FIXED-BUDGET AGENTIC PURE EXPLORATION WITH JUDGES

Anonymous authors
Paper under double-blind review

ABSTRACT

Pure exploration typically assumes that the learner can directly choose which actions to evaluate. However, agentic systems often differ from this classical interaction model because the actions available for evaluation are generated contextually by an agent from a large implicit space rather than directly selected by the learner, as in reasoning-and-acting (ReAct) systems and automated scientific discovery. We initiate the study of fixed-budget agentic pure exploration with judges, where a black-box action generator repeatedly proposes a small candidate set from a potentially large or infinite action space, and a separate judge decides which candidate to execute. The learner is additionally given offline reward data from different context–action pairs, together with a function class that captures shared reward structure across these pairs. The goal is to identify a near-optimal action for a target context (e.g., task) after a fixed budget of online interactions, even though the learner can only evaluate actions exposed by the generator. The offline data narrow the set of plausible reward functions, while online feedback progressively resolves the remaining uncertainty at the target context. We develop a simple judge that uses this evolving uncertainty through function class refinement, optimistic exploration, and pessimistic final selection. We establish a fixed-budget success guarantee whose error decays exponentially with the online budget, together with a nearly matching information-theoretic lower bound. The results show that the additional penalty induced by imperfect action generation is unavoidable in the worst case, while retaining the standard dependence on the remaining statistical parameters. Finally, we empirically evaluate the proposed framework and show improvements over corresponding baselines.

1 INTRODUCTION

Pure exploration is a fundamental problem in online learning, where the goal is to use sequential observations to identify a good action for a final recommendation rather than to optimize the cumulative rewards collected during the learning process (a.k.a. regret minimization). Two canonical formulations have been extensively studied. In the fixed confidence setting, the learner seeks to identify an optimal or approximately optimal action with a prescribed confidence level while using as few samples as possible. In the fixed budget setting, the learner is given a fixed budget and aims to maximize the probability that its final recommendation is optimal or approximately optimal. These formulations naturally arise when online interactions are used to support a final decision, including dose finding and treatment selection in clinical trials (Aziz et al., 2021; Wang & Tiwari, 2023), A/B testing (Kaufmann et al., 2014; Xiong et al., 2024), product development and optimization (Kanarios et al., 2024), and content discovery for recommendation systems (Aziz et al., 2022; Chen & Crawford, 2025). A rich literature (Even-Dar et al., 2006; Bubeck et al., 2009; 2013; Jedra & Proutiere, 2020; Li et al., 2022) has developed algorithms and statistical guarantees for pure exploration in multi-armed bandits, linear bandits, contextual bandits, and more general structured decision problems.

Classical pure exploration typically assumes that the learner is given a prescribed action set or can directly choose which actions to explore. Many modern agentic systems instead follow a generate-and-judge workflow, where a generator uses prior knowledge, the current context, and interaction history to propose a small set of promising candidates from a large or even infinite action space, while a separate judge, verifier, or policy decides which candidates to execute or pursue further. This broader pattern includes reasoning-and-acting (ReAct) systems (Yao et al., 2023), which interleaves

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

reasoning with actions and environmental observations. It also appears in robotic planning and manipulation, where generative models propose high-level or continuous actions that are evaluated by grounding models or history-aware verifiers (Brohan et al., 2023; Li et al., 2025), and in automated scientific discovery, where research agents generate hypotheses, experimental plans, or code variants that are evaluated by experiment managers or reviewer models (Weng et al., 2025; Yamada et al., 2025). In such systems, the hypothesis generator determines which actions become available, while the judge determines which of them are evaluated and receive online feedback. The generator is often a complex and potentially stochastic model whose future behavior is difficult to characterize exactly. This makes it difficult to design a fixed-confidence stopping rule that can reliably determine when the learner has explored enough. Even if the candidates observed so far are well understood, further interaction with the generator may expose substantially better actions that were previously unavailable. We therefore focus on the fixed-budget setting, which more closely reflects practical agentic workflows where the budget is typically specified in advance, and the goal is to maximize the probability of returning a near-optimal action within this budget.

Pure-exploration problems may also come with offline data from related past interactions, for example outcomes from previous experiments, historical evaluations of related products, or feedback collected on similar tasks. Such data can provide useful prior information before online interaction. Existing work has shown that historical observations can accelerate online bandit learning and reduce the online sample complexity of best arm identification (Cheung & Lyu, 2024; Yang et al., 2025). These works typically assume that offline reward observations are already available for the target bandit problem, often for the same prescribed arms that remain accessible for online sampling. In many agentic applications, however, the target task may be newly encountered at the start of online interaction, making such direct target-task data unavailable. Instead, one may have logged interactions and feedback from related tasks or contexts. When these source tasks share a common reward model or functional structure with the target task, their observations can provide transferable information about rewards at the target context (Kausik et al., 2025). Online feedback can then be used to resolve the remaining uncertainty over candidates generated during deployment.

In this paper, we study fixed-budget agentic pure exploration with offline data. The learner is given a context space $\mathcal{X} = \mathcal{X}_{\text{off}} \cup \{x_0\}$, a possibly infinite action space $\mathcal{A}$, an offline reward dataset $\mathcal{D}_{\text{off}}$ collected from source contexts in $\mathcal{X}_{\text{off}}$, and a function class $\mathcal{F}$ that models the mean reward over context-action pairs. The goal is to identify a good action for the target context x_0. In the automated scientific discovery example, x_0 may represent a new research problem, $\mathcal{A}$ the space of possible hypotheses, experimental plans, or code variants, $\mathcal{D}_{\text{off}}$ outcomes from related past research tasks, and $\mathcal{F}$ a class of models that predict the performance of candidate research actions. At each round t, a black-box action generator $\mathcal{M}$ proposes a finite candidate set $A_t \subseteq \mathcal{A}$, and a separate judge selects one action $W_t \in A_t$ for execution and observes its reward. The judge combines this online feedback with the offline data to progressively refine the set of plausible reward functions and guide subsequent selections. After a fixed budget of T online interactions, the learner recommends an action $\widehat{a}_T$ from those that have been executed. The goal is for $\widehat{a}_T$ to be ϵ-optimal with respect to the entire action space $\mathcal{A}$, including actions that may never have been proposed by the generator, with high probability.

Contributions. Our contributions are summarized as follows.

- We initiate the study of fixed-budget agentic pure exploration with judges, where the learner begins with offline reward observations from related contexts and acquires additional reward feedback through online interaction in a target context. The learner cannot directly choose from the underlying action space and instead explores through candidate actions adaptively exposed by a black-box action generator. We develop a simple judge that combines offline and online observations to progressively refine the plausible reward-function class, selects optimistic actions from the generated candidate lists, and ultimately returns a near-optimal action.
- We establish a fixed-budget success guarantee showing that $\widetilde{\mathcal{O}}\left(d/((1-\beta(\epsilon/2, x_0))\epsilon^2)\right)$ online reward observations are sufficient to return an ϵ-optimal action with high probability, where $\beta(\epsilon/2, x_0)$ characterizes the generator's probability of failing to expose an $\epsilon/2$-optimal action and d is a complexity measure of the reward-function class.
- We complement this upper bound with an information-theoretic lower bound of $\widetilde{\Omega}\left(d/((1-\beta(\epsilon/2, x_0))\epsilon^2)\right)$, showing that the leading dependence on action-generator coverage, function-class complexity, accuracy, and confidence is unavoidable in the worst case.

Moreover, this lower bound continues to hold even if the terminal recommendation is allowed to range over the entire action space, including actions that were never proposed or executed. Thus, enlarging the output range cannot improve the leading worst-case statistical rate, although it may still help on particular instances.

- We evaluate the framework in an exactly realizable synthetic environment and on noisy LCBench hyperparameter search, including a frozen-LLM baseline, to study how action generation, judging, offline information, and online feedback affect fixed-budget pure exploration.

Related Work. Pure exploration has been extensively studied in multi armed, linear, contextual, and more general structured bandit problems. Classical work develops best arm identification algorithms under fixed confidence and fixed budget criteria (Even-Dar et al., 2006; Bubeck et al., 2009; Audibert et al., 2010; Kalyanakrishnan et al., 2012; Bubeck et al., 2013; Jamieson et al., 2014), while later studies exploit structural assumptions to obtain sharper guarantees (Jedra & Proutiere, 2020; Li et al., 2022). Related objectives have also been studied in reinforcement learning (Jin et al., 2020; Menard et al., 2021; Narang et al., 2024). These formulations typically specify the relevant action set or decision process in advance, allowing the learner to directly choose what to explore. In contrast, our learner can only explore among candidate actions adaptively exposed by a history-dependent action generator. Sleeping-bandit models similarly restrict choices to currently available actions (Kanade et al., 2009; Kleinberg et al., 2010), but primarily study cumulative regret. We instead study fixed-budget terminal identification relative to the best action in the full underlying space, while transferring information from offline source contexts to online target-context exploration through a shared reward-function class.

Recent work has used LLMs and other generative models to search or construct actions in sequential decision problems. Language agents can generate candidate actions from the current context and interaction history, with a separate policy or search procedure (typically a judge) selecting which candidate to execute (Xu et al., 2024; Zhou et al., 2024). Prompt optimization methods similarly combine candidate generation with bandit based evaluation (Pryzant et al., 2023). In particular, Shi et al. (2024) study prompt optimization under a fixed evaluation budget by first constructing a candidate prompt pool and then applying best arm identification to select the best prompt. Most closely related, Russo et al. (2026) learn a history dependent pure exploration policy that directly chooses informative query actions, while Brooks et al. (2025) study bandits where a generative model transforms a selected query into an executed intervention under a cumulative regret objective. Our setting instead separates adaptive action generation from pure exploration selection. A black box generator reasons and determines which candidates are available at each round, while a separate judge must use a fixed online budget to explore among them and identify a near optimal final action.

A growing line of work studies how offline data can improve subsequent online decision making. In bandit learning, prior observations can reduce online exploration by providing information about arm rewards or shared reward structure (Cheung & Lyu, 2024; Kausik et al., 2025). More closely related, Yang et al. (2025) studies best-arm identification with possibly biased offline observations and characterizes when they can reduce the required online samples. These works typically assume a prescribed online action set and offline observations that directly inform the same target problem. This can be restrictive in agentic applications, where the target task may be new, offline data may come from related source contexts, and the learner cannot freely choose actions from the underlying space. In our setting, a function class links rewards across contexts and actions, allowing source-context data to rule out implausible reward functions and reduce uncertainty at the target context. The remaining uncertainty must then be resolved using only actions exposed by the history-dependent generator. This is important because offline data alone cannot guarantee a good final action. The generator must still expose actions that are either near-optimal or informative enough to distinguish among the remaining plausible functions.

2 PRELIMINARIES

Notations. For any $N \in \mathbb{N}$, we denote $[N] = \{1, \ldots, N\}$ and use $\Delta_N = \{x \in [0,1]^N : \sum_{i=1}^N x_i = 1\}$ to denote the probability simplex. For any set $\mathcal{A}$, we use $\Delta(\mathcal{A})$ to denote the probability simplex over $\mathcal{A}$. For any $x \in \mathbb{R}$, we write $(x)_+ = \max\{x, 0\}$ and $\sigma(x) = (1 + \exp(-x))^{-1}$.

We consider an agentic online learning setting where a decision-maker is given an action set $\mathcal{A}$, a context set $\mathcal{X}$, an action generator $\mathcal{M}$, and a judge. The decision-maker is given an offline dataset $D_{\text{off}} = \{(x_s, a_s, Y_s)\}_{s=1}^{n_0}$ containing n_0 reward samples, where Y_s is the reward observed from executing action a_s under context x_s. The context set $\mathcal{X} = \mathcal{X}^{\text{off}} \cup \{x_0\}$ consists of $\mathcal{X}^{\text{off}}$, the contexts appearing in the offline dataset, and a target context x_0, for which the agent aims to identify a near-optimal action. The decision-maker also has access to a finite function class $\mathcal{F} \subseteq \{f : \mathcal{X} \times \mathcal{A} \to [0, 1]\}$ (e.g., linear models, neural networks or LLMs) that models the mean reward. We assume that $\mathcal{F}$ is fixed before observing the reward labels in $\mathcal{D}_{\text{off}}$ used for localization, although it may be constructed from separate auxiliary data.

The action generator is a fixed stochastic kernel $\mathcal{M} : (x_0, \mathcal{H}_{t-1}) \to \Delta(\mathcal{A}^N)$, where $\mathcal{H}_t$ denotes the interaction history available up to and including round t, including the offline dataset, and $N \in \mathbb{N}$ is the number of actions proposed at each round. For instance, $\mathcal{M}$ can be instantiated by an LLM that, given the target context x_0 and the interaction history $\mathcal{H}_{t-1}$, generates a smaller action set of N candidate actions. More generally, our analysis treats $\mathcal{M}$ as a black-box action generator and does not impose any particular parametric form or generation mechanism. We write $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot|\mathcal{H}_{t-1}]$, and $\mathbb{P}_t(\cdot)$ for shorthand to denote the expectation and probability conditioning on history $\mathcal{H}_{t-1}$, respectively. The interaction protocol is as follows: for each round $t = 1, \ldots, T$

- The action generator proposes $A_t \sim \mathcal{M}(\cdot|x_0, \mathcal{H}_{t-1})$ such that $A_t \subseteq \mathcal{A}$ and $A_t = (a_{t,1}, \ldots, a_{t,N})$.
- Judge determines an action $W_t \in A_t$.
- Decision-maker executes the action W_t and observes a reward R_t.

For any context $x \in \mathcal{X}$ and action $a \in \mathcal{A}$, the observed reward sample of both online and offline is $f^\star(x, a) + \xi$, where ξ is independent zero-mean 1-sub-Gaussian. We make the following assumption, which captures the functional correlation across different (x, a) pairs and avoids explicit dependence on $|\mathcal{A}|$, which may be infinite.

Assumption 2.1 (Realizability). *There exists $f^\star \in \mathcal{F}$ such that for all s, $\mathbb{E}[Y_s|x_s, a_s] = f^\star(x_s, a_s)$ and $\mathbb{E}_s[R_s|A_s, W_s] = f^\star(x_0, W_s)$.*

Realizability is a standard assumption widely adopted in contextual and function-approximation-based online learning models (Chu et al., 2011; Agarwal et al., 2012; Foster & Rakhlin, 2020; Simchi-Levi & Xu, 2022). In practice, we construct $\mathcal{F}$ from available data. In this case, $f^\star$ may not exactly coincide with the true mean reward function, but can instead be interpreted as the function in $\mathcal{F}$ that best approximates it. We next state a coverage assumption on the action generator. For any $\epsilon \in (0, 1)$, we define $G_\epsilon(x_0) = \left\{a \in \mathcal{A} : f^\star(x_0, a) \geq \sup_{b \in \mathcal{A}} f^\star(x_0, b) - \epsilon\right\}$ as the set of ϵ-optimal actions.

Assumption 2.2 (Coverage). *There exists a function $\beta : [0, 1] \times \mathcal{X} \to [0, 1]$ such that for every t and every $r \in [0, 1]$, $\mathbb{P}_t\left(A_t \cap G_r(x_0) = \emptyset\right) \leq \beta(r, x_0)$.*

To keep the framework general, we treat the action generator as a black box, which may be an LLM, agentic search/sampling, retrieval system, or any other adaptive proposal mechanism. The coverage assumption characterizes the quality of the candidate set proposed by the action generator. It is mild since the generator quality is captured by an unknown and arbitrary function β, which generalizes the standard near-optimal reservoir-mass conditions used in infinitely many-armed bandits (Wang et al., 2008; Carpentier & Valko, 2015). This abstraction makes the judge and its theoretical analysis generator-agnostic. Appendix Lemma A.3 also gives the direct extension to time-varying coverage function β_t.

The goal is to maximize the probability of outputting a good action after T rounds. Formally, the decision-maker aims to maximize $\mathbb{P}(\hat{a}_T \in G_\epsilon(x_0))$ where $\hat{a}_T$ is the action output by the algorithm after T rounds.

3 A Judge for Fixed-Budget Agentic Pure Exploration

Our algorithm combines offline localization and online feedback to progressively refine the plausible reward-function class. The judge selects an optimistic action from each generated list, and a terminal pessimistic *decoder (i.e., the rule for selecting the final recommended action)* recommends one of the

executed actions. For a class of frozen LLM reward predictors, the values $f(x_0, a)$ come from those models, and online feedback updates only class membership, not the LLM parameters. Figure 2 in the Appendix illustrates this workflow.

3.1 Proposed Algorithm

Refinement of function class. The offline data and each new reward tell us which functions can still explain the observations. We retain several plausible functions rather than committing to the current best fit, so their remaining disagreement can guide exploration. For any $f \in \mathcal{F}$, define its cumulative squared loss up to round t as

$$S_t(f) := \sum_{s=1}^{n_0} \left(Y_s - f(x_s, a_s)\right)^2 + \sum_{i=1}^{t} \left(R_i - f(x_0, W_i)\right)^2. \tag{1}$$

For any $f, g \in \mathcal{F}$, we also define their cumulative disagreement on the observed context–action pairs by $D_t(f,g) := \sum_{s=1}^{n_0} \left(f(x_s, a_s) - g(x_s, a_s)\right)^2 + \sum_{i=1}^{t} \left(f(x_0, W_i) - g(x_0, W_i)\right)^2$. In particular, S_0 and D_0 depend only on the offline localization data. Only inputs with observed rewards enter these statistics. Predictions on unexecuted proposals do not provide additional statistical evidence.

The loss difference $S_t(f) - S_t(g)$ compares the fit of two functions, while $D_t(f,g)$ measures how much the observed inputs distinguish their predictions. Agreement on every observed input gives no evidence for choosing between them. Conversely, two functions that repeatedly predict different rewards should not both remain plausible merely because their realized losses happen to be close. Given a radius $\ell > 0$, fixed before using the localization rewards, we define the raw plausible class and its nested refinement as

$$\mathcal{V}_t := \left\{ f \in \mathcal{F} : \log \sum_{g \in \mathcal{F}} \exp\left(\frac{S_t(f) - S_t(g) + \frac{1}{2} D_t(f,g)}{4} \right) \le \ell \right\}, \tag{2}$$

$$\widetilde{\mathcal{V}}_0 := \mathcal{V}_0, \qquad \widetilde{\mathcal{V}}_t := \widetilde{\mathcal{V}}_{t-1} \cap \mathcal{V}_t \quad (t \ge 1). \tag{3}$$

The quantity inside each exponential is evidence against f relative to a competitor g. The loss difference is larger when f fits worse, and the disagreement correction makes this comparison more stringent when the observations distinguish the two predictions. Its positive sign does not arbitrarily penalize the truth. If f is correct, a new input (x, a) has conditional expected squared-loss difference $-(f(x,a) - g(x,a))^2$, so adding half this squared disagreement still leaves a negative expected contribution. The coefficients in Eq. (2) make the corresponding exponential a supermartingale under the stated noise condition, which controls erroneous exclusion as shown in Proposition A.1.

The log-sum-exp combines evidence from all competitors instead of selecting only the empirical minimizer. It always uses the original $\mathcal{F}$, keeping the comparison fixed as the active class shrinks. This refinement also controls uncertainty directly. Any two functions in $\mathcal{V}_t$ satisfy $D_t(f,g) \le 8\ell$, so two surviving models cannot disagree too much on the inputs already observed. Intersecting the raw sets retains all previous exclusions. Under Assumption 2.1, the true function belongs to every $\widetilde{\mathcal{V}}_t$ with probability at least $1 - |\mathcal{F}| e^{-\ell}$. We describe selection on this nonempty-class event. Appendix E specifies the complementary-path convention.

Optimistic action selection. Since the final output can only be selected from the actions played during the online phase, the judge should execute an action that is either potentially good or informative about the remaining reward uncertainty. For a nonempty $\widetilde{\mathcal{V}}_t$, define the optimistic and pessimistic values of action a by $U_t(a) := \max_{f \in \widetilde{\mathcal{V}}_t} f(x_0, a)$, $L_t(a) := \min_{f \in \widetilde{\mathcal{V}}_t} f(x_0, a)$. With high probability, $L_t(a) \le f^\star(x_0, a) \le U_t(a)$. If the true function were known, a natural choice would maximize it on the current list. Since it is unknown, the judge instead selects an action with the largest plausible reward, namely

$$W_t \in \operatorname*{argmax}_{a \in A_t} U_{t-1}(a). \tag{4}$$

The choice in Eq. (4) does not minimize worst-case per-round regret. Rather, if its uncertainty is small and the list contains a good action, then the selected action is itself good. If its uncertainty is

Algorithm 1: Judge with mixture refinement and a fixed online budget

Input. Offline data D_{off}, class $\mathcal{F}$, target context x_0, fixed generator $\mathcal{M}$, list size N, budget T, and radius $\ell > 0$ fixed before localization.

The judge computes $S_0, D_0, \mathcal{V}_0$, and $\widetilde{\mathcal{V}}_0$ using Eq. (1)–Eq. (3).

The initial history is $\mathcal{H}_0 \leftarrow \{D_{\text{off}}\}$.

for $t = 1, \ldots, T$ **do**

  Action generator proposes an action set $A_t = (a_{t,1}, \ldots, a_{t,N}) \sim \mathcal{M}(\cdot \mid x_0, \mathcal{H}_{t-1})$.

  Execute an action $W_t \in \operatorname{argmax}_{a \in A_t} \max_{f \in \widetilde{\mathcal{V}}_{t-1}} f(x_0, a)$ and observes reward R_t.

  Update statistics $S_t, D_t, \mathcal{V}_t$, and $\widetilde{\mathcal{V}}_t$ using Eq. (1)–Eq. (3).

  The judge updates history $\mathcal{H}_t$ by appending (A_t, W_t, R_t).

Output. The algorithm recommends action $\widehat{a}_T$ according to Eq. (5).

large, observing its reward contributes information that can shrink the plausible class. The selection requires only evaluating the finite candidate list and does not invoke a global maximization oracle. For an explicit function class, selection and refinement require $\mathcal{O}(|\mathcal{F}|N + |\mathcal{F}|^2)$ arithmetic per round, apart from the computational cost of evaluating the reward predictors.

Terminal recommendation. At the end of T rounds, the algorithm outputs an action by

$$\widehat{a}_T \in \operatorname*{argmax}_{a \in \{W_1, \ldots, W_T\}} \min_{f \in \widetilde{\mathcal{V}}_T} f(x_0, a). \tag{5}$$

At a high level, the pessimistic rule recommends an action that is good under every model the data have not yet ruled out. Nesting ensures that the lower value $L_t(a)$ cannot decrease over time. Therefore, once an executed action receives a good lower-confidence certificate, that certificate remains available for the terminal recommendation.

3.2 Main Result

We give a fixed-budget guarantee for the post-offline function class and then its worst-case specialization, using the eluder dimension (Russo & Van Roy, 2013) defined below.

Eluder dimension. For a function class $Q \subseteq \mathcal{F}$ and a scale $\alpha > 0$, an action $a \in \mathcal{A}$ is α-independent of $a_1, \ldots, a_m$ at the target context x_0 if there exist $f, g \in Q$ such that $\sum_{j=1}^{m} (f(x_0, a_j) - g(x_0, a_j))^2 \le \alpha^2, |f(x_0, a) - g(x_0, a)| > \alpha$. The eluder dimension $d_Q(\alpha)$ is the largest length of a sequence in which each action is α-independent of its predecessors at this same scale (Russo & Van Roy, 2013). We set $d_{\varnothing}(\alpha) = 0$ and use $d_{\mathcal{F}}(\alpha)$ for the full function class. In plain terms, two plausible reward functions may make similar predictions on all earlier actions but substantially different predictions on a new action. The eluder dimension counts how many such new uncertainties can arise in sequence before earlier observations rule them out. For a finite function class, $d_Q(\alpha) \le \binom{|Q|}{2}$, since each independent position requires a distinct unordered pair of witnessing models.

We condition on any independent auxiliary data used to construct $\mathcal{F}$, which is then fixed before localization. Each offline input is chosen before its reward, whose noise is conditionally mean-zero and 1-sub-Gaussian given the pre-reward information. The following result allows deterministic or adaptive offline inputs and does not require i.i.d. sampling.

Theorem 3.1 (Fixed-Budget Success Guarantee)**.** *Suppose Assumption 2.1, Assumption 2.2, and the sampling conditions above. Given an integer budget $T \ge 1$, target context x_0, and error level $\epsilon \in (0, 1)$, suppose $\beta(\epsilon/2, x_0) < 1$. For any radius $\ell > 0$ fixed before using the offline localization rewards, running Algorithm 1 guarantees*

$$\mathbb{P}(\widehat{a}_T \in G_\epsilon(x_0)) \ge \left(1 - |\mathcal{F}| e^{-\ell} - \mathbb{E}\left[\exp\left\{-\frac{\left((1 - \beta(\epsilon/2, x_0))T - d_{\mathcal{V}_0}(\epsilon/2)\lceil 32\ell/\epsilon^2 \rceil\right)_+^2}{2(1 - \beta(\epsilon/2, x_0))T}\right\}\right]\right)_+, \tag{6}$$

where the expectation is over the offline localization data. In particular, using $d_{\mathcal{V}_0}(\epsilon/2) \le d_{\mathcal{F}}(\epsilon/2)$ *and choosing* $\ell = (1-\beta(\epsilon/2, x_0))T\epsilon^2/(128\max\{1, d_{\mathcal{F}}(\epsilon/2)\})$ *gives the worst-case guarantee*

$$\mathbb{P}\left(\widehat{a}_T \in G_\epsilon(x_0)\right) \ge \left(1 - (|\mathcal{F}|+1)\exp\left\{-\frac{(1-\beta(\epsilon/2, x_0))T\epsilon^2}{128\max\{1, d_{\mathcal{F}}(\epsilon/2)\}}\right\}\right)_+. \tag{7}$$

As a consequence, for any $\delta \in (0,1)$*, under this horizon-calibrated choice a sufficient budget for success probability at least* $1-\delta$ *is* $T = \mathcal{O}\left(\frac{\max\{1, d_{\mathcal{F}}(\epsilon/2)\}\log((|\mathcal{F}|+1)/\delta)}{(1-\beta(\epsilon/2, x_0))\epsilon^2}\right)$.

The dimension $d_{\mathcal{V}_0}(\epsilon/2)$ measures target-relevant prediction uncertainty left after the realized offline data, not the number of models. Offline data help when they exclude functions whose disagreements contribute to this uncertainty, so the residual dimension can be much smaller than $d_{\mathcal{F}}(\epsilon/2)$. For a fixed radius and budget, Eq. (6) averages this benefit over offline datasets.

Conversely, if all models agree on every offline input, then $\mathcal{V}_0 = \mathcal{F}$ whenever $\ell \ge \log|\mathcal{F}|$, regardless of n_0. This is the uninformative case in Section 4. Thus informativeness at the target, not sample size alone, determines the improvement captured by the bound. The radius in Eq. (7) requires supplied valid coverage information. A coverage-free choice appears in Appendix B.4. The proof is in Appendix A.5.

4 Lower Bound

This section presents our lower-bound results. The proofs are deferred to Appendix C.

4.1 Fixed-Budget Problem Lower Bound

For a fixed function class and generator, $\mathbb{P}_f$ denotes the interaction transcript when $f \in \mathcal{F}$ is the true reward function, and $G_\epsilon^f(x_0)$ denotes its set of ϵ-optimal actions.

Theorem 4.1. *For every integer* $d \ge 1$, $0 < \epsilon \le 1/8$, $N \ge d+1$, *and prescribed value* $\beta(\epsilon/2, x_0) \in [0,1)$, *there is a function class* $\mathcal{F}$ *with* $|\mathcal{F}| = d+1$, $d_{\mathcal{F}}(\epsilon/2) = d$, *and a fixed generator with conditional coverage exactly* $1-\beta(\epsilon/2, x_0)$ *at tolerance* $\epsilon/2$, *such that every algorithm obeying* $W_t \in A_t$ *and* $\widehat{a}_T \in \{W_1, \dots, W_T\}$ *satisfies*

$$\max_{f\in\mathcal{F}} \mathbb{P}_f\left(\widehat{a}_T \notin G_\epsilon^f(x_0)\right) \ge \frac{1}{4}\exp\left(-\frac{8\left(1-\beta\left(\epsilon/2, x_0\right)\right)T\epsilon^2}{d}\right). \tag{8}$$

The same function class and generator work for all horizons $T \ge 1$ *and all algorithms.*

Offline sample size alone cannot improve the worst-case rate because the hard instance leaves all target uncertainty unresolved offline. Only a fraction $1-\beta(\epsilon/2, x_0)$ of the rounds then provide useful reward information, producing the multiplicative interaction between residual uncertainty and generator coverage. This is a statistical obstruction, not a limitation of a particular judge.

4.2 Enlarging Output Range Does Not Improve Leading Worst-Case Rate

We now keep the restriction $W_t \in A_t$ on reward queries, but allow the terminal recommendation to be any action in $\mathcal{A}$, including actions that were never proposed or executed. In particular, the learner may compute $\widehat{a}_T \in \operatorname{argmax}_{a\in\mathcal{A}} \widehat{f}_T(x_0, a)$ for any learned score $\widehat{f}_T$, or use any other measurable randomized decoder. We grant unlimited computation and knowledge of the function class and generator, but do not reveal the unknown true function or its maximizer.

For a precise worst-case comparison, consider all finite-action instances satisfying the assumptions with function class size at most $d+1$, dimension at scale $\epsilon/2$ at most d, size-N lists, and conditional coverage at least $1-\beta(\epsilon/2, x_0)$. Write $\mathcal{E}$ for the executed-only output rule $\widehat{a}_T \in \{W_1, \dots, W_T\}$, in contrast to the full-space rule $\widehat{a}_T \in \mathcal{A}$. Let $\mathfrak{e}_T^{\mathcal{E}}$ and $\mathfrak{e}_T^{\mathcal{A}}$ be the optimal worst-case error probabilities under these two rules over the same instance class. Both restrict reward queries to generated lists. Let $T_{\mathcal{E}}(\delta)$ and $T_{\mathcal{A}}(\delta)$ be the smallest integer budgets for which an algorithm under the respective output rule guarantees success at least $1-\delta$ uniformly over this class. Formal definitions appear in Appendix C.3.

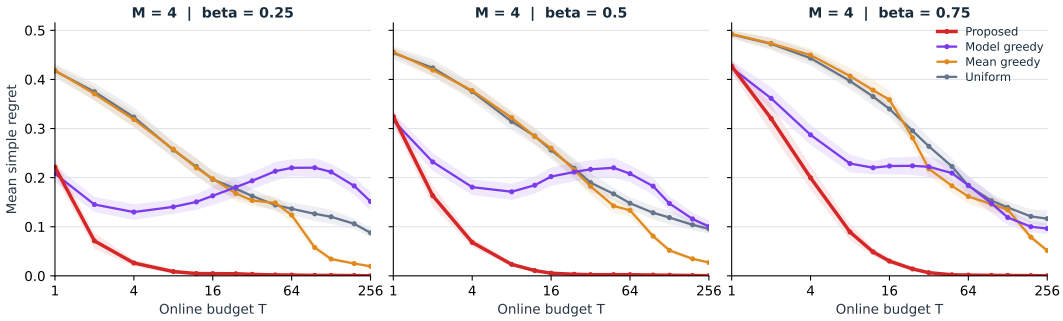


Figure 1: Synthetic simple regret with four reward models and three generator miss probabilities. Each curve averages 20 independent runs of 128 balanced episodes. Bands show one sample standard deviation across run means, not confidence intervals. Lower is better.

Theorem 4.2 (Fixed-Budget Comparison of Output Ranges)**.** *Under the parameter conditions of Theorem 4.1, its function class and generator also satisfy Eq. (8) for every decoder allowed to recommend from the entire action space. Moreover, for every $T \geq 1$,*

$$\frac{1}{4}\exp\left\{-\frac{8(1-\beta(\epsilon/2,x_0))T\epsilon^2}{d}\right\} \leq \mathfrak{e}_T^{\mathcal{A}} \leq \mathfrak{e}_T^{\mathcal{E}} \leq \min\left\{1,(d+2)\exp\left\{-\frac{(1-\beta(\epsilon/2,x_0))T\epsilon^2}{128d}\right\}\right\}. \quad (9)$$

As a consequence, for $\delta \in (0,1/4]$, the required budgets satisfy

$$\frac{d\log(1/(2\delta))}{16(1-\beta(\epsilon/2,x_0))\epsilon^2} \leq T_{\mathcal{A}}(\delta) \leq T_{\mathcal{E}}(\delta) \leq \left\lceil \frac{128d\log((d+2)/\delta)}{(1-\beta(\epsilon/2,x_0))\epsilon^2} \right\rceil. \quad (10)$$

For fixed-budget error, Eq. (9) shows the same worst-case exponential dependence on $(1-\beta(\epsilon/2,x_0))T\epsilon^2/d$, up to constants and the displayed class-size prefactor. Enlarging the recommendation range supplies no additional reward information while queries remain restricted to generated lists.

For sample complexity, the lower and upper bounds in Eq. (10) match up to constants when $\log(1/\delta)$ dominates $\log(d+1)$, for example when $\delta \leq \min\{1/4, 1/(d+1)\}$. In this worst-case, high-confidence regime, enlarging the output range does not improve the order of the required budget. It can still help on particular instances by allowing an unexposed good action to be recommended, as in Proposition C.3. The displayed comparison requires $N \geq d+1$. Appendix D gives a separate lower bound for every fixed $N \geq 2$.

5 Experiments

We evaluate the judge in an exactly realizable synthetic environment and on a public benchmark for hyperparameter optimization. Both experiments measure simple regret (Bubeck et al., 2009), the difference between the best true mean reward and that of the recommended action. Every method recommends an action among all played actions across T rounds. We compare the proposed optimistic selector with pessimistic decoding against *Model greedy*, *Mean greedy*, and *Uniform*, all using empirical-mean decoding. Model greedy follows the reward predictor with the smallest cumulative squared loss, Mean greedy favors untried candidates before exploiting observed means, and Uniform samples a candidate uniformly. To isolate judging algorithm quality while retaining reproducible candidate exposure, both experiments use a statistical hypothesis generator designed to mimic the selective proposal behavior of an LLM-based hypothesis generator. Further details appear in Appendix F.

5.1 Synthetic Experiment

The target has 50 actions, including a shared incumbent (i.e., a common baseline action with the same mean reward under every model) and configuration-specific variants. We evaluate our approach using

Table 1: LCBench simple regret. Entries are the mean and sample standard deviation across five target-context means, each averaged over ten seeds. Bold marks the lowest mean at each budget. These standard deviations describe variation across target contexts. Lower is better.

Method	$T = 16$	$T = 32$	$T = 64$	$T = 128$	$T = 256$
Proposed	$\mathbf{.0291} \pm .0234$	$\mathbf{.0271} \pm .0227$	$\mathbf{.0264} \pm .0215$	$\mathbf{.0296} \pm .0231$	$\mathbf{.0287} \pm .0219$
Model greedy	$.0370 \pm .0187$	$.0319 \pm .0125$	$.0330 \pm .0124$	$.0363 \pm .0148$	$.0358 \pm .0174$
Mean greedy	$.0409 \pm .0242$	$.0382 \pm .0216$	$.0366 \pm .0171$	$.0370 \pm .0172$	$.0300 \pm .0118$
Uniform	$.0485 \pm .0287$	$.0415 \pm .0224$	$.0398 \pm .0156$	$.0448 \pm .0179$	$.0394 \pm .0154$
LLM + empirical	$.0446 \pm .0272$	$.0378 \pm .0178$	$.0372 \pm .0157$	$.0372 \pm .0178$	$.0380 \pm .0178$
LLM + direct	$.0784 \pm .0431$	$.0638 \pm .0247$	$.0603 \pm .0304$	$.0781 \pm .0291$	$.0723 \pm .0138$

function classes of three different sizes, containing 4, 8, or 16 models. Sixteen noisy observations from eight source contexts separate the models into coarse groups while leaving the optimal target action unresolved. At each round, the action generator exposes 17 candidates. With probability $1 - \beta$, the candidate set contains an optimal action for every model, while otherwise it contains only uninformative low-reward actions. We vary $\beta \in 0.25, 0.50, 0.75$ and use Gaussian noise with standard deviation 0.25. The proposed method uses the known-noise refinement in Appendix B.5 with radius $\ell_T = \log(0.5MT)$, where $M = |\mathcal{F}|$, and is initialized separately for each of 14 budgets between 1 and 256.

Figure 1 shows that the proposed method rapidly reduces simple regret as the budget increases. At $T = 256$, its mean regret is between 0.00014 and 0.00027, compared with 0.01946–0.05156 for the best baseline in each panel. More frequent generator misses delay the improvement. The six additional panels for $M = 8$ and $M = 16$ appear in Figure 3 in Appendix F.1, with the same qualitative pattern. The advantage is not uniform at the smallest budgets, particularly for larger classes. Same-trajectory decoder comparisons in Table 3 show that both exploration and terminal decoding contribute to the improvement.

5.2 LCBench Hyperparameter Search

We evaluate the performance of our proposed judge on five LCBench tasks (Zimmer et al., 2021), treating each task as a target context. For each target context, the action space contains $2{,}000$ neural-network hyperparameter configurations. The mean reward of an action is its archived validation balanced accuracy, and each execution returns this mean plus independent Gaussian noise with standard deviation 0.10. At each round, all methods receive the same list of eight candidate actions, with four drawn from a fixed portfolio of 256 high-scoring actions for that target context and four from the rest of the action space. After a fixed budget, each method recommends one of its executed actions, and we measure simple regret relative to the true best action among all $2{,}000$ configurations. Unlike the realizable setting in our theory, the true mean reward function need not belong to the fitted reward-function class, while simple regret is still evaluated using the archived true means.

We compare the proposed method with the three baselines above and two LLM judge variants. The proposed method uses a frozen 16-member reward-function class. Each member is a linear ridge predictor on shared frozen Qwen3.5-0.8B representations. All members use the same encoder and differ only in their linear readouts, which are fitted using different reweightings of 20 source contexts. The bank is fixed before target interaction and is not trained on these five target contexts. The two LLM judge variants instead use a frozen Qwen3.5-0.8B to select directly from the eight candidates and share the same online trajectories. *LLM + empirical* uses empirical-mean terminal decoding, while *LLM + direct* independently scores each executed action using the LLM and recommends the action with the largest log-probability margin between "Yes" and "No". Proposed, Model greedy, Mean greedy, and Uniform share a reward-independent calibration prefix counted within the online budget, while the direct-LLM policy controls every round. Further details appear in Appendix F.2.

The proposed method has the lowest mean regret at every budget in Table 1, with an average regret of 0.02818 across the five reported budgets, compared with 0.03479 for Model greedy and 0.03894 for the LLM with empirical decoding. On the same LLM trajectories, replacing empirical decoding with direct LLM decoding increases the average regret from 0.03894 to 0.07056, showing that terminal recommendation quality can substantially affect performance in this benchmark.

AI USE STATEMENT

In this work, we used generative AI tools to help develop theoretical models and conceptual frameworks, formulate mathematical claims, suggest proof ingredients, assist in writing and checking proofs, refine hypotheses, design the synthetic data-generating process and experiments, implement methods, and interpret results. The pretrained LLM used as an experimental baseline is described in Appendix F.2. We did not use generative AI to replace the archived LCBench reward measurements with generated values, and qualitative or thematic data analysis is not applicable to this work. Additionally, we used generative AI for literature search, drafting and editing text, creating or editing code, preparing scientific figures, and formatting references and research artifacts. We did not use generative AI for translation or for cleaning or reformatting datasets, and qualitative or thematic data analysis is not applicable to this work.

Checks of the AI-assisted material included reviewing mathematical arguments and boundary cases against the stated assumptions and cited sources, running the synthetic implementation's correctness tests, and numerically reproducing all reported synthetic configurations. These checks include AI-assisted verification and do not replace the authors' review of the final text, claims, figures, and code. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work is methodological and does not involve human subjects, user studies, or the collection of new personal data. Our experiments use a synthetic environment and the public LCBench benchmark, and no learned system is deployed to users or used to make real-world decisions. The proposed framework is general and could potentially be applied in high-stakes domains. Such applications would require domain-specific validation, appropriate human oversight, and safeguards beyond the assumptions and guarantees studied in this work. We are not aware of additional ethical concerns specific to the experiments reported here.

REPRODUCIBILITY STATEMENT

The learning setting and assumptions are specified in Section 2, and the proposed judge and its guarantees are described in Section 3. Complete proofs of the upper bounds are provided in Appendix A, additional guarantees and radius choices in Appendix B, and the lower-bound proofs in Appendix C and Appendix D. Implementation details for the proposed judge appear in Appendix E. The experimental protocols are described in Appendix F, with the synthetic experiment in Appendix F.1 and the LCBench experiment in Appendix F.2. These sections specify the data construction, candidate generation, reward-function classes, compared methods, budgets, randomization, and evaluation procedures needed to reproduce the reported results. LCBench is a publicly available benchmark and is cited in the paper.

REFERENCES

Alekh Agarwal, Miroslav Dudík, Satyen Kale, John Langford, and Robert Schapire. Contextual bandit learning with predictable rewards. In *Artificial intelligence and statistics*, 2012.

Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. Best arm identification in multi-armed bandits. In *Conference on Learning Theory*, 2010.

Maryam Aziz, Emilie Kaufmann, and Marie-Karelle Riviere. On multi-armed bandit designs for dose-finding trials. *Journal of Machine Learning Research*, 22(14):1–38, 2021.

Maryam Aziz, Jesse Anderton, Kevin Jamieson, Alice Wang, Hugues Bouchard, and Javed Aslam. Identifying new podcasts with high general appeal using a pure exploration infinitely-armed bandit strategy. In *Conference on Recommender Systems*, 2022.

Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on robot learning*, 2023.

Marc Brooks, Gabriel Durham, Kihyuk Hong, and Ambuj Tewari. Generator-mediated bandits: Thompson sampling for GenAI-powered adaptive interventions. In *Advances in Neural Information Processing Systems*, 2025.

Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *International conference on Algorithmic learning theory*, 2009.

Séebastian Bubeck, Tengyao Wang, and Nitin Viswanathan. Multiple identifications in multi-armed bandits. In *International Conference on Machine Learning*, 2013.

Alexandra Carpentier and Michal Valko. Simple regret for infinitely many armed bandits. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1133–1141. PMLR, 2015.

Wenjing Chen and Victoria G. Crawford. Linear submodular maximization with bandit feedback. In *International Conference on Artificial Intelligence and Statistics*, 2025.

Wang Chi Cheung and Lixing Lyu. Leveraging (Biased) information: Multi-armed bandits with offline data. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *International conference on artificial intelligence and statistics*, 2011.

Devdatt P. Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009.

Eyal Even-Dar, Shie Mannor, Yishay Mansour, and Sridhar Mahadevan. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 2006.

Dylan Foster and Alexander Rakhlin. Beyond ucb: Optimal and efficient contextual bandits with regression oracles. In *International conference on machine learning*, 2020.

Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform Chernoff bounds via nonnegative supermartingales. *Probability Surveys*, 17:257–317, 2020.

Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. *The Annals of Statistics*, 49(2):1055–1080, 2021.

Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lil' UCB: An optimal exploration algorithm for multi-armed bandits. In *Conference on Learning Theory*, 2014.

Yassir Jedra and Alexandre Proutiere. Optimal best-arm identification in linear bandits. *Advances in Neural Information Processing Systems*, 2020.

Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, 2020.

Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. PAC subset selection in stochastic multi-armed bandits. In *International Conference on Machine Learning*, 2012.

Varun Kanade, H. Brendan McMahan, and Brent Bryan. Sleeping experts and bandits with stochastic action availability and adversarial rewards. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pp. 272–279, 2009.

Kellen Kanarios, Qining Zhang, and Lei Ying. Cost aware best arm identification. *arXiv preprint arXiv:2402.16710*, 2024.

Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of a/b testing. In *Conference on Learning Theory*, 2014.

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

Chinmaya Kausik, Kevin Tan, and Ambuj Tewari. Leveraging offline data in linear latent contextual bandits. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Robert Kleinberg, Alexandru Niculescu-Mizil, and Yogeshwer Sharma. Regret bounds for sleeping experts and bandits. *Machine Learning*, 80(2–3):245–272, 2010.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

Yishu Li, Xinyi Mao, Ying Yuan, Kyutae Sim, Ben Eisner, and David Held. Learn from what we HAVE: History-aware VErifier that reasons about past interactions online. In *Conference on Robot Learning*, 2025.

Zhaoqi Li, Lillian Ratliff, Kevin G Jamieson, Lalit Jain, et al. Instance-optimal pac algorithms for contextual bandits. In *Advances in Neural Information Processing Systems*, 2022.

Pierre Menard, Omar Darwiche Domingues, Anders Jonsson, Emilie Kaufmann, Edouard Leurent, and Michal Valko. Fast active learning for pure exploration in reinforcement learning. In *International Conference on Machine Learning*, 2021.

Adhyyan Narang, Andrew Wagenmaker, Lillian J Ratliff, and Kevin Jamieson. Sample complexity reduction via policy difference estimation in tabular reinforcement learning. In *Advances in Neural Information Processing Systems*, 2024.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search. In *Conference on Empirical Methods in Natural Language Processing*, 2023.

Alessio Russo, Ryan Welch, and Aldo Pacchiano. In-context learning for pure exploration. In *International Conference on Learning Representations*, 2026.

Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. In *Advances in Neural Information Processing Systems*, 2013.

Chengshuai Shi, Kun Yang, Zihan Chen, Jundong Li, Jing Yang, and Cong Shen. Efficient prompt optimization through the lens of best arm identification. In *Advances in Neural Information Processing Systems*, 2024.

David Simchi-Levi and Yunzong Xu. Bypassing the monster: A faster and simpler optimal algorithm for contextual bandits under realizability. *Mathematics of Operations Research*, 2022.

Jixian Wang and Ram Tiwari. Adaptive designs for best treatment identification with top-two thompson sampling and acceleration. *Pharmaceutical Statistics*, 22(6):1089–1103, 2023.

Yizao Wang, Jean-Yves Audibert, and Rémi Munos. Algorithms for infinitely many-armed bandits. In *Advances in Neural Information Processing Systems*, volume 21, pp. 1729–1736. Curran Associates, Inc., 2008.

Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang, and Linyi Yang. Cycleresearcher: Improving automated research via automated review. In *International Conference on Learning Representations*, 2025.

Zhihan Xiong, Romain Camilleri, Maryam Fazel, Lalit Jain, and Kevin Jamieson. A/b testing and best-arm identification for linear bandits with robustness to non-stationarity. In *International Conference on Artificial Intelligence and Statistics*, 2024.

Zelai Xu, Chao Yu, Fei Fang, Yu Wang, and Yi Wu. Language agents with reinforcement learning for strategic play in the werewolf game. In *International Conference on Machine Learning*, 2024.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint arXiv:2504.08066*, 2025.

Le Yang, Vincent Y. F. Tan, and Wang Chi Cheung. Best arm identification with possibly biased offline data. In *Proceedings of the Forty-first Conference on Uncertainty in Artificial Intelligence*, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models. In *International Conference on Machine Learning*, 2024.

Lucas Zimmer, Marius Lindauer, and Frank Hutter. Auto-PyTorch Tabular: Multi-fidelity metalearning for efficient and robust AutoDL. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(9):3079–3090, 2021.

APPENDIX CONTENTS

A Proof of the Main Result **15**
A.1 Validity and Geometry of the Mixture Refinement 15
A.2 Eluder Dimension and the Count of Wide Executions 16
A.3 Binomial Domination under Adaptive Coverage 17
A.4 A General Fixed-Budget Certificate 18
A.5 Proof of the Data-Dependent and Worst-Case Guarantees 19

B Additional Guarantees and Radius Choices **20**
B.1 Maximizing the Certified Success Probability 20
B.2 The General Tolerance Split 20
B.3 Sample Complexity as a Corollary 21
B.4 When Coverage Is Unknown 21
B.5 A General Known Sub-Gaussian Noise Proxy 21

C Proofs of the Lower Bounds and Output-Range Comparison **22**
C.1 Information Inequalities for Unrestricted Recommendations 22
C.2 The Hard Family, including Unrestricted Recommendations 22
C.3 Formal Minimax Comparison and Its Budget Consequence 23
C.4 The Necessary Qualification: an Instance Can Still Become Easier 24

D The Lower Bound with Only Two Candidates **24**

E Implementation and Interpretation **26**

F Experimental Details **27**
F.1 Synthetic Environment and Evaluation 27
F.2 LCBench Setup and Evaluation 29

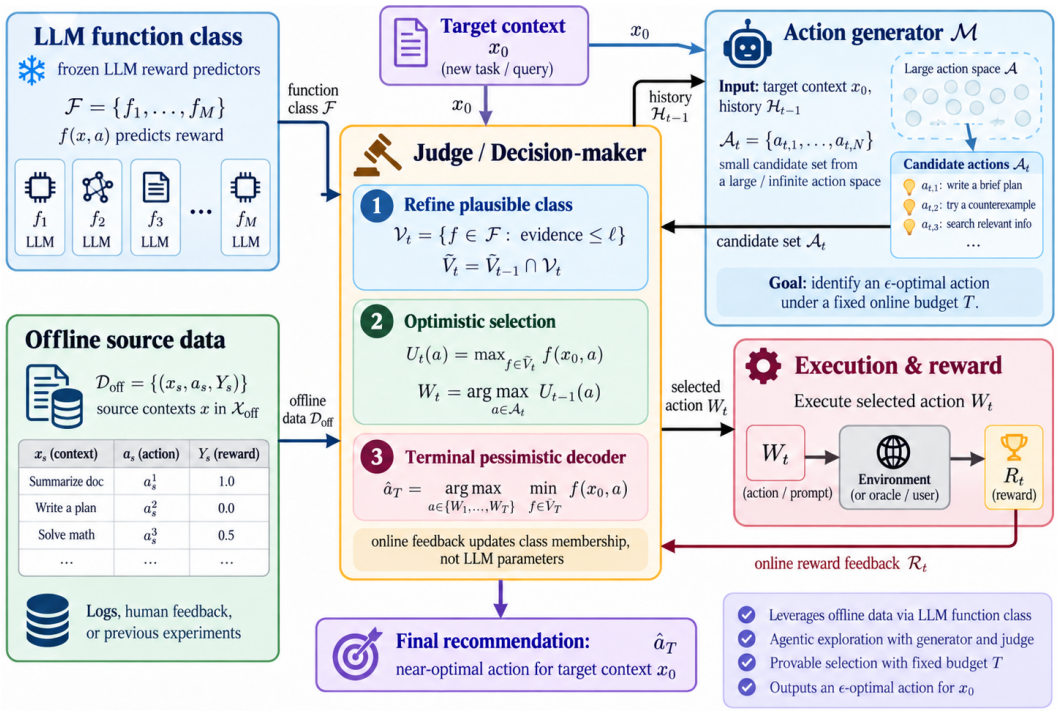


Figure 2: Framework of fixed-budget agentic pure exploration with judges. Offline reward data are used to localize a shared LLM reward-function class. At each online round, a black-box generator proposes a finite candidate set for the target context, the judge selects an action optimistically using the current plausible function class, and the observed reward is used to further refine this class. After a fixed budget of T interactions, a pessimistic decoder returns the final recommendation from the executed actions.

A Proof of the Main Result

For the appendices, write $M = |\mathcal{F}|$, $\mu(a) = f^\star(x_0, a)$, $\mu^\star = \sup_{a \in \mathcal{A}} \mu(a)$, and $\mathcal{E}_t = \{W_1, \ldots, W_t\}$. We write $e_T(\epsilon) = \mathbb{P}(\hat{a}_T \notin G_\epsilon(x_0))$. Unless otherwise stated, fix a tolerance $r \in [0, \epsilon)$, put $\alpha = \epsilon - r$, and use a deterministic coverage lower bound $p \in (0, 1]$ satisfying $\mathbb{P}_t(A_t \cap G_r(x_0) \neq \varnothing) \ge p$. The radius $\ell > 0$ is fixed before localization. These conventions permit the sharper arbitrary-radius result as well as both guarantees in Theorem 3.1. Throughout, $\mathcal{F}$ and the target context are fixed before localization, and reward noise is conditionally mean-zero and 1-sub-Gaussian given the pre-reward information, as specified in Section 3.2.

For integer k, define the comparison tail used only in the appendix

$$B_{T,p}(k) = \mathbb{P}\left(\mathrm{Bin}(T, p) \le k\right),$$

with $B_{T,p}(k) = 0$ for $k < 0$ and $B_{T,p}(k) = 1$ for $k \ge T$. For $p = 1$, the comparison variable equals T deterministically. The actual generated lists need not be independent.

A.1 Validity and Geometry of the Mixture Refinement

For convenience, define the pairwise log evidence and its aggregate by

$$H_t(f, g) = \frac{S_t(f) - S_t(g) + \frac{1}{2} D_t(f, g)}{4}, \qquad \Phi_t(f) = \log \sum_{g \in \mathcal{F}} e^{H_t(f,g)}. \tag{11}$$

Thus $\mathcal{V}_t = \{f \in \mathcal{F} : \Phi_t(f) \le \ell\}$ and $\widetilde{\mathcal{V}}_t = \bigcap_{s=0}^{t} \mathcal{V}_s$. We apply the standard mixture-supermartingale method (Howard et al., 2021); the calculation below identifies its consequences for the specific filter in Eq. (2).

Proposition A.1 (Truth retention and surviving-pair disagreement). *Under Assumption 2.1 and the pre-reward conditional noise condition, with $\mathcal{F}$ fixed before localization, for every predictable query policy,*

$$\mathbb{P}\left(f^\star \in \widetilde{\mathcal{V}}_t \text{ for all } t \geq 0\right) \geq 1 - Me^{-\ell}. \tag{12}$$

Moreover, on every *sample path, $f, g \in \mathcal{V}_t$ implies*

$$D_t(f, g) \leq 8\ell. \tag{13}$$

On a fixed transcript, $\mathcal{V}_t$ is a subset of the loss-only square-loss set $\{f : S_t(f) - \min_g S_t(g) \leq 4\ell\}$; the same inclusion holds after taking nested intersections.

Proof of Proposition A.1. Condition first on any independent auxiliary data used to construct $\mathcal{F}$, and concatenate the offline and online observations into (z_n, Y_n), where z_n is a context–action pair and $Y_n = f^\star(z_n) + \xi_n$. Each input is chosen before its reward, and ξ_n is conditionally mean-zero and 1-sub-Gaussian given that pre-reward information. Thus inputs may depend on earlier rewards but not their own reward noise. Start the following process *before* the first localization reward, with $\mathcal{F}$ and ℓ fixed. For each fixed $g \in \mathcal{F}$, set $d_n = g(z_n) - f^\star(z_n)$. Under Assumption 2.1,

$$Z_n(g) = \exp\left(\frac{1}{2}\sum_{j=1}^{n} \xi_j d_j - \frac{1}{8}\sum_{j=1}^{n} d_j^2\right) \tag{14}$$

is a nonnegative supermartingale starting at one. Conditional on the information before the reward, d_n is known and the expected multiplicative increment is at most $\exp(d_n^2/8 - d_n^2/8) = 1$.

At the index $n = n_0 + t$, expanding the losses gives

$$S_t(f^\star) - S_t(g) = 2\sum_{j=1}^{n_0+t} \xi_j d_j - \sum_{j=1}^{n_0+t} d_j^2,$$

so $e^{H_t(f^\star, g)} = Z_{n_0+t}(g)$. Their fixed average $\overline{Z}_n = M^{-1}\sum_{g\in\mathcal{F}} Z_n(g)$ is also a nonnegative supermartingale starting at one. If $f^\star$ is excluded from any raw set, then $\overline{Z}_{n_0+t} > e^\ell/M$ for some t. By Ville's inequality (Howard et al., 2020, Lemma 1), this has probability at most $Me^{-\ell}$. On the complement, truth belongs to every raw set and hence every nested intersection.

For the deterministic disagreement bound, $f \in \mathcal{V}_t$ implies $e^{H_t(f,g)} \leq \sum_{h\in\mathcal{F}} e^{H_t(f,h)} \leq e^\ell$ for every g. Therefore $H_t(f, g) \leq \ell$. If $g \in \mathcal{V}_t$ as well, $H_t(g, f) \leq \ell$. Adding and using symmetry of D_t gives

$$\frac{D_t(f, g)}{4} = H_t(f, g) + H_t(g, f) \leq 2\ell,$$

which is Eq. (13).

Finally, $D_t(f, g) \geq 0$ and $H_t(f, g) \leq \ell$ imply $S_t(f) - S_t(g) \leq 4\ell$ for all $g \in \mathcal{F}$. Minimizing over g proves the inclusion in the loss-only set. Taking intersections preserves inclusion on the same data. On the truth-retention event, $L_t(a) \leq \mu(a) \leq U_t(a)$ for every action and every t. □

The pairwise bound is deterministic whenever both models survive. On identical labeled data, the mixture sets are contained in the loss-only sets at the same radius.

A.2 Eluder Dimension and the Count of Wide Executions

Restricting a function class cannot increase $d_Q(\alpha)$, directly from its definition in Section 3.2. The following is the fixed-scale specialization of the width-counting argument of Russo & Van Roy (2013, Proposition 3 and its proof). We record the pairwise-energy form needed here, with its integer rounding.

Lemma A.2 (Exact-scale width count). *Let Q be a finite function class, let $Q_{t-1} \subseteq Q$ be nonempty, and let $a_1, \ldots, a_T$ be actions. Suppose $B > 0$ and every $f, g \in Q_{t-1}$ satisfy*

$$\sum_{i<t} \left(f(x_0, a_i) - g(x_0, a_i)\right)^2 \leq B. \tag{15}$$

Then, for $w_{Q_{t-1}}(a) = \max_{f,g \in Q_{t-1}} |f(x_0, a) - g(x_0, a)|$,

$$\#\left\{t \le T : w_{Q_{t-1}}(a_t) > \alpha\right\} \le d_Q(\alpha) \left\lceil \frac{B}{\alpha^2} \right\rceil.$$

The statement is deterministic; the function classes and actions may be chosen retrospectively on a sample path.

Fixed-scale partition argument. Process only the indices with width greater than α, in chronological order. Partition their actions into subsequences, each α-independent with respect to Q. Append a new wide action to an existing subsequence whenever it is α-independent of that subsequence; otherwise start a new subsequence. Each subsequence has length at most $d_Q(\alpha)$.

Suppose a new subsequence is needed when k subsequences already exist. A pair in $Q_{t-1} \subseteq Q$ witnessing the current width differs by more than α at the new action. Since that action is not independent of any existing subsequence, this same pair has squared disagreement strictly greater than α^2 on each of them. Their indices are disjoint and precede t, so Eq. (15) gives $k\alpha^2 < B$. Consequently the number of subsequences after adding the new one is at most $\lceil B/\alpha^2 \rceil$. The first subsequence is covered because $B > 0$; if $d_Q(\alpha) = 0$, no wide action exists. Multiplying the number of subsequences by their maximum length proves the claim. This uses the exact scale in Section 3.2, rather than a scale-monotone variant of the dimension. □

For nonempty nested classes, write $w_t(a) = U_t(a) - L_t(a)$. On the truth-retention event, apply the lemma with the realized fixed function class $Q = \mathcal{V}_0$, $Q_{t-1} = \widetilde{\mathcal{V}}_{t-1}$, $a_t = W_t$, and $B = 8\ell > 0$. The offline terms in D_{t-1} are nonnegative, so Eq. (13) implies Eq. (15). Define $d_0 = d_{\mathcal{V}_0}(\alpha)$ and $K_0 = d_0 \lceil 8\ell/\alpha^2 \rceil$. Therefore, pathwise,

$$\#\left\{t \le T : w_{t-1}(W_t) > \alpha\right\} \le K_0. \tag{16}$$

The lemma uses the *pre-query* set: the reward at round t is not counted as evidence before selecting W_t.

A.3 Binomial Domination under Adaptive Coverage

Lemma A.3 (Conditional coverage comparison)**.** *Let* $C_t = \mathbb{I}\{A_t \cap G_r(x_0) \neq \varnothing\}$ *and suppose Assumption 2.2. For any* $\mathcal{H}_0$*-measurable nonnegative integer* K*,*

$$\mathbb{P}\left(\sum_{t=1}^{T} C_t \le K \mid \mathcal{H}_0\right) \le B_{T,p}(K) \quad \textit{almost surely.}$$

For any deterministic integer $k \ge 0$*,*

$$B_{T,p}(k) \le \exp\left(-\frac{(pT - k)_+^2}{2pT}\right).$$

Proof. Condition on $\mathcal{H}_0$ and its value $K = k$. Write $A_t^{\mathrm{cov}} = \sum_{i=1}^{t} C_i$ and let $F_j(s) = \mathbb{P}(\mathrm{Bin}(j, p) \le k - s)$. This is nonincreasing in s. If $q_t = \mathbb{E}[C_t \mid \mathcal{H}_{t-1}] \ge p$, then

$$\begin{aligned}
\mathbb{E}\left[F_{T-t}(A_t^{\mathrm{cov}}) \mid \mathcal{H}_{t-1}\right] &= (1 - q_t) F_{T-t}(A_{t-1}^{\mathrm{cov}}) + q_t F_{T-t}(A_{t-1}^{\mathrm{cov}} + 1) \\
&\le (1 - p) F_{T-t}(A_{t-1}^{\mathrm{cov}}) + p F_{T-t}(A_{t-1}^{\mathrm{cov}} + 1) \\
&= F_{T-t+1}(A_{t-1}^{\mathrm{cov}}).
\end{aligned}$$

Iterate from $t = T$ to 1, using $F_0(A_T^{\mathrm{cov}}) = \mathbb{I}\{A_T^{\mathrm{cov}} \le k\}$ and $F_T(0) = B_{T,p}(k)$. This argument uses only conditional Bernoulli probabilities and allows arbitrary history dependence.

The binomial tail is the multiplicative Chernoff bound (Dubhashi & Panconesi, 2009, Theorem 1.1), applied with relative deviation $1 - k/(pT)$ when $0 < k < pT$. The case $k = 0$ follows by a limit, $k \ge pT$ is trivial, and $p = 1$ follows directly from the deterministic binomial variable. □

Time-varying coverage. The analysis extends directly to round-dependent coverage bounds $p_t = 1 - \beta_t(r, x_0)$. Let

$$\Lambda_T(r) := \sum_{t=1}^{T} p_t$$

and let

$$B_{T,\mathbf{p}}(k) := \mathbb{P}\left(\sum_{t=1}^{T} Z_t \le k\right), \qquad Z_t \overset{\text{ind}}{\sim} \text{Bernoulli}(p_t),$$

denote the corresponding Poisson-binomial lower tail. The conclusion of Lemma A.3 remains valid with $B_{T,p}(K)$ replaced by $B_{T,\mathbf{p}}(K)$. Moreover,

$$B_{T,\mathbf{p}}(K) \le \exp\left(-\frac{(\Lambda_T(r) - K)_+^2}{2\Lambda_T(r)}\right),$$

where the right-hand side is defined as one when $\Lambda_T(r) = 0$.

Consequently, Theorem A.4 and Theorem 3.1 remain valid after replacing $B_{T,p}$ by $B_{T,\mathbf{p}}$ and every occurrence of pT by $\Lambda_T(r)$. In particular, for $d \ge \max\{1, d_{\mathcal{F}}(\alpha)\}$ and

$$\ell_T = \frac{\Lambda_T(r)\alpha^2}{32d},$$

the analogue of B.2 is

$$\mathbb{P}(\widehat{a}_T \in G_\epsilon(x_0)) \ge \left(1 - (M+1)\exp\left\{-\frac{\Lambda_T(r)\alpha^2}{32d}\right\}\right)_+.$$

The algorithm itself is unchanged.

A.4 A General Fixed-Budget Certificate

Theorem A.4 (Mixture-filter fixed-budget bound)**.** *Suppose Assumption 2.1 and Assumption 2.2. Fix an integer $T \ge 1$, $\epsilon \in (0,1)$, $r \in [0, \epsilon)$, $\alpha = \epsilon - r$, and a radius $\ell > 0$ fixed before localization. Let $p > 0$ be a deterministic coverage lower bound at tolerance r. Run Algorithm 1. Set*

$$d_0 = d_{\mathcal{V}_0}(\alpha), \qquad K_0 = d_0 \left\lceil \frac{8\ell}{\alpha^2} \right\rceil.$$

Then

$$\mathbb{P}\left(\widehat{a}_T \in G_\epsilon(x_0)\right) \ge \left(1 - Me^{-\ell} - \mathbb{E}\left[B_{T,p}(K_0)\right]\right)_+. \tag{17}$$

In particular, if the deterministic integer $d \ge d_{\mathcal{F}}(\alpha)$ and $K_d(\ell) = d\lceil 8\ell/\alpha^2 \rceil$, then

$$\mathbb{P}\left(\widehat{a}_T \in G_\epsilon(x_0)\right) \ge \left(1 - Me^{-\ell} - B_{T,p}\left(K_d(\ell)\right)\right)_+,$$

$$\mathbb{P}\left(\widehat{a}_T \in G_\epsilon(x_0)\right) \ge \left(1 - Me^{-\ell} - \exp\left(-\frac{(pT - K_d(\ell))_+^2}{2pT}\right)\right)_+. \tag{18}$$

Any deterministic almost-sure upper bound on d_0 for the chosen radius can replace $d_{\mathcal{F}}(\alpha)$.

Proof of Theorem A.4. Let $\mathcal{C}$ be the truth-retention event in Eq. (12). On $\mathcal{C}$, suppose $\sum_{t=1}^{T} C_t > K_0$. By Eq. (16), some covered round t has $w_{t-1}(W_t) \le \alpha$. Optimism and truth membership imply

$$U_{t-1}(W_t) = \max_{a \in A_t} U_{t-1}(a) \ge \max_{a \in A_t} \mu(a) \ge \mu^\star - r.$$

Consequently $L_{t-1}(W_t) \ge \mu^\star - r - \alpha = \mu^\star - \epsilon$. Since $W_t \in \mathcal{E}_T$ and the confidence sets are nested,

$$\mu(\widehat{a}_T) \ge L_T(\widehat{a}_T) = \max_{a \in \mathcal{E}_T} L_T(a) \ge L_T(W_t) \ge L_{t-1}(W_t) \ge \mu^\star - \epsilon.$$

Thus failure is contained in $\mathcal{C}^c \cup \{\sum_t C_t \le K_0\}$. The random variable K_0 is $\mathcal{H}_0$-measurable. Apply Proposition A.1, then Lemma A.3 conditionally on $\mathcal{H}_0$, and average:

$$e_T(\epsilon) \le Me^{-\ell} + \mathbb{E}\left[B_{T,p}(K_0)\right].$$

This proves Eq. (17). If $d_0 \le d$ almost surely, then $K_0 \le K_d(\ell)$, and monotonicity of the binomial CDF gives the deterministic binomial bound. The tail estimate in Lemma A.3 gives Eq. (18). When the right-hand error allowance exceeds one, clipping the success lower bound at zero is valid. □

Random localization. The expectation in Eq. (17) is essential: d_0 is random. The proof conditions the coverage comparison on $\mathcal{H}_0$, not the truth-retention event. The cap $\binom{|\mathcal{V}_0|}{2}$ can replace d_0 *inside* that expectation. A terminal dimension alone does not account for the observations spent eliminating earlier uncertainty. Selecting the radius retrospectively from the same localization rewards is not covered by this theorem.

A.5 Proof of the Data-Dependent and Worst-Case Guarantees

Proof of Theorem 3.1. Take $r = \alpha = \epsilon/2$ and $p = 1 - \beta(\epsilon/2, x_0)$ in Theorem A.4. Its random threshold is $K_0 = d_{\mathcal{V}_0}(\epsilon/2)\lceil 32\ell/\epsilon^2 \rceil$. Applying the exponential comparison from Lemma A.3 pointwise to K_0 in Eq. (17) gives

$$e_T(\epsilon) \le Me^{-\ell} + \mathbb{E}\left[\exp\left\{-\frac{(pT - K_0)_+^2}{2pT}\right\}\right],$$

which proves Eq. (6). Only the coverage comparison is conditioned on $\mathcal{H}_0$: K_0 is then known, while truth-retention failure is bounded unconditionally. No independence between K_0 and the generated lists is assumed, and truth-retention failure is counted only once.

For Eq. (7), use exactly the stated radius $\ell = pT\epsilon^2/(128\max\{1, d_{\mathcal{F}}(\epsilon/2)\})$. Because $\mathcal{V}_0 \subseteq \mathcal{F}$ on every sample path, if $pT \ge 4\max\{1, d_{\mathcal{F}}(\epsilon/2)\}$ then

$$K_0 \le \max\{1, d_{\mathcal{F}}(\epsilon/2)\}\left\lceil \frac{pT}{4\max\{1, d_{\mathcal{F}}(\epsilon/2)\}} \right\rceil \le \frac{pT}{4} + \max\{1, d_{\mathcal{F}}(\epsilon/2)\} \le \frac{pT}{2}.$$

Consequently, Eq. (6) bounds failure by $Me^{-\ell} + e^{-pT/8} \le (M+1)e^{-\ell}$, since $\ell \le pT/128$. If $pT < 4\max\{1, d_{\mathcal{F}}(\epsilon/2)\}$, then $\ell < \epsilon^2/32 < 1/32$ and $(M+1)e^{-\ell} > 1$, so the lower bound in Eq. (7) is zero and remains valid. This proves the worst-case statement for every positive integer budget.

For the sample-complexity consequence, every integer budget satisfying

$$T \ge \left\lceil \frac{128\max\{1, d_{\mathcal{F}}(\epsilon/2)\}}{p\epsilon^2} \log\frac{M+1}{\delta} \right\rceil$$

makes $\ell \ge \log((M+1)/\delta)$ and hence $(M+1)e^{-\ell} \le \delta$. Integer rounding is absorbed by the stated order since $\max\{1, d_{\mathcal{F}}(\epsilon/2)\} \ge 1$, $p \le 1$, $\epsilon < 1$, and $\log((M+1)/\delta) \ge \log 2$. The radius is fixed before localization, not chosen from the realized $d_{\mathcal{V}_0}$. The general and worst-case statements concern their respective radius choices, not simultaneous calibrations of one run.

If $K_0 = 0$, the sharper binomial contribution in Eq. (17) is $(1-p)^T$, including zero when $p = 1$; the exponential upper bound used in the main theorem is valid but may be looser. If $K_0 \ge pT$, its displayed exponential contribution is one. Empty-class paths are charged to truth-retention failure despite the convention $d_\varnothing = 0$. These observations also cover zero full-class dimension and perfect coverage. □

When offline data help. For fixed T, ℓ, and coverage, the exponential term in Eq. (6) is nondecreasing in $d_{\mathcal{V}_0}(\epsilon/2)$. This compares certificates, not the actual success probabilities on different noisy transcripts. In particular, $d_{\mathcal{V}_0}(\epsilon/2) = 0$ means pairwise prediction differences at most $\epsilon/2$ on every target action. On the truth-retention event, no execution then has width greater than $\epsilon/2$, so a covered round suffices. An empty class can also have dimension zero, but its paths are already charged to truth-retention failure.

If all functions agree on every offline input, then $S_0(f)$ is the same for all f and $D_0(f, g) = 0$ for all pairs, independently of the observed rewards. Hence $\Phi_0(f) = \log M$ and $\mathcal{V}_0 = \mathcal{F}$ whenever $\ell \ge \log M$, so no residual-dimension reduction occurs. If $\ell < \log M$, the initial class is empty instead; the general bound is trivial because $Me^{-\ell} > 1$. This is not useful localization. Thus more observations alone do not guarantee improvement, and no i.i.d. input assumption is required for either conclusion.

B Additional Guarantees and Radius Choices

We retain the conventions $M = |\mathcal{F}|$, $\alpha = \epsilon - r$, and coverage lower bound p from Appendix A. This appendix gives results omitted from the main text: exact optimization of the certified success probability, the general tolerance split, sample-complexity corollaries, a coverage-free radius, and a general known noise proxy. Each radius below is selected before localization and kept fixed throughout its run.

B.1 Maximizing the Certified Success Probability

Theorem B.1 (Finite radius optimization)**.** *Suppose p, α, and a deterministic integer $d \geq \max\{1, d_{\mathcal{F}}(\alpha)\}$ are supplied. Define*

$$J_T = \left\lfloor \frac{T-1}{d} \right\rfloor, \qquad \ell_j = \frac{j\alpha^2}{8}, \qquad b_j = \left(1 - Me^{-j\alpha^2/8} - B_{T,p}(jd)\right)_+ .$$

If $\max_{1\leq j\leq J_T} b_j > 0$, choose a maximizing j before localization and use $\ell = \ell_j$. If this maximum is zero or the index set is empty, use the default $\ell = \log\left(M(T+1)^2\right)$. Then the resulting algorithm satisfies

$$\mathbb{P}\left(\widehat{a}_T \in G_\epsilon(x_0)\right) \geq \sup_{\ell>0} \left(1 - Me^{-\ell} - B_{T,p}\left(K_d(\ell)\right)\right)_+ = \max\left(\{0\} \cup \{b_j : 1 \leq j \leq J_T\}\right).$$

Proof. On $((j-1)\alpha^2/8, j\alpha^2/8]$, $K_d(\ell) = jd$, and the certificate is nondecreasing in ℓ. Its best point in that interval is the right endpoint. If $jd \geq T$, the binomial tail is one and the certificate is zero. If all certificates vanish, any radius attains the zero supremum; the default avoids an unnecessarily empty initial function class in the absence of localization data. □

Only scalar binomial-CDF evaluations are needed. This maximizes the stated guarantee over the radius family; it does not prove optimality of the actual error probability over all algorithms. A common deterministic bound d must apply to all candidate radii. No target error δ is an input to this calibration.

B.2 The General Tolerance Split

Corollary B.2 (General horizon-calibrated success probability)**.** *Under the assumptions of Theorem A.4, take a deterministic integer $d \geq \max\{1, d_{\mathcal{F}}(\alpha)\}$. With*

$$\ell_T = \frac{pT\alpha^2}{32d},$$

Algorithm 1 satisfies, for every $T \geq 1$,

$$\mathbb{P}\left(\widehat{a}_T \in G_\epsilon(x_0)\right) \geq \left(1 - (M+1)\exp\left(-\frac{pT\alpha^2}{32d}\right)\right)_+ . \tag{19}$$

The optimized calibration in Theorem B.1 satisfies the same bound. Taking $r = \epsilon/2$ gives the full-class exponential bound, without requiring i.i.d. offline inputs.

Proof. If $pT \geq 4d$, then $K_d(\ell_T) \leq pT/4 + d \leq pT/2$. By Eq. (18), the error is at most $Me^{-\ell_T} + e^{-pT/8} \leq (M+1)e^{-\ell_T}$, because $\alpha < 1$ and $d \geq 1$. If $pT < 4d$, then $\ell_T < 1/8$ and $(M+1)e^{-\ell_T} > 1$, so the clipped claim is zero. Optimizing the exact certificate is at least as good as this particular choice. □

When $\ell_T \leq \log M$, the displayed lower bound is necessarily zero. In that regime the default radius in Theorem B.1 can be used instead without changing Eq. (19). The theorem concerns a horizon-specific algorithm.

B.3 Sample Complexity as a Corollary

Corollary B.3 (A sufficient budget for success at least $1-\delta$)**.** *Let $\delta \in (0,1)$ and let $d \geq d_{\mathcal{F}}(\alpha)$ be a deterministic nonnegative integer. Set*

$$\ell_\delta = \log \frac{2M}{\delta}, \qquad K_\delta = d\left\lceil \frac{8}{\alpha^2} \log \frac{2M}{\delta} \right\rceil, \qquad T_\delta = \left\lceil \frac{\max\{2K_\delta,\, 8\log(2/\delta)\}}{p} \right\rceil.$$

Every integer $T \geq T_\delta$ gives success at least $1-\delta$ with radius ℓ_δ. If $d \geq 1$, the optimized calibration of Theorem B.1 has the same implication. Alternatively, the simple horizon calibration is sufficient whenever

$$T \geq \left\lceil \frac{32d}{p\alpha^2} \log \frac{M+1}{\delta} \right\rceil \qquad (d \geq 1).$$

Taking $r = \epsilon/2$ yields the online sample-complexity order

$$\mathcal{O}\left(\frac{d\log(2M/\delta)/\epsilon^2 + \log(2/\delta) + 1}{1 - \beta(\epsilon/2, x_0)} \right).$$

Proof. The prescribed budget makes $K_\delta \leq pT/2$, $Me^{-\ell_\delta} = \delta/2$, and $e^{-pT/8} \leq \delta/2$. Apply Eq. (18). The optimized exact certificate is no worse. Inverting Eq. (19) proves the displayed horizon-calibrated budget; the order statement follows from $\lceil z \rceil \leq z+1$. □

B.4 When Coverage Is Unknown

For a fixed prescribed ℓ, the selector and decoder do not require p, d, or r. Those quantities are needed only for the optimized calibrations and their numerical certificates. The fully specified choice $\ell = \log\left(M(T+1)^2\right)$ requires none of them and obeys

$$e_T(\epsilon) \leq \min\left\{ 1, \frac{1}{(T+1)^2} + B_{T,p}\left(d\left\lceil \frac{8\log\left(M(T+1)^2\right)}{\alpha^2} \right\rceil \right) \right\}.$$

Here d is any deterministic nonnegative integer satisfying $d \geq d_{\mathcal{F}}(\alpha)$; p, d enter only the analysis.

B.5 A General Known Sub-Gaussian Noise Proxy

Suppose instead that the known proxy $\sigma^2 > 0$ satisfies $\mathbb{E}[e^{u\xi} \mid \text{pre-reward information}] \leq e^{\sigma^2 u^2/2}$. Replace Eq. (11) by

$$H_t(f,g) = \frac{S_t(f) - S_t(g) + \frac{1}{2}D_t(f,g)}{4\sigma^2}.$$

The same mixture and nesting give truth-retention failure at most $Me^{-\ell}$ and the pathwise bound $D_t(f,g) \leq 8\sigma^2\ell$. In Theorem A.4 and the sample-complexity argument of Corollary B.3, replace 8ℓ in the width count by $8\sigma^2\ell$, and use grid endpoints $\ell_j = j\alpha^2/(8\sigma^2)$. In particular, the sufficient budget order becomes

$$\mathcal{O}\left(\frac{\sigma^2 d\log(2M/\delta)/\alpha^2 + d + \log(2/\delta) + 1}{p} \right).$$

The additive terms cannot be discarded for very small noise.

For a simple exponential bound valid at *every* noise level, put $v = \max\{\sigma^2, \alpha^2\}$ and $\ell_T = pT\alpha^2/(32dv)$ for $d \geq 1$. Then

$$\mathbb{P}\left(\widehat{a}_T \in G_\epsilon(x_0)\right) \geq \left(1 - (M+1)\exp\left(-\frac{pT\alpha^2}{32dv}\right)\right)_+.$$

To verify these claims, the pairwise supermartingale in Eq. (14) becomes $\exp\{\sum_j \xi_j d_j/(2\sigma^2) - \sum_j d_j^2/(8\sigma^2)\}$; the other arguments are unchanged. For the displayed exponential bound, if $pT \geq 4d$, the count is at most $pT\sigma^2/(4v) + d \leq pT/2$, and $\ell_T \leq pT/(32d) \leq pT/8$. If $pT < 4d$, then $\ell_T < 1/8$ and the lower bound is zero. At $\sigma^2 = 1$, $\alpha < 1$ gives $v = 1$ and recovers Corollary B.2, and its full-class specialization at $r = \epsilon/2$.

C Proofs of the Lower Bounds and Output-Range Comparison

This appendix proves Theorem 4.1 and Theorem 4.2. The lower-bound argument permits any measurable terminal recommendation from $\mathcal{A}$, although the reward queries must remain in the current generated list. We use standard change-of-measure tools (Kaufmann et al., 2016; Lattimore & Szepesvári, 2020) and verify below how the generated-list protocol enters the transcript divergence.

For this appendix, write $p = 1 - \beta(\epsilon/2, x_0)$ and $e_T^{\max} = \max_{f \in \mathcal{F}} \mathbb{P}_f(\widehat{a}_T \notin G_\epsilon^f(x_0))$.

C.1 Information Inequalities for Unrestricted Recommendations

For $u, v \in (0,1)$, write $\mathrm{kl}(u,v) = u\log(u/v) + (1-u)\log((1-u)/(1-v))$, with its usual limiting values on the boundary. For probability distributions P, Q, write $\mathrm{KL}(P\|Q) = \mathbb{E}_P[\log(dP/dQ)]$, taking the value $+\infty$ when absolute continuity fails.

Lemma C.1 (Coarsening, testing, and adaptive Gaussian information)**.** *For probability distributions P, Q and a measurable event B, data processing and the Bretagnolle–Huber inequality (Lattimore & Szepesvári, 2020, Exercise 14.10 and Theorem 14.2) give*

$$\mathrm{KL}(P\|Q) \geq \mathrm{kl}\left(P(B), Q(B)\right), \tag{20}$$

$$P(B^c) + Q(B) \geq \tfrac{1}{2}\exp\left(-\mathrm{KL}(P\|Q)\right). \tag{21}$$

For two reward functions f, g with independent variance-one Gaussian noise, identical offline distributions, and the same conditional generator and algorithmic kernels, the horizon-T transcript distributions satisfy

$$\mathrm{KL}(P_f^T\|P_g^T) = \frac{1}{2}\mathbb{E}_f\left[\sum_{t=1}^{T}\left(f(x_0,W_t) - g(x_0,W_t)\right)^2\right]. \tag{22}$$

Including any randomized data-dependent recommendation from the entire action space does not change this identity.

Application to the generated-list protocol. The first two inequalities are the cited standard results. For Eq. (22), apply divergence decomposition (Lattimore & Szepesvári, 2020, Lemma 15.1 and Exercise 15.8). Conditional on the same history, the list-generation, action-selection, and terminal-recommendation kernels are identical under f and g; their conditional divergences are zero. The offline contribution is zero by assumption. Only the Gaussian reward term remains, and $\mathrm{KL}(\mathcal{N}(u,1)\|\mathcal{N}(v,1)) = (u-v)^2/2$ yields the identity. Appending a full-space recommendation therefore adds no information, even when its marginal distribution differs between the two models. □

C.2 The Hard Family, including Unrestricted Recommendations

Proof of Theorem 4.1, including full-space output. **Construction.** Take $b, a_1, \ldots, a_d$ together with enough distinct zero-mean dummy actions to form size-N lists. Suppress x_0 and define $\mathcal{F} = \{f_0, f_1, \ldots, f_d\}$ by

$$f_0(b) = \frac{1}{2}, \qquad f_0(a_j) = \frac{1}{2} - 2\epsilon, \qquad f_i(a_i) = \frac{1}{2} + 2\epsilon,$$

with $f_i = f_0$ everywhere except at a_i. All means lie in $[0,1]$. Independently each round, return a list containing $b, a_1, \ldots, a_d$, padded by dummies, with probability p; otherwise return only dummies. This one generator is common to all models and independent of the learner's choices. For any prescribed n_0, offline inputs may all equal one fixed context–action pair at a source context where every model has mean $1/2$, with independent unit-variance Gaussian reward noise. These observations have identical distributions under all models. Also, $S_0(f)$ is identical for all f and $D_0(f,g) = 0$, so $\Phi_0(f) = \log M$ and $\mathcal{V}_0 = \mathcal{F}$ whenever $\ell \geq \log M$. Its residual dimension is then $d_{\mathcal{V}_0}(\epsilon/2) = d$, the uninformative-offline case described after Theorem 3.1.

Coverage and dimension. Only b is ϵ-optimal under f_0, and only a_i is ϵ-optimal under f_i. The second-best gap is 2ϵ, and dummy actions are worse. Thus informative lists contain an $\epsilon/2$-optimal action in every model, whereas dummy lists do not; conditional coverage is exactly p.

The sequence $a_1, \ldots, a_d$ is independent at scale $\epsilon/2$: the pair f_0, f_i witnesses position i and agrees on its predecessors. Conversely, all model disagreements occur on these d coordinates. A repeated coordinate cannot be independent, because any witnessing pair disagrees there by 4ϵ, which already exceeds the scale at its earlier appearance. Hence $d_{\mathcal{F}}(\epsilon/2) = d$.

Information available from the queries. Let $N_i(T) = \sum_{t=1}^{T} \mathbb{I}\{W_t = a_i\}$. At most one such coordinate can be queried on each informative round and none on a dummy round. Therefore

$$\sum_{i=1}^{d} \mathbb{E}_0 N_i(T) \le pT. \tag{23}$$

Some i satisfies $\mathbb{E}_0 N_i(T) \le pT/d$. Let P_0^T, P_i^T be the distributions of the entire transcripts, including the unrestricted recommendation. The Gaussian transcript identity Eq. (22) gives

$$\mathrm{KL}(P_0^T \| P_i^T) = 8\epsilon^2 \mathbb{E}_0 N_i(T) \le \frac{8pT\epsilon^2}{d}. \tag{24}$$

Indeed, the two mean functions differ by 4ϵ only at a_i; generator, selection, and output kernels add no likelihood-ratio contribution.

Every terminal decoder faces the same testing problem. Take $B = \{\widehat{a}_T = b\}$. Under f_0, B^c is exactly failure; under f_i, B implies failure. The testing inequality Eq. (21) yields

$$2e_T^{\max} \ge P_0^T(B^c) + P_i^T(B) \ge \tfrac{1}{2} \exp\left(-\mathrm{KL}(P_0^T \| P_i^T)\right).$$

Together with Eq. (24), this gives $e_T^{\max} \ge \frac{1}{4} e^{-8pT\epsilon^2/d}$, which implies Eq. (8). No step constrained the recommendation to observed actions. The cited testing inequality and the protocol-specific divergence calculation are collected in Appendix C.1. □

Corollary C.2 (Necessary sample complexity with unrestricted output)**.** *For $0 < \delta \le 1/4$, uniform success at least $1 - \delta$ on the family in Theorem 4.1 requires*

$$T \ge \frac{d\ \mathrm{kl}(1-\delta, \delta)}{8p\epsilon^2} \ge \frac{d}{16p\epsilon^2} \log \frac{1}{2\delta}, \tag{25}$$

where $\mathrm{kl}(u, v) = u \log(u/v) + (1-u) \log((1-u)/(1-v))$.

Proof. For every i, correctness gives $P_0^T(B) \ge 1 - \delta$ and $P_i^T(B) \le \delta$. The data-processing inequality Eq. (20) implies

$$8\epsilon^2 \mathbb{E}_0 N_i(T) \ge \mathrm{kl}(1-\delta, \delta).$$

Here binary divergence is increasing in its first argument and decreasing in its second when the first is larger; this follows by differentiating its formula. Sum over i and use Eq. (23). Finally, $\mathrm{kl}(1-\delta, \delta) = (1-2\delta)\log((1-\delta)/\delta) \ge \frac{1}{2}\log(1/(2\delta))$ for $\delta \le 1/4$. □

The direct exponential lower bound implies another valid corollary, $T \ge [d/(8p\epsilon^2)] \log(1/(4\delta))$ for $\delta < 1/4$. The binary-divergence statement above is proved separately to avoid losing its sharper confidence dependence.

C.3 Formal Minimax Comparison and Its Budget Consequence

Fix $d \ge 1$, $N \ge d+1$, $0 < \epsilon \le 1/8$, and $p \in (0, 1]$. Let $\mathfrak{I}$ be the class of finite-action instances satisfying the assumptions with function class size at most $d+1$, exact-scale dimension at $\epsilon/2$ at most d, size-N lists, and conditional coverage at least p at that tolerance. Each instance includes the true member of its given function class, generator, and localization-data distribution. Function classes are supplied to the learner; their true member is not. The infima below allow knowledge of the structural parameters and arbitrary computation, and both obey $W_t \in A_t$:

$$\mathfrak{e}_T^{\mathcal{E}} = \inf_{\text{executed-output algorithms}} \sup_{I \in \mathfrak{I}} e_T(I),$$

$$\mathfrak{e}_T^{\mathcal{A}} = \inf_{\text{full-space-output algorithms}} \sup_{I \in \mathfrak{I}} e_T(I).$$

Here $e_T(I)$ denotes the probability of a non-ϵ-optimal output on instance I. The label $\mathcal{E}$ denotes the executed-only output rule defined in Section 4; the first infimum requires $\widehat{a}_T \in \{W_1, \ldots, W_T\}$ and the second allows $\widehat{a}_T \in \mathcal{A}$. The budgets $T_{\mathcal{E}}(\delta)$ and $T_{\mathcal{A}}(\delta)$ are the smallest $T \in \mathbb{N}$ for which an algorithm of the respective type satisfies $\sup_{I \in \mathfrak{I}} e_T(I) \le \delta$. This definition requires an attaining algorithm for the success guarantee, not merely an infimum error equal to δ. The constructive upper bound below ensures that both budgets are finite.

Proof of Theorem 4.2. The proof in Appendix C.2 never restricts the recommendation to an observed action, so it establishes the claimed stronger lower bound for every full-space decoder on the same function class and generator. This family belongs to $\mathfrak{I}$. Consequently, for every full-space algorithm its worst-instance error is at least $\frac{1}{4}e^{-8pT\epsilon^2/d}$, and taking the infimum preserves this lower bound. Every executed-output algorithm is feasible in the full-space model, which gives $\mathfrak{e}_T^{\mathcal{A}} \le \mathfrak{e}_T^{\mathcal{E}}$.

Finally, run Algorithm 1 with $\ell = pT\epsilon^2/(128d)$ using the common lower bound p. Every instance in $\mathfrak{I}$ satisfies this coverage bound and has $M \le d+1$. Applying Corollary B.2 at $r = \epsilon/2$ is uniform on $\mathfrak{I}$ and gives $\mathfrak{e}_T^{\mathcal{E}} \le \min\{1, (d+2)e^{-pT\epsilon^2/(128d)}\}$. These three inequalities prove Eq. (9); no attainment of either infimum is assumed.

For the budget comparison, every uniformly correct full-space algorithm must satisfy Eq. (25) on the hard family. This yields the left inequality in Eq. (10) after substituting $p = 1 - \beta(\epsilon/2, x_0)$. An executed-only algorithm is also a full-space algorithm at the same budget, giving $T_{\mathcal{A}}(\delta) \le T_{\mathcal{E}}(\delta)$. Finally, the explicit executed-only algorithm above is uniformly correct at every integer $T \ge \lceil 128d\log((d+2)/\delta)/(p\epsilon^2)\rceil$, which proves the right inequality. If $\delta \le \min\{1/4, 1/(d+1)\}$, then $\log(1/(2\delta)) \ge \frac{1}{2}\log(1/\delta)$ and $\log((d+2)/\delta) \le \frac{5}{2}\log(1/\delta)$. Both budgets therefore have order $d\log(1/\delta)/(p\epsilon^2)$, including integer rounding. □

The statistical lower bound concerns all measurable recommendations, not only a particular fitted-model argmax. A learned score can be optimized exactly over the entire action space without supplying any new reward information.

C.4 The Necessary Qualification: an Instance Can Still Become Easier

Proposition C.3. *For any $N \ge 1$, $0 < p < 1$, and $0 < \epsilon < 1$, there is a known-function instance with dimension zero for which the best executed-only success probability is $1 - (1-p)^T$, whereas full-space output succeeds with probability one without observing any rewards.*

Proof. Use a singleton function class and disjoint size-N sets of known mean-one and known mean-zero actions. Return the former with probability p, independently each round, and the latter otherwise. Before the first mean-one list, no executed action is ϵ-optimal, so every executed-output algorithm fails on an event of probability $(1-p)^T$. Executing a mean-one action whenever available attains the complementary success probability. An unrestricted decoder can return a known mean-one action immediately, whether or not it was ever proposed. □

This does not contradict Theorem 4.2, which concerns worst cases with $d \ge 1$. It also shows why the executed-only additive exposure computational cost cannot be claimed as a universal full-space lower bound. The precise conclusion is *no improvement of the leading worst-case statistical rate*, not *no possible benefit from a larger output range*.

D The Lower Bound with Only Two Candidates

The elementary family in Theorem 4.1 requires $N \ge d+1$. The following independent refinement works for every fixed $N \ge 2$. Its useful dimension-dependent comparison is in the stated high-confidence range; it is not a matching constant-confidence claim for every prescribed (d, M, N).

Theorem D.1 (Full-space lower bound for fixed list size)**.** *Let $d \ge 1$, $m = d+1$, $N \ge 2$, $0 < \epsilon \le 1/8$, and $p \in (0, 1]$. There is a function class of exactly m functions with $d_{\mathcal{F}}(\epsilon/2) = d$ and a size-N*

history-dependent generator of coverage p such that every budget-T algorithm, even with full-space output, satisfies

$$e_T^{\max} > \min\left\{\frac{1}{8192m^5},\ \frac{1}{2m}\exp\left(-\frac{512pT\epsilon^2}{m}\right)\right\}. \tag{26}$$

Consequently, uniform error at most $\delta \le 1/(8192m^5)$ requires

$$T > \frac{m}{512p\epsilon^2}\log\frac{1}{2m\delta} \ge \frac{m}{1024p\epsilon^2}\log\frac{1}{\delta}.$$

Proof of Theorem D.1. **Models and coverage.** For every coordinate $j \in [m]$, create actions $a_{j,+}, a_{j,-}$. Under model f_i, set

$$\theta_j^{(i)} = \begin{cases} +1, & j = i, \\ -1, & j \neq i, \end{cases} \qquad f_i(a_{j,+}) = \frac{1}{2} + 2\epsilon\theta_j^{(i)}, \qquad f_i(a_{j,-}) = \frac{1}{2} - 2\epsilon\theta_j^{(i)}.$$

Add zero-mean dummy actions. Offline observations, if supplied, use a source input on which all models agree, with independent Gaussian noise as in Appendix C.2. Under every model, each coordinate has one action of mean $1/2 + 2\epsilon$, the common global optimum, and one of mean $1/2 - 2\epsilon$. The lower action is not ϵ-optimal.

Draw an independent Bernoulli(p) indicator each round. On failure, output only dummies. On success, output the next pair in the deterministic cycle $1, 2, \ldots, m, 1, 2, \ldots$, padding to size N with dummies. Advance the coordinate counter only on successful indicators. The counter is determined by the observed history, so the construction is an admissible fixed history-dependent kernel. Every informative pair contains a global optimum under every model; coverage is exactly p.

Dimension. The actions $a_{1,+}, \ldots, a_{m-1,+}$ are independent at scale $\epsilon/2$, witnessed at position j by f_j, f_m. A coordinate cannot be repeated in an independent sequence, since any pair disagreeing on one sign action disagrees by the same magnitude 4ϵ on the other sign action. After $m-1$ distinct coordinates have appeared, every pair of distinct models already disagrees by 4ϵ on at least one preceding coordinate. No further independence is possible. Thus the dimension is $m - 1 = d$.

A more informative experiment. Grant an observation $X \sim \mathcal{N}(2\epsilon\theta_j, 1)$ on every informative round at coordinate j, regardless of the action chosen. Querying $a_{j,+}$ is simulated by reward $1/2 + X$, and querying $a_{j,-}$ by $1/2 - X$; Gaussian symmetry gives the correct reward distributions. A dummy reward can be simulated independently. Hence this granted experiment can simulate every actual adaptive procedure. Also grant terminal access to every action. A lower bound in this more informative experiment is therefore sufficient.

Put a uniform prior on the m models, and let J be the number of informative rounds by time T. Then $\mathbb{E}J = pT$. Conditional on the list indicators, coordinate j has a deterministic sample count $n_j \le J/m + 1$. Relative to the auxiliary all-negative Gaussian distribution, define

$$Z_j = 4\epsilon\sum_{s=1}^{n_j} X_{j,s}, \qquad \pi_j = \frac{e^{Z_j}}{\sum_{i=1}^m e^{Z_i}}.$$

The reference distribution need not be in the function class. The Gaussian log likelihood ratio at coordinate j is $4\epsilon X$, so π_j is exactly the posterior probability of model f_j. Conditional on the list indicators and an actual model, Z_j is Gaussian with mean $\pm 8\epsilon^2 n_j$ and variance $16\epsilon^2 n_j$. Algorithmic randomization or independently simulated dummy rewards add no information about the model beyond this granted data.

A fixed-budget posterior-uncertainty event. Define, directly from T,

$$b_T = \max\left\{\log 64 + 2\log m,\ \frac{256p\epsilon^2 T}{m}\right\}, \qquad \delta_T = \frac{e^{-2b_T}}{2m}.$$

Markov's inequality gives $\mathbb{P}(J \le 4pT) \ge 3/4$. On this event,

$$n_j \le \frac{4pT}{m} + 1 \le \frac{b_T}{64\epsilon^2} + 1 \le \frac{b_T}{32\epsilon^2}.$$

The last step uses $b_T \geq 64\epsilon^2$, which follows from $\epsilon \leq 1/8$, $m \geq 2$, and the first term in its definition. Consequently $|\mathbb{E}Z_j| \leq b_T/4$ and $\mathrm{Var}(Z_j) \leq b_T/2$.

The standard two-sided Gaussian tail bound (Lattimore & Szepesvári, 2020, Theorem 5.3), with $|\mathbb{E}Z_j| \leq b_T/4$ and $\mathrm{Var}(Z_j) \leq b_T/2$, gives, conditional on any such list-indicator sequence and any actual model,

$$\mathbb{P}\left(|Z_j| > b_T \mid \text{list indicators, model}\right) \leq 2e^{-9b_T/16}.$$

The same inequality holds for $n_j = 0$, when $Z_j = 0$. Since $b_T \geq \log 64 + 2\log m \geq (16/9)\log(8m)$, a union bound makes $\mathbb{P}(\max_j |Z_j| \leq b_T \mid \text{list indicators, model}) \geq 3/4$. Thus, under the prior mixture,

$$\mathbb{P}\left(J \leq 4pT,\ \max_j |Z_j| \leq b_T\right) \geq \frac{9}{16}.$$

On this event, every posterior weight satisfies $\pi_j \geq e^{-2b_T}/m = 2\delta_T$. Since $m \geq 2$, the other weights sum to at least $2\delta_T$, so also $\pi_j \leq 1 - 2\delta_T$. Action $a_{j,+}$ is optimal exactly under model f_j, and action $a_{j,-}$ is optimal exactly under the other models. Every action, even one not previously proposed, therefore has conditional error probability at least $2\delta_T$; dummies are always bad. This remains true for randomized recommendations. Averaging gives Bayesian error at least $(9/16)\,2\delta_T = 9\delta_T/8 > \delta_T$, and the worst-model error is at least the Bayesian error. Finally,

$$\delta_T = \min\left\{\frac{1}{8192m^5},\ \frac{1}{2m}\exp\left(-\frac{512pT\epsilon^2}{m}\right)\right\},$$

which proves Eq. (26) without first assuming a target δ.

For the sample-complexity conclusion, if $\delta \leq 1/(8192m^5)$ and $T \leq [m/(512p\epsilon^2)]\log(1/(2m\delta))$, both terms in Eq. (26) are at least δ, contradicting error at most δ. Finally, $\delta \leq 1/(4m^2)$ implies $\log(1/(2m\delta)) \geq \frac{1}{2}\log(1/\delta)$; the stated range is stronger. This proves the sample-complexity consequence. □

E Implementation and Interpretation

Incremental statistics. For the predictions $v_f = f(x_0, W_t)$ at the executed action, update

$$S_t(f) = S_{t-1}(f) + (R_t - v_f)^2, \qquad D_t(f,g) = D_{t-1}(f,g) + (v_f - v_g)^2.$$

Offline observations use the same updates at their own context–action pairs. Every original model continues to be scored, including models eliminated from the active set. Equivalently, the log-evidence matrix can be maintained directly through

$$H_t(f,g) = H_{t-1}(f,g) + \frac{(R_t - v_f)^2 - (R_t - v_g)^2 + \frac{1}{2}(v_f - v_g)^2}{4}.$$

The squared-loss difference in this update also equals $(v_g - v_f)(2R_t - v_f - v_g)$, which avoids subtracting two large squared residuals.

Numerically stable mixture scores. For each row let $h_f = \max_g H_t(f,g)$ and compute

$$\Phi_t(f) = h_f + \log \sum_{g \in \mathcal{F}} \exp\left(H_t(f,g) - h_f\right).$$

This is exactly the log-sum-exp in Eq. (11), not an approximation. Keep f active only if it was active previously and this score is at most ℓ. Do not drop competitors from the sum or renormalize it by the current active-function-class size: those operations are not the proved filter.

Computational cost. For an evaluable explicit function class, optimistic selection takes at most $\mathcal{O}(MN)$ prediction evaluations and arithmetic per round. Updating the disagreement matrix and the mixture scores uses $\mathcal{O}(M^2)$ arithmetic and storage. These updates need no additional model evaluations beyond predictions of all original function class members at the labeled input. The terminal decoder uses $\mathcal{O}(MT)$ evaluations, or arithmetic alone if predictions at executed actions were cached using $\mathcal{O}(MT)$ storage. No global action-space optimization is used by the proposed algorithm. Full-space optimization is granted only to the comparison algorithms in the lower bounds.

Parameters. There is one statistical radius ℓ; its value is specified by the fixed-budget calibration or the desired-success corollary. The log-evidence coefficient $1/4$ and disagreement coefficient $1/2$ are fixed by the sub-Gaussian proof, not additional hyperparameters. A known noise proxy changes the first coefficient as described in Appendix B.5. The fixed-budget optimization uses supplied coverage and dimension bounds, not estimates treated as exact without justification.

Empirical interpretation On the same transcript the mixture filter is no larger than the loss-only confidence set at the same radius. This can remove models that inflate confidence widths and depress terminal lower bounds. However, different filters can choose different queries and induce different future lists; same-transcript inclusion does not imply uniformly higher terminal success. The comparison concerns identical transcripts; different filters may induce different future data. Exact realizability, a function class fixed before localization, and a valid pre-reward conditional noise bound are required by the confidence argument. Offline inputs need not be i.i.d.; they may be selected predictably using earlier observations. The misspecified experiments in Section 5 do not certify these assumptions, and more aggressive elimination can be harmful when they fail.

Completion outside the nonempty-class event. The main algorithm displays only the nonempty-class update and selection rules. To define its distribution on every transcript, after an empty intersection choose the first candidate in each remaining round and recommend W_1 at termination, including when emptiness occurs at initialization or exactly at T. This convention requires no change to the main pseudocode. On the truth-retention event in Eq. (12), it is never invoked; all other paths are already covered by the single $Me^{-\ell}$ term. Thus the stated unconditional guarantees remain valid without an additional error allowance. The reported experiments retain this same convention rather than discarding such episodes.

There is a deterministic exception under noiseless observations. If every reward equals its true mean, then $S_t(f^\star) = 0$ and $S_t(g) = D_t(f^\star, g)$. Hence $H_t(f^\star, g) = -D_t(f^\star, g)/8 \le 0$, so $\Phi_t(f^\star) \le \log M$. Therefore, when $\ell \ge \log M$, the true function remains in every raw and nested set on every noiseless path. The horizon-calibrated radius need not satisfy that inequality for very small budgets; its corresponding success certificate is then zero.

F Experimental Details

The synthetic experiment isolates uncertainty remaining after source observations and limited access to informative actions. The LCBench experiment evaluates the same generate-and-judge structure with archived hyperparameter rewards. Both use true mean rewards only for evaluation, and all terminal recommendations are restricted to executed actions.

F.1 Synthetic Environment and Evaluation

Reward models and source observations. A model is indexed by $h = (g, j, b)$, with $g, b \in \{0, 1\}$ and $j \in \{0, \dots, M/4 - 1\}$. We use $M \in \{4, 8, 16\}$ and retain both coarse groups g at every class size. The target actions consist of an incumbent with mean 0.55, eight upgrade configurations at four intensities, and 17 closed-list actions with mean 0.20 under every model. For configuration $k \in \{0, \dots, 7\}$ and intensity $s \in \{1/4, 1/2, 3/4, 1\}$, the upgrade mean is

$$f_h(x_0, (k, s)) = 0.55 + s \begin{cases} -0.10, & k = 4g + j,\ b = 0, \\ 0.35, & k = 4g + j,\ b = 1, \\ -0.35, & k \ne 4g + j. \end{cases}$$

The incumbent is optimal for conservative models with $b = 0$, and the matching full-intensity upgrade has optimal mean 0.90 for improvement models with $b = 1$. Each run contains $128/M$ episodes of every true model in random order. The episode-specific optimum, rather than its balanced average 0.725, is used to compute regret.

At eight source contexts x_c^{src}, indexed by $c \in \{0, \dots, 7\}$, a fixed action has mean

$$f_h(x_c^{\mathrm{src}}, 0) = 0.20 + 0.60g + 0.01 \cos\left(\frac{2\pi(c - j)}{8}\right) + 0.005(2b - 1).$$

Table 2: The four synthetic methods. Every decoder maximizes its score over distinct executed actions. The empirical mean uses only that action's observed online rewards.

Method	Candidate selection	Terminal decoding
Proposed	Largest prediction over the surviving class	Largest minimum prediction over the terminal class
Model greedy	Largest prediction of the global squared-loss minimizer	Largest empirical mean
Mean greedy	An untried candidate first, otherwise the largest empirical mean	Largest empirical mean
Uniform	Uniform candidate position	Largest empirical mean

Each context contributes two observations. Offline and online rewards add independent Gaussian noise with standard deviation 0.25 and are not clipped. The class is fixed before these labels are drawn. The offline data strongly distinguish g but only weakly distinguish j and b. The model-based methods use all 16 source observations, whereas the empirical-mean methods do not treat source rewards as target-action observations.

Candidate generation and refinement. Independently each round, an open list is drawn with probability $1-\beta$. It contains the incumbent, all eight full-intensity upgrades, and eight distinct actions sampled uniformly from the 24 lower-intensity upgrades. A closed list contains the 17 common low-reward actions. Both list types are randomly permuted and contain 17 distinct actions. Thus β is exactly the miss probability at the evaluation tolerance $\epsilon = 0.05$.

The proposed method uses the statistics in Eq. (1)–Eq. (3) with log evidence divided by $4\sigma^2$, as in Appendix B.5, and $\sigma = 0.25$. The radius $\ell_T = \log(0.5MT)$ is fixed before localization and throughout each budget-specific run. Every mixture score includes all original models, including eliminated competitors. Localization is applied after the full offline log, and the online class is nested. Empty-class paths follow the completion rule in Appendix E and remain in the reported averages. This empirical radius is not the separate horizon-calibrated radius used to obtain the exponential guarantee in Eq. (7).

Selectors and decoders. Table 2 summarizes the four plotted methods. Model greedy minimizes cumulative squared prediction error over the original class using both source data and its own online history. Mean greedy assigns an untried candidate an infinite selection score and otherwise uses that action's observed mean. Its final decoder excludes unexecuted actions. Source observations, candidate lists, and the action-indexed online noise are paired across methods, but each method observes only its own executed rewards.

Score-based selection ties favor the earliest position in the permuted list, and model ties favor the smaller model index. Empirical-decoder ties favor more observations, then earlier first execution, then smaller action identifier. Pessimistic-decoder ties use earlier first execution and then smaller identifier.

Budgets and aggregation. The evaluated budgets are

$$T \in \{1, 2, 4, 8, 12, 16, 24, 32, 48, 64, 96, 128, 192, 256\}.$$

The proposed method is initialized separately for each budget. The three baseline selectors do not depend on the horizon, so their trajectory prefixes may be reused. The main grid uses 20 independent runs of 128 balanced episodes, and a separate-seed replication uses ten such runs. Results first average episodes within a run and then average the run means. Figure bands show one sample standard deviation across run means, with the lower band clipped at zero for display only. All episodes contribute to regret, including truth exclusions and empty-class paths.

Additional results. Across the six panels in Figure 3, the proposed method's mean regret at $T = 256$ is between 0.00014 and 0.00055. The independent-seed replication also places it below the three baselines at this budget in all nine settings. Its early-budget advantage is not uniform. For

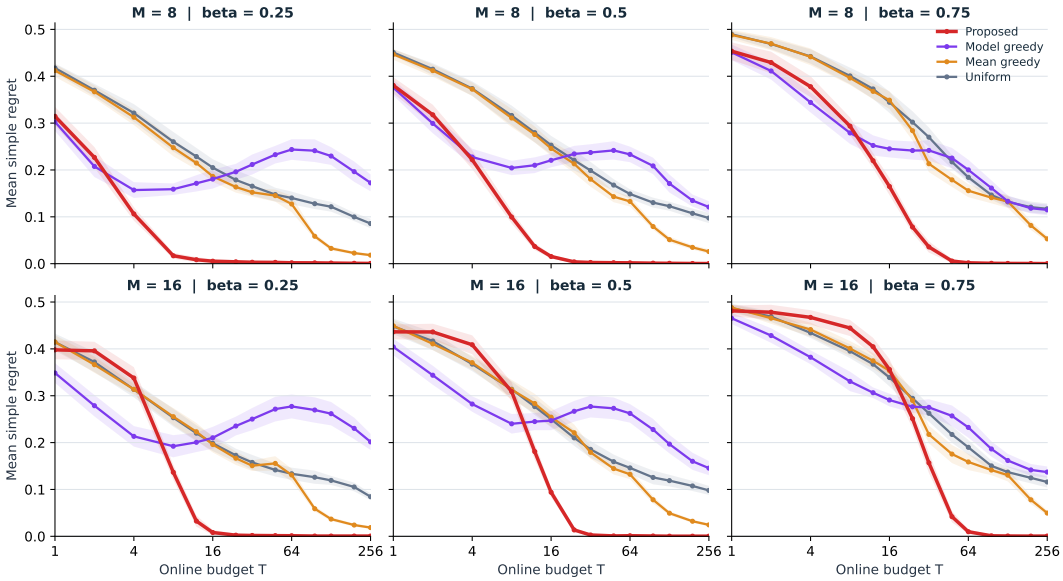


Figure 3: Additional synthetic simple-regret curves. The top row uses $M = 8$ and the bottom row uses $M = 16$. Columns vary the generator miss probability β. The methods, budgets, aggregation, and standard-deviation bands are identical to Figure 1.

Table 3: Same-trajectory decoder comparisons in the central synthetic setting with $M = 8$ and $\beta = 0.50$. Entries are mean simple regret. Each pair changes only the terminal decoder, keeping all executed actions and observations fixed.

Selector	Decoder	$T = 64$	$T = 256$
Model greedy	Empirical	0.23320	0.12090
Model greedy	Pessimistic	0.11225	0.11197
Optimistic	Empirical	0.08054	0.00852
Optimistic	Pessimistic	0.00181	0.00027

example, at $M = 16$ and $\beta = 0.75$, it is best at every tested budget from 24 onward but not at every smaller budget.

The shared incumbent provides no information that distinguishes the models. When the empirical-risk minimizer is conservative, Model greedy selects this incumbent on open rounds and uninformative actions on closed rounds. The relative model losses then stop changing, which can prevent discovery of an improvement. In the central setting $M = 8$, $\beta = 0.50$, its regret at $T = 256$ consists of an access gap of 0.09406 and a decoding gap of 0.02684. Here the access gap compares the optimum with the best executed action, and the decoding gap compares the best executed action with the recommendation.

Table 3 separates the contributions of exploration and decoding. With empirical decoding fixed, optimistic selection improves over Model greedy. With the optimistic trajectory fixed, pessimistic decoding improves over empirical decoding. Removing only the offline evidence in this central setting raises the proposed regret at $T = 16$ from 0.01535 to 0.06189. These comparisons concern this disclosed, exactly realizable construction rather than universal superiority over exploration methods.

F.2 LCBench Setup and Evaluation

Problem instantiation and candidate exposure. We use the LCBench archive for the `albert`, `jannis`, `bank-marketing`, `MiniBooNE`, and `higgs` tasks (Zimmer et al., 2021). Each of these five tasks is treated as a target context, with 2,000 neural-network hyperparameter configurations forming its action space. A configuration specifies batch size, learning rate, momentum, weight decay, number of layers, maximum units, and maximum dropout. Its mean reward is the archived

validation balanced accuracy at array index 51. Executing an action returns this mean plus fresh Gaussian noise with standard deviation 0.10. The archived means and the catalogue optimum are used only for evaluation.

Each round presents eight candidate actions. Four are drawn from a fixed portfolio for that target context containing the top 256 configurations under a frozen anchor score, and four are drawn from the remaining catalogue. The candidate sequence depends on the target context, replicate, seed, and round, but not on the selected method or its reward history, so all methods face identical candidate lists. The online noise is also paired by target context, replicate, seed, budget, action, and visit count without sharing observations across methods.

Reward-function class and calibration. The reward-function class contains 16 linear ridge predictors built on a shared frozen, adapter-free Qwen3.5-0.8B representation. For a context-action pair, the encoder supplies an action-span mean-pooled hidden representation, and each class member applies its own linear ridge readout followed by a fixed alignment and the common target calibration. The members share the same encoder and differ only in their linear readouts. These readouts are fitted with different reweightings of 20 source contexts, using a mixture of Dirichlet weights and a uniform floor to produce a diverse bank. A separate ridge anchor on the same frozen features supplies the portfolio score and common alignment scale. No bank member is trained on these five target contexts.

Proposed, Model greedy, Mean greedy, and Uniform begin with the same reward-independent calibration prefix, consisting of eight rounds at $T = 16$ and 16 rounds at larger budgets. The prefix counts toward the online budget. Its actions alternate between the lowest and highest frozen-anchor score in the current candidate list, so observed rewards cannot alter the prefix. These rewards fit the common target calibration of the bank predictions and are not used as refinement evidence. The proposed method then uses nested refinement, optimistic candidate selection over the surviving class, and pessimistic terminal decoding. The fitted bank is not assumed to contain the true LCBench reward function, and the experiment therefore evaluates the method under practical model misspecification rather than testing the exact realizability assumption used in the theory.

Compared methods. Model greedy selects according to the original bank member with the smallest cumulative squared loss. Mean greedy prioritizes an unexecuted candidate and otherwise uses empirical mean reward, while Uniform samples a candidate position uniformly. These three methods use empirical-mean terminal decoding. The proposed decoder instead recommends the executed action with the largest minimum calibrated prediction over the terminal class.

The LLM judge baselines(LLM+empirical and LLM+direct) use pretrained Qwen3.5-0.8B in bfloat16 without LCBench-specific training, adapters, the source-trained judge bank, or target calibration. At each round, its prompt contains statistics of the target context, descriptions of the hyperparameters, its observed history, and the eight current candidates. The LLM selects one candidate by comparing the conditional next-token probabilities of the eight option labels. Its history contains only executed configurations, their visit counts, and empirical observed means. It never receives unexecuted rewards, catalogue ranks, or true means.

We apply two terminal decoders to the same LLM trajectory. The empirical decoder recommends the executed action with the largest empirical mean reward. The direct LLM decoder instead scores each distinct executed action independently in its own prompt, showing that action together with its visit count and empirical mean reward. The prompt asks whether this action has the highest true reward among those executed, and the decoder scores it by the difference between the log probabilities assigned to "Yes" and "No". The action with the largest score is recommended. Because each action is scored separately, the decoder does not depend on its position relative to other candidates.

Evaluation and interpretation. The budgets are $T \in \{16, 32, 64, 128, 256\}$, with ten seeds for each of the five target contexts and one fixed localization replicate. Thus each method contributes 50 episodes per budget and 250 across all budgets. We first average seeds within each target context and then give each context equal weight. The standard deviations in Table 1 are the sample standard deviations of the five target-context means.

Across budgets, the proposed method improves the target-context-averaged regret by 0.00661 over Model greedy, 0.00834 over Mean greedy, 0.01465 over Uniform, and 0.01076 over the LLM with empirical decoding.