K\})$ worst-case adaptation penalty.

Positivity embedding (L dependence). Consider $K = 2$ with unit-variance Gaussian leaves. Let

$$\mu_{1,1} = 0, \qquad \mu_{1,j} = M \quad (j \geq 2),$$

where $M > 0$ is sufficiently large. We may reveal the entire first subtree to the learner, so that its value $v_1 = 0$ is known. This only makes the identification problem easier.

For subtree 2, consider either the base instance

$$\mu_{2,j} = \beta \quad \text{for all } j \in [L],$$

or, for an unknown $r \in [L]$, an alternative

$$\mu_{2,r} = -\alpha, \qquad \mu_{2,j} = \beta \quad (j \neq r),$$

where $\alpha, \beta > 0$. In the base instance, subtree 2 is optimal, whereas in every alternative, subtree 1 is optimal. Since $v_1 = 0$ is known, the Positivity problem reduces to max–min identification on this instance family.

Applying the construction in the proof of Degenne [8, Theorem 6] with $\theta = 0$, $m = \beta$, and $\ell = -\alpha$ shows that at least one instance has an asymptotic difficulty ratio of at least

$$L\frac{\alpha^2}{(\alpha+\beta)^2}.$$

Taking $\alpha = \beta$ yields an $\Omega(L)$ lower bound on the worst-case adaptation penalty.

This construction extends to arbitrary $K \geq 2$ by adding $K - 2$ easy subtrees whose leaves have sufficiently negative known means. These subtrees remain strictly suboptimal under both the base instance and every alternative. Revealing them to the learner only makes the problem easier. Thus, the $\Omega(L)$ lower bound is preserved.

BAI embedding (K dependence). Conversely, ordinary K-arm Gaussian BAI can be embedded into max–min identification for arbitrary L. Consider a BAI instance with arm means $\theta_1, \ldots, \theta_K$. We construct a (K, L) max–min tree by setting

$$\mu_{i,1} = \theta_i, \qquad \mu_{i,j} = M \quad (j \geq 2),$$

where M is a sufficiently large known mean. Then $v_i = \theta_i$, so identifying the optimal subtree is equivalent to ordinary BAI. Applying this reduction to the hard instances in Degenne [8, Theorem 8] shows that, for any fixed-budget algorithm, there exists an instance with an $\Omega(\log K)$ gap in the asymptotic difficulty ratio.

Comparison and remaining gap. Combining the two embeddings, we obtain an unavoidable worst-case adaptation penalty of

$$\Omega(\max\{L, \log K\})$$

over the class of (K, L) max–min instances. On the other hand, the gap between our upper- and lower-bound complexity is at most $O(L\overline{\log}(KL))$. Since $\max\{L, \log K\} \geq L$, the known lower and upper adaptation penalties are within an $O(\overline{\log}(KL))$ factor.

We also considered whether the two constructions could be combined to establish an $\Omega(L \log K)$ adaptation penalty. Both proofs use a common base instance and construct alternatives by changing the mean of a single arm so that the correct answer changes. However, a direct combination does not appear to work. For a suboptimal subtree, raising a non-minimizing leaf does not change its min-value; only raising its minimizing leaf can increase the subtree's min-value. This makes it difficult to construct alternatives that change the optimal subtree in the same way. Determining the optimal joint dependence on (K, L) remains open.

F.2 Critical leaves and fixed-budget optimality

The fixed-confidence counterpart of maxmin action identification is now relatively well understood. The asymptotically optimal allocation is usually described through the characteristic time

$$T^\star(\mu)^{-1} = \sup_{w \in \Sigma} \inf_{\lambda \in \mathsf{Alt}(\mu)} \sum_a w_a d(\mu_a, \lambda_a),$$

where $\mathsf{Alt}(\mu)$ denotes the set of alternative instances with a different correct answer. For maxmin game trees, Kaufmann and Koolen [17] show in Section 5.2.2 that the generic GLR stopping rule, combined with Tracking, gives an asymptotically optimal fixed-confidence procedure for depth-two game trees under the required regularity assumptions on the oracle weights. In that setting, the problem has rank $L + 1$, and the oracle weights are obtained from the variational problem above; they are described as computable by disciplined convex optimization tools, but only numerically computable in general.

Our results address a different criterion. Fixed-confidence guarantees concern a data-dependent stopping time and a prescribed error probability δ, whereas fixed-budget guarantees concern the error probability after a deterministic budget T. Therefore, the fixed-confidence optimality result does not directly provide a fixed-budget error exponent

Nevertheless, the fixed-confidence analysis provides useful intuition about why maximin identification is structurally different from ordinary best-arm identification. In Section 6 of Garivier et al. [12], the authors analyze the simplest nontrivial $(2, 2)$ case and show that the optimal allocation can depend delicately on the relative positions of the leaf means. Translating their notation to ours, let

$$\mu_1 := \mu_{1,1}, \qquad \mu_2 := \mu_{1,2}, \qquad \mu_3 := \mu_{2,1}, \qquad \mu_4 := \mu_{2,2}.$$

In the regime $\mu_4 > \mu_2$, their fixed-confidence oracle allocation satisfies $w_4^\star(\mu) = 0$. Hence an asymptotically optimal strategy may sample leaf $(2, 2)$ only a vanishing fraction of the time, even though this leaf is part of the tree. This illustrates a phenomenon that is central to our lower-bound discussion: some leaves may be irrelevant to the closest way of confusing the identity of the optimal subtree.

A closely related phenomenon appears in our fixed-budget lower bound. For a base instance ν, the proof constructs alternative instances that change the identity of the optimal subtree while incurring small information cost. Such a closest confusing alternative need not perturb every leaf. In the $(2, 2)$ example above, the alternative may leave leaf $(2, 2)$ unchanged, so the corresponding lower-bound exponent does not involve the gap associated with that leaf. From an information-theoretic point of view, this leaf is not needed to create the most economical confusion.

This does not immediately imply a matching upper bound. In an upper-bound proof, the algorithm must operate without knowing in advance which leaves are structurally critical for the current instance. In particular, it must avoid a failure mode specific to max–min trees: if the true minimizing leaf of a suboptimal subtree is removed while the subtree remains active, then the surviving minimum of that subtree is artificially increased, and later phases may compare against a distorted max–min value. Therefore a sound fixed-budget algorithm may need to spend samples to distinguish whether a leaf is irrelevant or whether it is the minimizer that certifies the subtree's value. This is the reason our elimination rule is designed to be tree-safe.

Recent work on fixed-budget best-arm identification suggests that this phenomenon should not be surprising. Even in ordinary unstructured BAI, sharp fixed-budget optimality is more delicate than fixed-confidence optimality. Komiyama et al. [21] study minimax normalized error exponents and show that one natural optimal rate, denoted R^{go}, is tied to an oracle allocation depending on the final empirical distribution; they emphasize that achieving this rate by an actual adaptive algorithm is nontrivial. Their delayed optimal tracking construction attains a related asymptotic rate, but is primarily theoretical because it requires optimizing over high-dimensional allocation functions. Degenne [8] further shows that some fixed-budget identification problems do not admit a single complexity, in the sense of a lower bound on the error exponent that is attained by one algorithm on all instances; this failure already occurs in simple Bernoulli best-arm identification with two arms.

Taken together, these observations suggest that closing the remaining gap for fixed-budget max–min identification may require a sharper alignment between two objects: on the algorithmic side, sampling rules that adaptively discover which leaves are critical without corrupting the max–min structure; and on the information-theoretic side, lower bounds based on the truly closest confusing alternatives. Our results provide one step in this direction by giving an explicit ε-agnostic fixed-budget upper bound and a complementary lower-bound analysis that isolates a subset of structurally critical leaves.

G Experiments

In this section, we evaluate SR-MCTS against three fixed-budget baselines: uniform sampling, Bottom-up SAR, and SAR+Compare. We also compare it with budgeted adaptations of three fixed-confidence algorithms. SR-MCTS consistently outperforms the three fixed-budget baselines in our experiments, while budgeted M-LUCB achieves better performance on some instances. We further examine the sample allocation patterns of each method and show that SR-MCTS exhibits the desired adaptive allocation behavior. Finally, we provide empirical evidence supporting our theoretical bound. All simulations were run on a standard CPU machine; no GPU was used. SR-MCTS has low computational/memory cost. It stores the count and empirical mean of each leaf, uses $O(KL)$ memory, and requires only simple updates and min/max scans over the active leaves.

Baselines. The Bottom-up SAR baseline applies the multi-bandit Successive Accepts and Rejects algorithm for multi-bandit best arm identification in Bubeck et al. [4] to the negated rewards within subtrees, thereby identifying one candidate minimum leaf per subtree. SAR+Compare first uses this bottom-up stage with budget fraction α, and then uses Successive Rejects [1] to compare the selected candidate minima across subtrees. We set $\alpha = 0.8$ for all experiments.

Algorithm 3 MULTIBANDITSAR Subroutine

Require: A collection of bandit problems, budget B

1: Apply the Successive Accepts and Rejects algorithm of Bubeck et al. [4].
2: In each bandit problem, return one estimated best arm and its empirical mean.

Algorithm 4 SR Subroutine

Require: A set of arms, budget B

1: Apply Successive Rejects to the arms using budget B.
2: Return the final surviving arm and its empirical mean.

Algorithm 5 Uniform Sampling Baseline

Require: Budget T, depth-2 tree with K subtrees and L leaves per subtree

1: Sample all KL leaves as evenly as possible using budget T.
2: Compute the empirical min-value of each subtree:

$$\widehat{v}_i = \min_{j \in [L]} \widehat{\mu}_{i,j}.$$

3: **return**

$$\widehat{i}_T \in \arg\max_{i \in [K]} \widehat{v}_i.$$

Algorithm 6 Bottom-up SAR Baseline

Require: Budget T, depth-2 tree with K subtrees and L leaves per subtree

1: Treat each subtree as one bandit problem with arms given by its leaves.
2: Apply MULTIBANDITSAR with budget T to the negated rewards $-X_{i,j}$.
3: Let $\widehat{j}(i)$ be the selected leaf in subtree i.
4: Estimate each subtree by the empirical mean of its selected candidate minimum:

$$\widehat{m}_i = \widehat{\mu}_{i,\widehat{j}(i)}.$$

5: **return**

$$\widehat{i}_T \in \arg\max_{i \in [K]} \widehat{m}_i.$$

Algorithm 7 SAR+Compare Baseline

Require: Budget T, split parameter $\alpha \in (0, 1)$, depth-2 tree with K subtrees and L leaves

1: $T_{\min} \leftarrow \lfloor \alpha T \rfloor$ and $T_{\text{cmp}} \leftarrow T - T_{\min}$.
2: Apply MULTIBANDITSAR with budget $T_{\min}$ to the negated rewards $-X_{i,j}$.
3: Let $\widehat{j}(i)$ be the selected candidate minimum in subtree i.
4: Apply SR with budget T_{cmp} to the candidate leaves

$$\{(i, \widehat{j}(i)) : i \in [K]\},$$

using the original rewards.
5: **return** the subtree containing the surviving candidate leaf.

Budgeted Fixed-Confidence Algorithms. We reimplemented three fixed-confidence algorithms and adapted them to our fixed-budget setting: M-LUCB and M-Racing from Garivier et al. [12], and the asymptotically optimal Track-and-Stop (TaS) method from Kaufmann and Koolen [16]. We evaluate these methods on the structured and random instances. To adapt these methods to the fixed-budget setting, we do not use their stopping rules and run each sampling rule for exactly T samples. At the end of the budget, each method recommends the empirical maximin subtree. We set the confidence parameter to $\delta = 0.1$ for M-LUCB and M-Racing. The fixed-budget version of TaS does not require δ because its D-tracking sampling rule is independent of the confidence parameter.

Construction of (K, L)-Maxmin Trees. Since the tree value is determined by the minimum leaf value within each subtree, we directly control the minimum values of the subtrees. Specifically, we set these minimum values to be evenly spaced between 0 and -0.18. For the remaining leaf nodes, we consider two variants. In a structured version, for each subtree i, we sample or assign its leaf values from the interval between its minimum value and that value plus 0.04. In a random version, we sample the remaining leaf values uniformly from the interval between the subtree minimum value and 1. For all constructions, we set $K = 10$ and $L = 10$. Figure 1 visualizes the mean values of all leaf nodes for both instances. We use both the structured and random instances in Experiments 1 and 2, and use only the structured instance in Experiments 3 and 4. The default budget is 2000. For the first experiment, we vary the budget in the range of 2000 to 10000. Unless stated otherwise, we run 1000 independent trials and report standard errors whenever applicable.

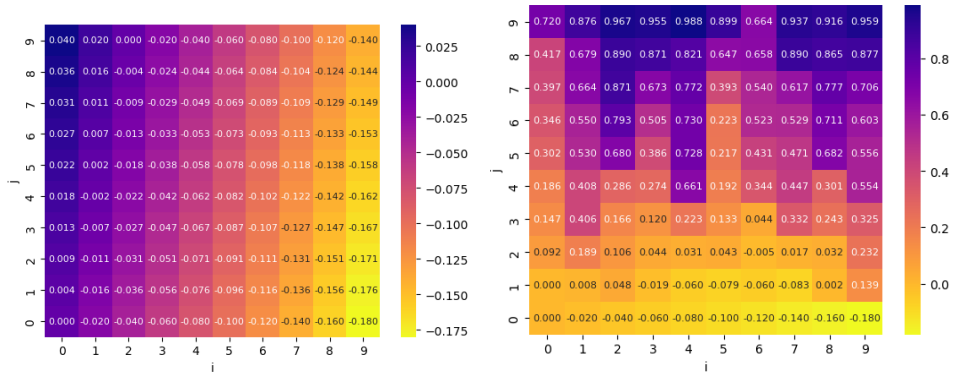

Figure 1: Visualization of the mean values in a max-min tree. We use the convention (i, j) to denote a leaf node, where i indexes the subtree and j indexes the leaf within that subtree. Top: structured instance. Bottom: random instance.

Experiment 1: Best-Subtree Identification. The first experiment evaluates best-subtree identification, where the goal is to correctly identify the subtree with the largest minimum value. Figure 2 illustrates the probability of misidentification as a function of the sampling budget T. As expected, the probability of misidentification decreases for all methods as more samples become available. Notably, SR-MCTS consistently statistically outperforms the competing fixed-budget methods across the entire range of budgets. These results demonstrate that SR-MCTS is highly sample-efficient, identifying the optimal subtree more reliably with fewer samples.

Figure 2 compares the misidentification probabilities of SR-MCTS, the three budgeted fixed-confidence methods, and the baselines on the structured and random instances. The misidentification probability decreases with the budget for every method on both instances. Among the fixed-confidence methods, budgeted M-LUCB performs best across the full budget range on both instances. When comparing SR-MCTS with budgeted M-LUCB, we observe different behavior on the two instances. On the structured instance, budgeted M-LUCB performs slightly better than SR-MCTS. However, on the random instance, the advantage of budgeted M-LUCB over SR-MCTS increases with the budget. At $T = 10000$, the misidentification probability of budgeted M-LUCB is approximately 0.18, compared with 0.31 for SR-MCTS. Overall, these results show that budgeted M-LUCB has strong performance that deserves further investigation.

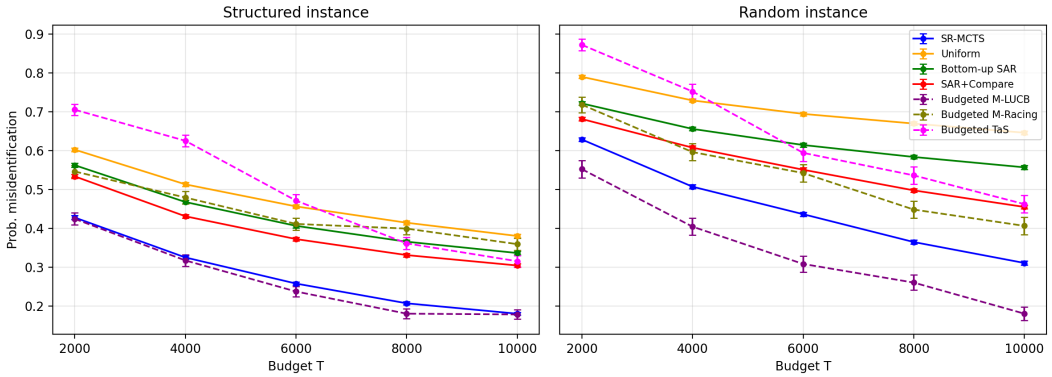

Figure 2: Probability of misidentification for all methods as the budget T increases (left: structured instance; right: random instance). Solid lines: the fixed-budget algorithms (10000 trials). Dashed lines: budgeted adaptations of the fixed-confidence algorithms of Garivier et al. [12] and Kaufmann and Koolen [16], each run for exactly T samples and recommending the empirical maximin subtree (1000/500 trials on the structured/random instances; bars are standard errors).

Experiment 2: ε-Good Subtree Identification. The second experiment considers ε-good subtree identification, where the goal is to identify any subtree whose minimum value is within ε of the optimal subtree. Recall from the preliminaries that subtree i is ε-good if $\mu_{i,1} \geq \mu_{1,1} - \varepsilon$. We note that all algorithms are ε-agnostic: the value of ε is not provided as an input. Nevertheless, the algorithms can still be evaluated under this relaxed objective by checking whether the returned subtree is ε-good. As ε increases, the problem becomes easier because more subtrees satisfy the ε-good condition. Consequently, we expect the probability of misidentification to decrease for larger values of ε. In our max-min tree construction, when $\varepsilon = 0.08$, approximately half of the subtrees are ε-good, making the identification problem substantially easier. This trend is observed for all methods in Figure 3. Across all values of ε, SR-MCTS consistently achieves a lower probability of misidentification, indicating that it is more effective than the competing methods at identifying near-optimal subtrees.

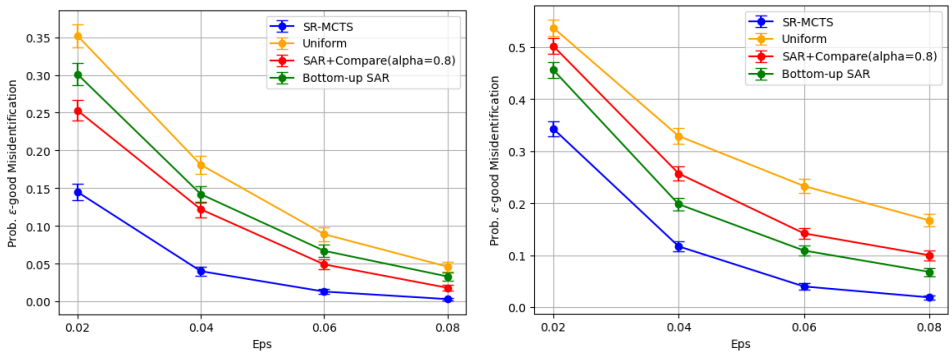

Figure 3: Probability of ε-good misidentification for all methods as ε increases. Top: structured instance. Bottom: random instance.

Experiment 3: Sample Allocation Patterns. The third experiment examines how samples are allocated across leaf nodes by SR-MCTS, SAR+Compare, and Bottom-up SAR as shown Figure 4, 5, and 6 respectively. The heatmaps show the average number of samples assigned to each leaf node, where i indexes the subtree and j indexes the leaf within that subtree. SR-MCTS exhibits a noticeably different allocation pattern compared with SAR+Compare and Bottom-up SAR. Both SAR+Compare and Bottom-up SAR allocate samples more uniformly across subtrees, with slightly higher sampling effort on the minimum-valued leaves within each subtree. In contrast, SR-MCTS concentrates more samples on critical regions of the tree, especially on leaves that are most informative for distinguishing promising subtrees from suboptimal ones. This adaptive behavior is desirable in the max-min tree setting. Once a subtree is clearly suboptimal, SR-MCTS can reduce sampling effort on that subtree and reallocate samples to more competitive regions. For example, bad subtrees whose maximum leaf value is already below the minimum value of the optimal subtree are likely to be eliminated early. As a result, SR-MCTS spends fewer samples on clearly inferior subtrees and focuses more on the leaf nodes that are most relevant for identifying the best subtree.

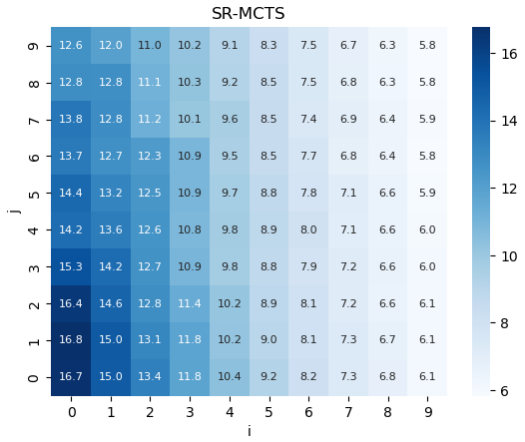

Figure 4: Heatmap of SR-MCTS Sampling Behavior

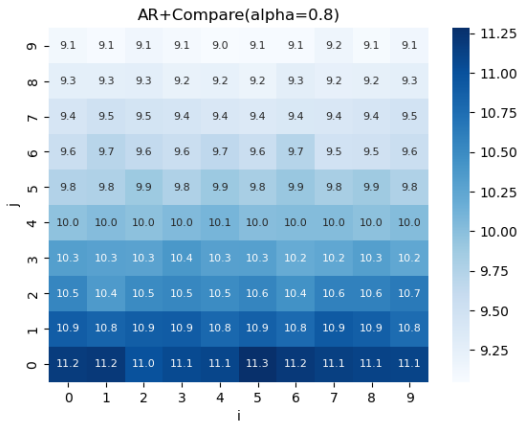

Figure 5: Heatmap of SAR+Compare Sampling Behavior

Phase Skipping. We modified SR-MCTS to record the number of phase skips and the identity of the subtree eliminated in each skip. We reran the structured instance from Figure 4 under the same setting: $K = L = 10$, $\sigma = 0.5$, and subtree minima ranging from 0 to -0.18. We evaluate budgets from $T = 2000$ to 10000, using 10000 trials per budget. We count a phase skip whenever SR-MCTS eliminates a subtree that still contains more than one active leaf. Since the instance has $KL = 100$ leaves, the schedule without phase skipping consists of 99 elimination phases.

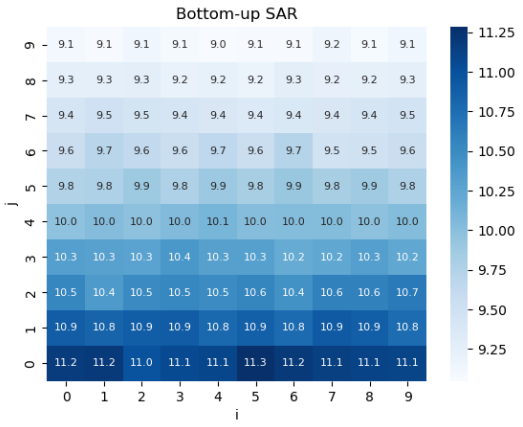

Figure 6: Heatmap of Bottom-up SAR Sampling Behavior

Figures 7(a,b) show, for each subtree, the fraction of trials in which that subtree is removed over the course of an SR-MCTS run. The probability of a subtree being eliminated increases monotonically with its suboptimality. At $T = 2000$, it rises from 0.38 for subtree 1 to 0.99 for subtree 9. A larger budget sharpens the pattern in both directions: at $T = 6000$, elimination is near-certain for every clearly suboptimal subtree (at least 0.88 from subtree 3 onward), while the probability for the optimal subtree drops from 0.18 to 0.08. This explains the mechanism behind the decreasing sample allocation across subtrees in Figure 4: subtrees with larger suboptimality gaps to the optimal subtree are removed earlier, so their leaves stop receiving samples in the later, larger phases.

Figure 7(c) shows the effect on the schedule. As T grows from 2000 to 10000, the number of elimination rounds per run falls from 78 to 62, while the number of skipped phases rises from 21 to 37. These two quantities sum to the 99 phases of the no-skipping schedule by construction. As more samples are allocated, SR-MCTS becomes more confident in eliminating subtrees, resulting in more phase skips.

Experiment 4: Empirical Validation of Theoretical Bounds. Next, we empirically examine whether the dependence on $H_2(\varepsilon)$ in our upper-bound analysis, defined in Eq. 6, is reflected in practice. Figure 8 plots the log probability of misidentification against the normalized theoretical scaling term $\frac{-T+KL}{\log(KL)H_2(\varepsilon)}$, ignoring constant factors. The approximately linear trend observed in the plot is consistent with our theoretical analysis and supports the interpretation of $H_2(\varepsilon)$ as a key problem-dependent complexity term governing the difficulty of max-min tree identification.

Experiment 5: Tree versus Flattened Complexity. We examine whether SR-MCTS follows the tree complexity or the flattened complexity. We conduct experiments on the instance from Proposition 26. We vary $\rho = 2^{-k/2}$ for $k = 2, \ldots, 9$. Over this range, ρ decreases from $1/2$ to 0.044, while the ratio $H_1^{\mathrm{tree}}(\rho)/H_1^{\mathrm{flatten}}(\rho)$ grows from 1.9 to roughly 500, so the two complexities increasingly diverge. Following Figure 8, Figure 9 plots the log probability of misidentification against the budget normalized by the two complexities, $\frac{T-KL}{H_1^{\mathrm{tree}}(\rho)}$ and $\frac{T-KL}{H_1^{\mathrm{flatten}}(\rho)}$, ignoring constant factors. Under the tree complexity, the curves for all eight values of ρ collapse onto essentially a single straight line (left panel), while under the flattened complexity the same curves spread far apart (right panel). For each ρ and each complexity H_1, we fit the linear model $\log \mathbb{P}(\widehat{i}_T \notin \mathcal{G}_0) = a - b\,\frac{T-KL}{H_1}$ and report the fitted slope b. If SR-MCTS follows the complexity H_1, then b should remain approximately constant as ρ varies. Under the tree complexity, the slopes are almost the same for all eight values of ρ, with a max-min spread of 1.21. By contrast, under the flattened complexity, the slopes vary substantially, with a max-min spread of 215. These results indicate that, on the instance from Proposition 26, SR-MCTS follows the tree complexity rather than the flattened complexity.

Experiment 6: Dependence on K and L. We conduct experiments on the (α, β)-spiked instances while varying K and L. Let $\alpha, \beta > 0$. The (α, β)-spiked instance is defined as follows: $\mu_{1,1} = 0$, $\mu_{i,1} = -\alpha$ for all $i \in [K] \setminus \{1\}$, and $\mu_{i,j} = \beta$ for all $(i, j) \in [K] \times ([L] \setminus \{1\})$. Our gap definitions

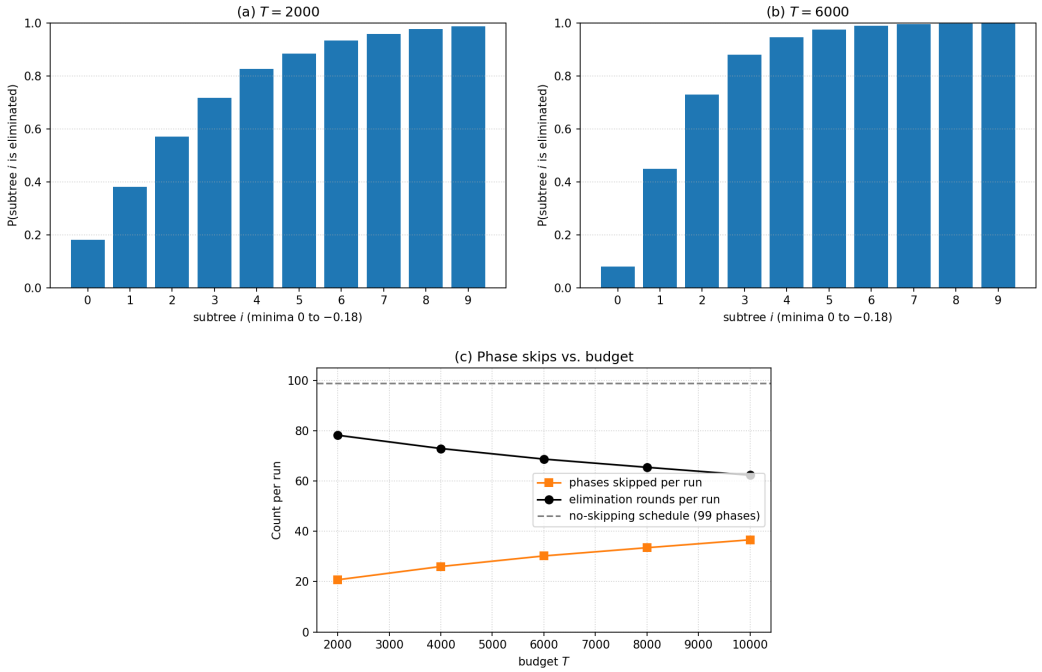

Figure 7: Phase skipping by SR-MCTS on the structured instance from Figure 4. (a,b) Probability that each subtree is block-eliminated at $T = 2000$ and $T = 6000$, respectively. (c) Average numbers of skipped phases and elimination rounds per run as the budget varies. Without phase skipping, the schedule requires 99 elimination rounds. Results are averaged over 10000 trials.

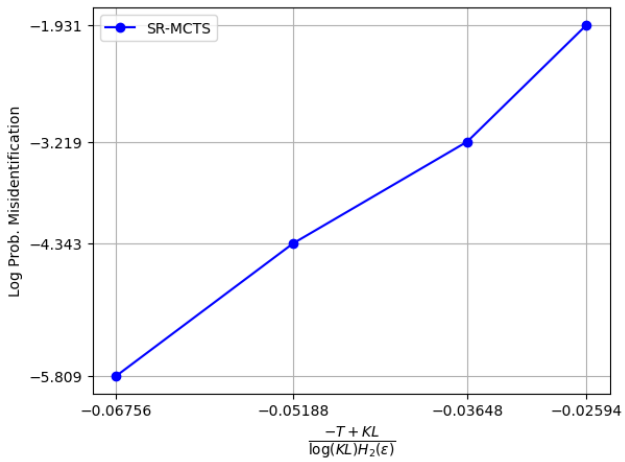

Figure 8: Log probability of misidentification for SR-MCTS as a function of the theoretical scaling term $\frac{-T+KL}{\log(KL)H_2(\varepsilon)}$

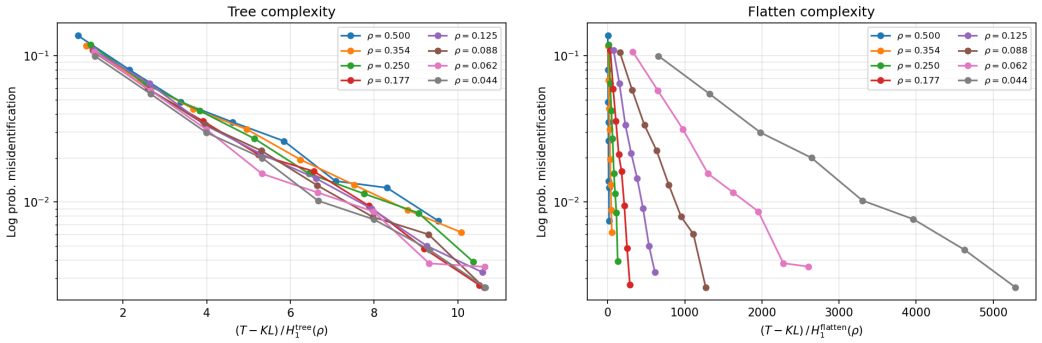

Figure 9: Comparison of SR-MCTS using the tree-complexity scaling term (left) and the flattened-complexity scaling term (right) on the instance from Proposition 26, with $\rho = 2^{-k/2}$ for $k = 2, \dots, 9$. Results are averaged over 10000 trials.

give

$$\Delta_{1,1} = \alpha, \qquad \Delta_{i,1} = \alpha \quad (i \geq 2), \qquad \Delta_{i,j} = \alpha + \beta \quad (j \geq 2,\ i \in [K]).$$

Therefore,

$$H_{\mathrm{lb}}(\nu) = \frac{K-1}{\alpha^2} + \frac{L-1}{(\alpha+\beta)^2},$$
$$H_1(0) = \frac{K}{\alpha^2} + \frac{K(L-1)}{(\alpha+\beta)^2},$$
$$H_2(0) = \max\left\{\frac{K}{\alpha^2}, \frac{KL}{(\alpha+\beta)^2}\right\}.$$

We consider two regimes. In the first regime, we set $\beta = \alpha = 0.5$, and in the second regime, we set $\alpha = 0.5$ and $\beta = 2.5$. In each regime, we study the effects of K and L separately. We vary the number of subtrees over $K \in \{2, 4, 8, 16, 32\}$ with $L = 8$ fixed, and we vary the number of leaves per subtree over $L \in \{2, 4, 8, 16, 32, 64, 128\}$ with $K = 8$ fixed.

We compare three complexity quantities: $H_{\mathrm{lb}}(\nu)$ from Eq. (7), which has $\Theta(K + L)$ dependence; $H_1(0)$, which has $\Theta(KL)$ dependence; and $\overline{\log}(KL)H_2(0)$, which is the complexity term appearing in our upper bound. To characterize the empirical behavior of our algorithm, we estimate H_{emp} by fitting the model $\mathbb{P}(\widehat{i}_T \notin \mathcal{G}_0) \approx C \exp(-T/H_{\mathrm{emp}})$. For each instance and budget, we run 10000 trials.

Figure 10 plots raw values so the levels of the measures relative to H_{emp} can be read directly. Across all four panels, H_{emp} stays within a small constant factor of the upper-bound complexity $\overline{\log}(KL)H_2(0)$. Figure 11 shows the growth rate of each complexity measure, obtained by dividing the raw values of each measure by its value at the first point of the sweep. In this view, the growth rate of H_{emp} is closer to $H_1(0)$ than to $H_{\mathrm{lb}}(\nu)$.

The varying-K plots reveal little difference among the measures, since every quantity, including $H_{\mathrm{lb}}(\nu)$, grows roughly linearly in K. The measures separate in the varying-L plots, where H_{emp} grows markedly faster than the $\Theta(K + L)$ quantity $H_{\mathrm{lb}}(\nu)$. This suggests that our algorithm follows the KL dependence rather than the $K + L$ dependence.

Experiment 7: Multiple Optimal Subtrees. We examine SR-MCTS on instances with multiple optimal subtrees. We define an (α, β)-spiked instance with q optimal subtrees as follows: $\mu_{i,1} = 0$ for $i \in [q]$, $\mu_{i,1} = -\alpha$ for all $i \in [K] \setminus [q]$, and $\mu_{i,j} = \beta$ for all $(i, j) \in [K] \times ([L] \setminus \{1\})$. A recommendation is considered correct if it selects any of these optimal subtrees.

We consider two parameter regimes: $(\alpha, \beta) = (0.5, 1.5)$ and $(\alpha, \beta) = (0.5, 2.5)$. We fix $K = L = 8$ and vary the number of optimal subtrees over $q \in \{2, 4, 6\}$. In both regimes, we can derive

$$H_2(0) = \max\left\{\frac{K}{\alpha^2}, \frac{K + q(L-1)}{\beta^2}, \frac{KL}{(\alpha+\beta)^2}\right\}$$

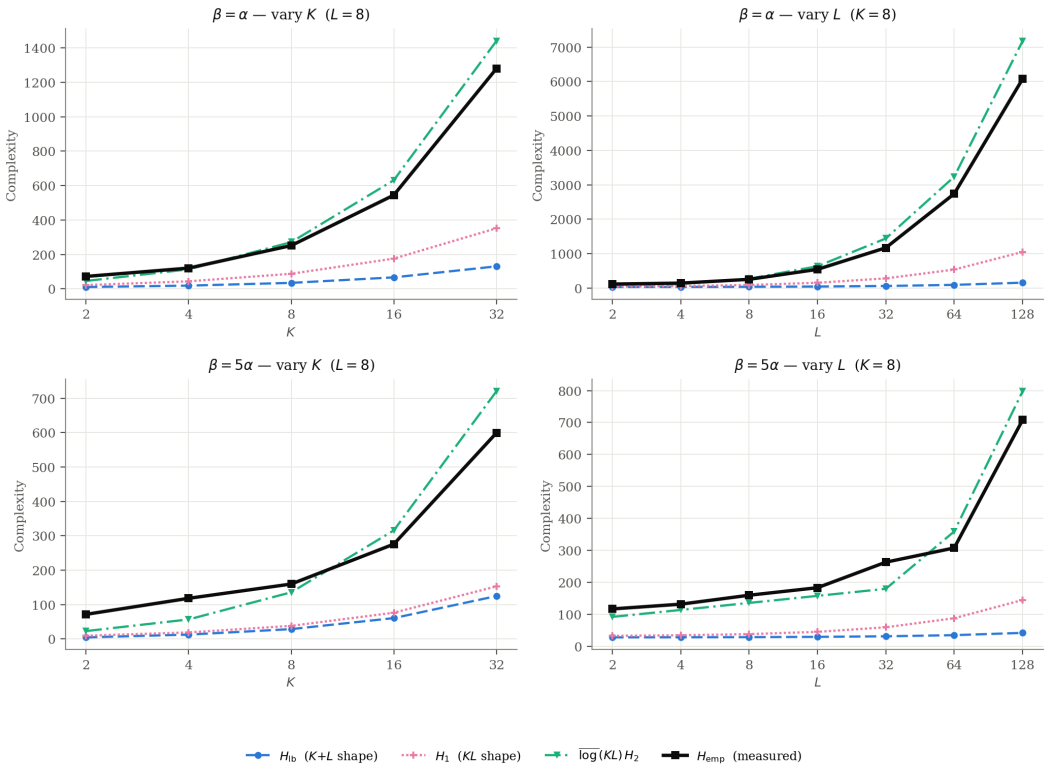

Figure 10: Comparison of the empirical complexity H_{emp} of SR-MCTS with three theoretical complexity measures on (α, β)-spiked instances. The top row uses $\beta = \alpha$, and the bottom row uses $\beta = 5\alpha$, with $\alpha = 0.5$ in both regimes. The left column varies K with $L = 8$, while the right column varies L with $K = 8$. Results are averaged over 10000 trials.

for all $q \geq 2$, thus the upper-bound complexity $\overline{\log}(KL)H_2(0)$ remains fixed at 135.80.

Figure 12 shows H_{emp}, estimated as in Experiment 6, as q varies over $\{2, 4, 6\}$, alongside the upper-bound complexity $\overline{\log}(KL)H_2(0)$. The measured H_{emp} remains below our upper bound and exhibits no systematic growth with the number of optimal subtrees. Thus, multiple optimal subtrees do not appear to significantly affect SR-MCTS, supporting that our analysis can hold without the uniqueness of the optimal subtree.

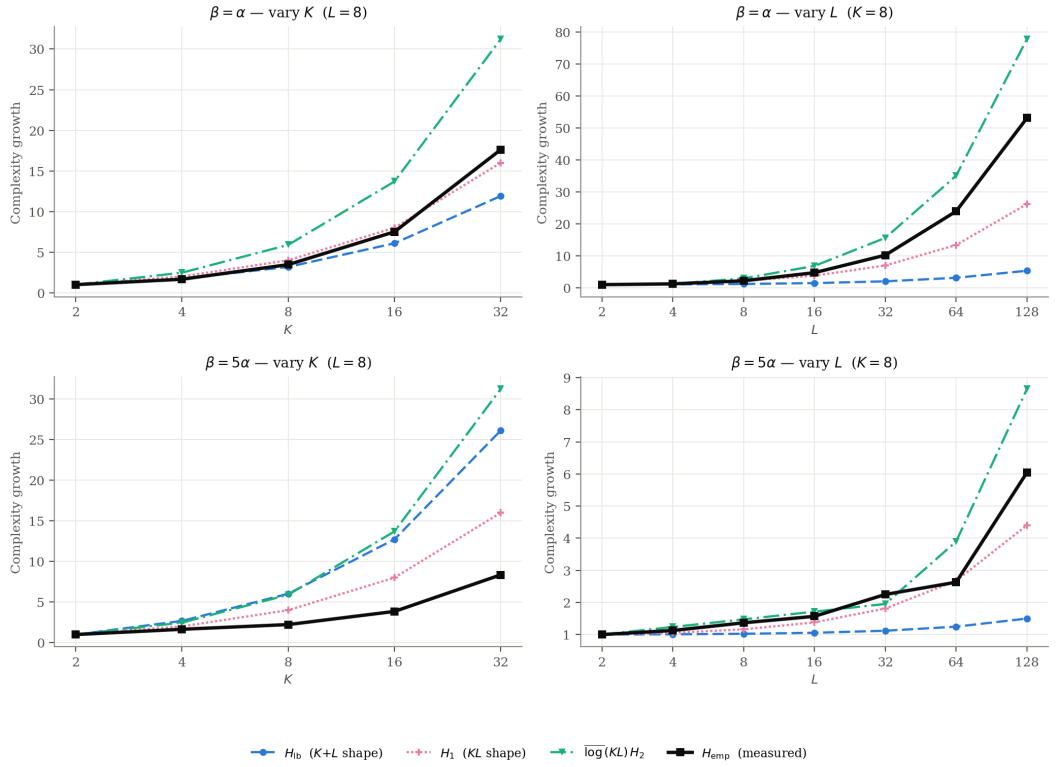

Figure 11: Growth of the empirical complexity H_{emp} of SR-MCTS compared with three theoretical complexity measures on (α, β)-spiked instances. The top row uses $\beta = \alpha$, and the bottom row uses $\beta = 5\alpha$, with $\alpha = 0.5$ in both regimes. The left column varies K with $L = 8$, while the right column varies L with $K = 8$. Results are averaged over 10000 trials.

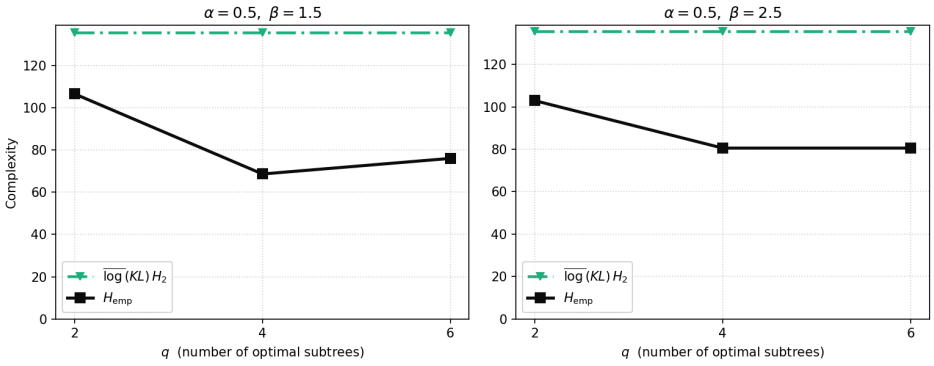

Figure 12: Performance of SR-MCTS on (α, β)-spiked instances with multiple optimal subtrees. We fix $K = L = 8$ and vary the number of optimal subtrees over $q \in \{2, 4, 6\}$. The left panel uses $\alpha = 0.5$ and $\beta = 1.5$, while the right panel uses $\alpha = 0.5$ and $\beta = 2.5$. A recommendation is considered correct if it selects any of the tied optimal subtrees. Results are averaged over 10000 trials.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the main theoretical contributions: an ε-agnostic fixed-budget algorithm for depth-2 max–min action identification, an instance-dependent upper bound, and lower-bound evidence. The scope is stated as depth-2 max–min trees with sub-Gaussian leaf rewards, and the relationship to fixed-confidence work is discussed in the introduction and Appendix A.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses limitations and open problems, including the remaining upper–lower gap, the restriction to depth-2 max–min trees, and the challenge of extending the analysis to deeper trees and more general tree-search/structured pure-exploration objectives. Additional discussion appears in Appendix F.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: The assumptions are stated in Section 2 and in the theorem statements, including the sub-Gaussian and Gaussian assumptions used for the upper and lower bounds, respectively. Complete proofs of the main upper bound, the complexity comparison lemma, and the lower bound are provided in Appendices B–D.2.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper includes synthetic experiments in Appendix G. The experimental setup specifies the instance families, reward distributions, baselines, budgets, evaluation metric, and number of independent trials. The experiments are illustrative and are not required for the main theoretical claims.

   Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [No]

   Justification: The experiments are synthetic and do not rely on external data. We do not include code in the submission. However, the instance definitions, baselines, budgets, and evaluation procedures are described in Appendix G, so the experiments can be independently reimplemented.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Appendix G describes the synthetic Gaussian problem instances, the algorithms compared, the budget range, the evaluation metric, and the number of repetitions. There is no training/test split or optimizer, since the experiments concern fixed-budget sampling algorithms rather than trained predictive models.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Appendix G reports results averaged over independent repetitions and includes uncertainty information, such as standard errors, for the plotted empirical error probabilities.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The experiments are lightweight synthetic simulations of fixed-budget bandit algorithms. Appendix G reports the basic compute environment or states that the experiments can be run on a standard CPU machine without specialized hardware.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work is theoretical research (see the main result in Section 3) on bandit algorithms and does not involve human subjects, private data, deployed systems, or high-risk data/model release. We are not aware of any aspect of the work that conflicts with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [N/A]

Justification: This is foundational theoretical work on fixed-budget pure exploration in structured bandit models. It does not introduce a deployed system, dataset, or model with a direct path to immediate societal impact.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: The paper does not release data or models, and the work does not involve high-risk artifacts such as pretrained generative models, scraped datasets, or deployed decision systems.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [N/A]

Justification: The paper does not use existing datasets, codebases, pretrained models, or other external assets as part of its main contribution.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not introduce new datasets, code packages, models, benchmarks, or other assets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human-subject research.

    Guidelines:

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

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

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

    Answer: [N/A]

Justification: The paper does not involve crowdsourcing or human-subject research, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: The core method development and theoretical results do not use LLMs as an important, original, or non-standard component.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.