PAC-CF: CALIBRATING IRREVERSIBLE FRONTIER PRUNING IN LLM-GUIDED SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM-guided search explores multiple candidate trajectories, but at substantial test-time cost. Pruning low-scoring frontier candidates can control this cost, yet it also turns potentially biased evaluator scores into irreversible decisions: systematic ranking errors can persist under repeated scoring and remove useful branches. We propose Probably Approximately Correct Conformal Filtering (PAC-CF). Its fixed-frontier analysis formulates elimination as an (ε, δ)-PAC problem under bounded evaluator bias; its operational rule separately calibrates a score-gap threshold on held-out tasks by running the original controller without PAC-CF and using post-search verifier labels to measure the deficit of solution-preserving candidates relative to the frontier leader. Conditional on exchangeable native-controller tasks with nonempty protected exposure, conformal calibration gives finite-sample coverage for retaining at least one verifier-defined valid continuation at every protected frontier on the native trajectory. At deployment, PAC-CF removes only candidates whose gap from the highest frontier score exceeds the frozen threshold. We evaluate PAC-CF across three domains, five controllers, and four request budgets from B100 to B500. In the cross-domain/controller macro averages, the point estimates for all three workload measures are lower at every budget; the paired-bootstrap 95% confidence interval for utility excludes zero at B100 and B200. For pruning-aware ToolTree, the full-test-set cross-domain utility difference is $+4.38$ points at each tested budget; on the natural-termination sensitivity cohort, physical requests decrease by 18.94–18.95% and end-to-end token usage by 23.57–23.76%.

1 INTRODUCTION

LLM-based agents increasingly decompose high-level goals into subgoals, interleave reasoning with environment interactions, and invoke external tools over multiple steps (Wei et al., 2022; Wang et al., 2023; Yao et al., 2023b; Schick et al., 2023; Qin et al., 2024). Tree search is commonly used to explore alternative action trajectories (Yu et al., 2025). In existing approaches, an LLM-based evaluator often scores candidate nodes, and a controller uses those scores to determine which branches to expand. Compared with single-path greedy search without backtracking, tree search maintains alternative continuations and has improved performance on several reasoning, planning, and tool-use benchmarks (Yao et al., 2023a; Hao et al., 2023; Zhou et al., 2024), at the cost of additional test-time computation.

A standard LLM-guided search generates and scores candidate next actions, and a controller retains a subset for subsequent exploration; we call the scored candidate set a *frontier*. Restricting further exploration through *pruning* reduces cost but may discard useful branches. Prior work studies bounded-width selection, score-based prioritization, and explicit action or state pruning (Xie et al., 2023; Koh et al., 2025; Zhuang et al., 2024; Yang et al., 2026; Wang et al., 2025; Liu et al., 2026). When admission is driven by LLM-based scores, these methods can inherit systematic errors from miscalibration, presentation effects, or model misspecification (Haldar & Hockenmaier, 2025; Li & Yang, 2024; Shi et al., 2025; Thakur et al., 2025; Park et al., 2025). Averaging repeated evaluations can reduce stochastic variation, but need not remove persistent ranking error. This matters because pruning is effectively irreversible within a run: a removed candidate's subtree is not explored unless the controller later regenerates it.

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.

A controlled study on 50-node Euclidean TSP instances (Kool et al., 2019) illustrates that such evaluator bias can remain substantial even on relatively simple tasks. Despite averaging 16 independent evaluator scores per candidate, the globally optimal continuation ranks only 5.11 on average at conflict frontiers. Consequently, fixed-cardinality top-4 pruning removes the optimal continuation at 64.4% of these frontiers, while top-6 pruning still removes it at 39.1%. The effect compounds at the task level: among 24 conflict instances, none preserves the optimal reference path across all inspected frontiers under top-4 pruning, and only 1 of 24 does so under top-6. These results show that repeated evaluation can reduce stochastic score variation, yet persistent ranking errors can remain and lead to irreversible pruning errors. This motivates explicitly calibrating irreversible pruning decisions against evaluator uncertainty.

PAC-CF follows a *theory–calibration–deployment* pipeline. The fixed-frontier theory models bounded evaluator bias and derives a score margin sufficient for reliable elimination. Separately, calibration runs the original (*native*) controller and uses post-search verifier labels to measure the deficit of solution-preserving candidates relative to the frontier leader. The resulting threshold is frozen before deployment. The deployment rule removes only candidates whose score gap exceeds that threshold and then returns the survivors to the native controller. We evaluate the same rule on Game24 (Yao et al., 2023a), Blocksworld (Valmeekam et al., 2023), and GTA (Wang et al., 2024) with five representative controllers (Zhuang et al., 2024; Zhou et al., 2024; Yang et al., 2026). Across budgets of 100, 200, 300, and 500 requests per task, the cross-domain/controller macro-average point estimates for all three workload metrics are lower, while the utility confidence interval excludes zero only at B100 and B200. At B500, the full-paired-set macro reductions are 9.47% for physical requests, 12.40% for Net Graph nodes, and 10.25% for tokens. Our contributions are:

- **Theory:** We formulate irreversible frontier pruning as a PAC decision problem. The analysis models persistent bias through bounded evaluator bias, separates it from reducible sampling noise, derives the effective separation $\Delta - 4L$ sufficient for reliable elimination, and identifies lower-bound regimes where repeated scoring cannot make deletion reliable.
- **Calibration:** We calibrate PAC-CF on trajectories generated by the native controller. Each calibration task contributes one task-level maximum score gap as its conformal nonconformity score, and the resulting margin is fixed before deployment. Conditional on nonempty protected exposure, this calibration provides finite-sample coverage for retaining at least one verifier-defined valid continuation at every protected frontier of an exchangeable native-controller trajectory.
- **Deployment:** We use the calibrated threshold in a score-gap pruning rule whose form follows the fixed-frontier analysis, while leaving the rest of the controller unchanged. The same rule is tested across three domains, five controllers, and four budgets. On pruning-aware ToolTree under a 100-request budget, PAC-CF achieves $+4.38$ **full-test-set utility points**; on the natural-termination sensitivity cohort, it uses 18.95% **fewer physical requests and** 23.76% **fewer end-to-end tokens**.

2 Related Work

Search and pruning. Multi-step LLM agents combine reasoning with environment actions and tool use (Yao et al., 2023b; Shinn et al., 2023; Schick et al., 2023; Qin et al., 2024; Yang et al., 2024). Tree-based methods such as ToT, RAP, TS-LLM, LATS, ToolChain*, and ToolTree explicitly search alternative trajectories (Yao et al., 2023a; Hao et al., 2023; Wan et al., 2024; Zhou et al., 2024; Zhuang et al., 2024; Yang et al., 2026). Evaluator-guided beam and best-first search concentrate expansion on promising branches, while FETCH and EquivPruner prune redundant states or actions (Xie et al., 2023; Koh et al., 2025; Wang et al., 2025; Liu et al., 2026). PAC-CF builds on established work in tree-search pruning and conformal calibration, with a specific focus on calibrating a frozen score-gap margin for irreversible frontier admission from task-level maxima on native-controller traces.

These approaches intervene at different points in the search process. ToT, RAP, TS-LLM, and LATS organize how candidate trajectories are proposed, evaluated, and revisited; ToolChain* and ToolTree incorporate controller-specific mechanisms for navigating or pruning the action space; FETCH and EquivPruner target redundant exploration. PAC-CF instead treats an existing controller and evaluator as the interface to be calibrated: it inserts a frozen admission threshold and otherwise returns the surviving candidates to the native controller. It neither learns a semantic-equivalence model nor replaces the controller's expansion policy. The comparison is therefore between complementary

control points in LLM-guided search, rather than between interchangeable implementations of one pruning rule.

Calibration under evaluator error. LLM judges and reward models can exhibit systematic bias, miscalibration, and reward overoptimization (Shi et al., 2025; Thakur et al., 2025; Park et al., 2025; Khalaf et al., 2025). Recent verifier-aware analyses show how imperfect verification can be amplified during generation and how verifier quality interacts with sampling and coverage (Rohatgi et al., 2026; Mukherjee et al., 2026). Conformal methods calibrate LM output or action sets, dynamic beam widths, and planning constraints (Quach et al., 2024; Ren et al., 2023; Deutschmann et al., 2024; Moss et al., 2024; Vishwakarma et al., 2025). SCOPE-Gen sequentially filters generated samples while calibrating retention of an admissible output (Kladny et al., 2025); Conformal Path Reasoning calibrates path-level sets for knowledge-graph QA using PUCT-collected trajectories (Lin et al., 2026). The corresponding calibration objects are not the same. Prior conformal work considers, among other targets, sampled output sets, admissible action sets, beam widths, safety constraints, sequentially filtered outputs, and path-level answer sets. PAC-CF assigns one score to each calibration task: the largest leader-to-protected-action gap over protected frontiers encountered on that task's native-controller trajectory. Calibrating this task-level maximum supports a simultaneous path-survival event along an exchangeable native trajectory, conditional on the trajectory exposing at least one protected frontier. It does not by itself provide coverage for frontiers induced after PAC-CF changes the controller's trajectory, nor does it turn verifier-defined feasibility into optimal downstream utility. These distinctions specify the target and support of the guarantee without implying priority over the broader conformal-search literature. PAC-CF's specific focus is a frozen margin for irreversible admission on history-dependent frontiers, calibrated from the maximum leader-to-protected-action gap along native-controller trajectories. Its fixed-frontier analysis relates deletion to best-arm identification, confidence sequences, and robust bandits (Kaufmann et al., 2016; Howard et al., 2021; Lykouris et al., 2018; Mukherjee et al., 2021; Garcelon et al., 2022).

3 PRELIMINARIES

Agent reasoning and tool planning as tree search. We consider a task instance x with initial state z_0. At a state z, the agent may choose an admissible action $a \in \mathcal{A}(z)$. Executing a produces a reasoning-model, tool, or environment response

$$o \sim P(\cdot \mid z, a),$$

after which the state is updated according to

$$z' = T(z, a, o).$$

A verifier $\mathcal{V}$ determines whether a completed trajectory solves the task, while U measures its utility.

A search node v at depth ℓ represents a partial trajectory starting from z_0 and consisting of actions $a_1, \ldots, a_\ell$ and their corresponding responses $o_1, \ldots, o_\ell$. We denote this trajectory by $h(v)$. The states along the trajectory evolve recursively as

$$z_i = T(z_{i-1}, a_i, o_i), \qquad i = 1, \ldots, \ell,$$

and we write $z(v) = z_\ell$ for the state represented by node v.

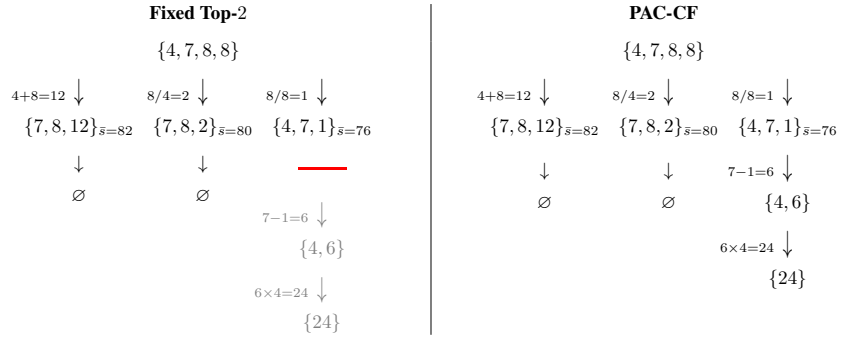
Tree search repeatedly selects a partial trajectory, proposes child actions, evaluates the resulting continuations, and propagates value information. In MCTS (Kocsis & Szepesvári, 2006; Browne et al., 2012; Silver et al., 2016; Zhou et al., 2024), selection balances $Q_{\mathrm{MCTS}}(v, a)$ with exploration, expansion adds children, a rollout or tool execution supplies evidence, and backup updates ancestors. Best-first, A*-style, and Levin-style controllers (Koh et al., 2025; Zhuang et al., 2024; Orseau et al., 2018) order the open set differently, but all must decide which scored continuations may enter future search.

Frontier admission and irreversible pruning. Let p_t be the t-th scored frontier at selected node v_t. Our algorithm returns a frontier of M_t candidates, $\mathcal{C}(p_t) = \{a_{t,1}, \ldots, a_{t,M_t}\} \subseteq \mathcal{A}(z(v_t))$, where an action includes its reasoning content or tool arguments. For each candidate a, R evaluator observations $Y_{t,a,1:R}$ are aggregated into an online score $s_t(a)$. The native rule selects an **admitted**

set $\mathcal{K}^{\mathrm{N}}(p_t) \subseteq \mathcal{C}(p_t)$, commonly by top-$K$ selection or a fixed threshold, and the controller continues only through candidates in this set.

Thus pruning removes a candidate edge or child node at one frontier occurrence, not an action from the global vocabulary. If $a \notin \mathcal{K}^{\mathrm{N}}(p_t)$, no downstream work proceeds through (v_t, a) in that run, but the action may reappear from another state. Let $\mathcal{P}(p_t) \subseteq \mathcal{C}(p_t)$ contain the candidates whose subtrees include a completion accepted by the verifier; we call these candidates *valid continuations*. A frontier is *protected* when $\mathcal{P}(p_t) \neq \varnothing$. For a task x, let $\mathcal{E}_{\mathrm{N}}(x)$ denote the protected frontiers exposed by the native controller; the event $\mathcal{E}_{\mathrm{N}}(x) \neq \varnothing$ is called *protected exposure*. Our reliability target is existential path preservation: admission retains at least one valid continuation at every protected frontier. PAC-CF changes only this set; the native controller still allocates the remaining budget.

Running example: Game24. For $z_0 = \{4, 7, 8, 8\}$, consider three candidate next steps whose repeated evaluator scores are averaged before pruning. Suppose the evaluator assigns mean scores 82, 80, and 76 to the resulting states $\{7, 8, 12\}$, $\{7, 8, 2\}$, and $\{4, 7, 1\}$, respectively. The first two branches cannot reach 24, whereas the lowest-scored branch preserves the valid path $\{4, 7, 1\} \to \{4, 6\} \to \{24\}$. The resulting pruning decisions are illustrated below:



Fixed top-2 pruning removes the solution-preserving branch, whereas PAC-CF retains it whenever $82 - 76 \leq \widehat{\tau}_Q$. Thus PAC-CF does not correct the evaluator score itself, but calibrates the score gap tolerated before an irreversible pruning decision.

4 METHODOLOGY

PAC-CF separates fixed-frontier evidence, a calibrated score-gap threshold, and the online pruning decision. We analyze reliable elimination under bounded evaluator bias, calibrate a path-preserving threshold, and insert it into a native controller. Algorithm 1 summarizes calibration and deployment.

4.1 BIAS-AWARE ELIMINATION UNDER BOUNDED EVALUATOR BIAS

Following Section 3, PAC-CF receives a scored candidate set $\mathcal{C}(p_t)$ and returns an admitted set $\mathcal{K}_t \subseteq \mathcal{C}(p_t)$. For fixed-frontier analysis, write $\mathcal{A}_t \equiv \mathcal{C}(p_t)$. Candidate m has downstream value μ_m, optimal value $\mu^* = \max_j \mu_j$, and gap $\Delta_m = \mu^* - \mu_m$; its r-th evaluator observation is $Y_{m,r}$.

Standing model for the fixed-frontier results in Appendix B.1–B.7.

Assumption 1 (Bounded evaluator bias)**.** *For every active candidate and evaluation,*

$$|\mathbb{E}[Y_{m,r} \mid \mathcal{F}_{r-1}] - \mu_m| \leq L,$$

and the centered residual is conditionally σ^2-sub-Gaussian.

Let $b_m(n)$ be the empirical mean after n evaluator samples and write $b_m(t) := b_m(n_m(t))$ at inspected frontier step t. Define

$$u_{\mathrm{stat}}(n) = \sqrt{\frac{2\sigma^2 \ln(\pi^2 n^2 |\mathcal{A}_t|/(3\delta))}{n}}, \qquad u_{\mathrm{dist}}(n) = u_{\mathrm{stat}}(n) + L. \tag{1}$$

Proof: Appendix B.1.

Lemma 1 (Frontier-wise time-uniform concentration)**.** *Under Assumption 1, with probability at least $1-\delta$, simultaneously for every $m \in \mathcal{A}_t$ and every $n \geq 1$,*

$$|b_m(n) - \mu_m| \leq u_{\mathrm{stat}}(n) + L = u_{\mathrm{dist}}(n). \tag{2}$$

Lemma 1 separates reducible sampling uncertainty from the persistent term that matters for deletion. Repeated scoring can drive $u_{\mathrm{stat}}(n)$ toward zero, but it does not shrink the bias allowance L.

Proof: Appendix B.2.

Theorem 1 (Step-wise PAC upper bound)**.** *At a fixed inspected frontier, on the simultaneous concentration event of Lemma 1, interval elimination safely prunes a suboptimal candidate m whenever*

$$2u_{\mathrm{stat}}(n_m) + 2u_{\mathrm{stat}}(n_{m^*}) < \Delta_m - 4L - \varepsilon. \tag{3}$$

Hence the effective certifiable gap is $\Delta_{\mathrm{eff},m} := \Delta_m - 4L$. If $\Delta_{\mathrm{eff},m} > \varepsilon$, the pairwise evaluation complexity has leading scaling

$$N_m = O\left(\frac{\sigma^2[\log(|\mathcal{A}_t|/\delta) + \log(\Delta_{\mathrm{eff},m}^{-2})]}{(\Delta_{\mathrm{eff},m} - \varepsilon)^2}\right). \tag{4}$$

Corollary 1 (Fixed-frontier (ε, δ)-PAC pruning safety)**.** *Suppose candidate m is eliminated only when, for some candidate j,*

$$b_m(n_m) + u_{\mathrm{dist}}(n_m) < b_j(n_j) - u_{\mathrm{dist}}(n_j) - \varepsilon.$$

Under Assumption 1, with probability at least $1-\delta$, no candidate satisfying $\mu^* - \mu_m \leq \varepsilon$ is eliminated at any inspected sample count. Consequently, the surviving set contains an ε-optimal candidate.

Proof: Appendix B.3.

Theorem 1 makes the distinction between ranking and deletion explicit: a small observed score gap can be enough to order two candidates but still be insufficient for irreversible elimination. In particular, $\Delta_m \leq 4L + \varepsilon$ cannot satisfy this sufficient certificate merely by adding more repeated scores.

4.2 Conformal Calibration on Native-Controller Trajectories

The fixed-frontier analysis does not prescribe the deployed threshold directly. Instead, it identifies the requirement that motivates calibration: repeated scoring can shrink u_{stat} but does not shrink the bounded bias allowance L, so irreversible elimination requires a positive score margin. Rather than estimating the theoretical pointwise bias bound L, PAC-CF uses native-controller trajectories to calibrate a task-level operational score-gap threshold: the empirical score gap tolerated before pruning for a particular controller–evaluator pair. On those tasks, PAC-CF is disabled while the native controller records frontiers and evaluator scores. After search terminates, offline verifier information is used to identify $\mathcal{P}(p) \subseteq \mathcal{C}(p)$, the set of candidates whose subtrees contain at least one completion accepted by the verifier.

For every recorded native frontier p, each candidate receives a fixed number R of evaluator observations, aggregated as $\bar{Y}(a) = R^{-1}\sum_{r=1}^{R} Y_{p,a,r}$. These repetitions are fixed before the frontier decision rather than adaptively extended until a pruning certificate fires. For a protected frontier define

$$C_{\mathrm{path}}(p) = \left[\max_{a\in\mathcal{C}(p)} \bar{Y}(a) - \max_{a\in\mathcal{P}(p)} \bar{Y}(a)\right]_+, \quad C_{\mathrm{task}}^{\mathrm{N}}(x) = \max_{p\in\mathcal{E}_{\mathrm{N}}(x)} C_{\mathrm{path}}(p), \tag{5}$$

for tasks with nonempty protected exposure $\mathcal{E}_{\mathrm{N}}(x)$. The construction is *existential*: at each protected frontier, at least one valid continuation must survive. It is also frontier-wise and task-level: each protected action is compared only with siblings at the same irreversible decision, and the maximum over exposed native frontiers produces one calibration score per task. Accordingly, the guarantee is one of feasibility/path survival, not optimality or maximum graded utility of the retained continuation.

Algorithm 1: PAC-CF — Conformal Calibration and frozen score-gap threshold Filtering

Require: native controller Π_{N}; calibration tasks $\mathcal{D}_{\mathrm{cal}}$; target coverage Q; request budget B; slack $\varepsilon \geq 0$; fixed repetitions R.

Offline Calibration

1: **for each** task $x \in \mathcal{D}_{\mathrm{cal}}$ **do**
2: Run Π_{N} with PAC-CF disabled and record every complete scored frontier p. For every $a \in \mathcal{C}(p)$, obtain R matched evaluator observations and compute $s_p(a) = R^{-1} \sum_{r=1}^{R} Y_{p,a,r}$.
3: After the native run terminates, use exact or independent downstream verification available for the domain to label $\mathcal{P}(p) \subseteq \mathcal{C}(p)$, the candidates whose recorded native subtrees contain a completion accepted by the verifier.
4: **if** $\mathcal{E}_{\mathrm{N}}(x) \neq \varnothing$ **then** set $C^{\mathrm{N}}_{\mathrm{task}}(x) \leftarrow \max_{p \in \mathcal{E}_{\mathrm{N}}(x)} \left[\max_{a \in \mathcal{C}(p)} s_p(a) - \max_{a \in \mathcal{P}(p)} s_p(a)\right]_+$.
5: **end for**
6: Sort the n nonempty task scores as $C_{(1)} \leq \cdots \leq C_{(n)}$, set $k_Q \leftarrow \lceil (n+1)Q \rceil$, and freeze $\widehat{\tau}_Q \leftarrow C_{(k_Q)}$.

Online Deployment

7: Initialize native search and request count $\kappa \leftarrow 0$.
8: **while** $\kappa < B$ and the task is not terminated **do**
9: Let Π_{N} construct its next complete frontier p_t and apply its native eligibility checks; for ToolTree, retain the native τ_{pre} eligibility gate.
10: For every $a \in \mathcal{C}(p_t)$, obtain R matched evaluator observations and compute $s_t(a) = R^{-1} \sum_{r=1}^{R} Y_{t,a,r}$.
11: Set $s_t^* \leftarrow \max_{a \in \mathcal{C}(p_t)} s_t(a)$ and $\mathcal{K}_t \leftarrow \{a \in \mathcal{C}(p_t) : s_t^* - s_t(a) \leq \widehat{\tau}_Q + \varepsilon\}$.
12: Prune $\mathcal{C}(p_t) \setminus \mathcal{K}_t$ at this frontier occurrence and return $\mathcal{K}_t$ with the original scores to Π_{N}.
13: Let Π_{N} perform its unchanged ordering, selection, execution/rollout, backup, verification, and termination logic; update κ.
14: **end while**
15: **return** the solution returned by Π_{N}, if one is found.

ToolTree interface. PAC-CF retains the native τ_{pre} eligibility gate, replaces only the subsequent fixed-cardinality top-K admission with Line 11, and does not reapply top-K to PAC-CF survivors.

Connection to the fixed-frontier analysis. Let

$$\mu^*_{\mathcal{P}} = \max_{a \in \mathcal{P}(p)} \mu_a.$$

On the concentration event of Lemma 1, when every candidate receives R evaluations,

$$C_{\mathrm{path}}(p) \leq \left(\mu^* - \mu^*_{\mathcal{P}}\right) + 2u_{\mathrm{stat}}(R) + 2L.$$

Indeed, the largest empirical score is at most $\mu^* + u_{\mathrm{stat}}(R) + L$, whereas the largest protected empirical score is at least $\mu^*_{\mathcal{P}} - u_{\mathrm{stat}}(R) - L$. Thus the observed path gap combines bounded evaluator bias, finite-R sampling uncertainty, and any mismatch between downstream value and the verifier-defined protected set. PAC-CF calibrates this operational gap directly rather than estimating these components separately.

For exchangeable calibration scores $C_{(1)} \leq \cdots \leq C_{(n)}$, freeze

$$k_Q = \lceil (n+1)Q \rceil, \qquad \widehat{\tau}_Q = C_{(k_Q)}, \tag{6}$$

with the usual infeasible-rank convention.

Proposition 2 (Calibration trajectory coverage)**.** *Under exchangeability of calibration and future tasks generated by the same native controller procedure,*

$$\Pr\left(C^{\mathrm{N}}_{\mathrm{task}}(X_{\mathrm{new}}) \leq \widehat{\tau}_Q \mid \mathcal{E}_{\mathrm{N}}(X_{\mathrm{new}}) \neq \varnothing\right) \geq Q.$$

Because $C^{\mathrm{N}}_{\mathrm{task}}$ maximizes over protected frontiers, Proposition 2 provides simultaneous path coverage along the exposed native-controller trajectory. At $Q = .95$, nominal task-level miscoverage is at most 5% up to the discrete-rank improvement; calibration counts and ranks are reported in Appendix H.

4.3 Online frontier pruning

At test time, with $s_t(a) = R^{-1} \sum_{r=1}^{R} Y_{t,a,r}$ and $s_t^* = \max_{a \in \mathcal{C}(p_t)} s_t(a)$, PAC-CF keeps

$$\mathcal{K}_t = \{a \in \mathcal{C}(p_t) : s_t^* - s_t(a) \leq \widehat{\tau}_Q + \varepsilon\}. \tag{7}$$

Thus pruning depends on score gap rather than rank.

Corollary 2 (Path survival on calibration trajectories)**.** *On the coverage event in Proposition 2, Eq.* (7) *with $\varepsilon \geq 0$ retains at least one valid continuation at every protected frontier of the matched trajectory from the native controller. Consequently, conditional on nonempty protected exposure, simultaneous path survival along the native-controller trajectory occurs with probability at least Q.*

Proof: Appendix C.

Proposition 2 provides policy-conditional coverage on exchangeable native-controller trajectories; Section 5 evaluates the resulting closed-loop behavior, with implementation details in Appendix C.

Table 1: TSP50 endpoint summary. CPER is the fraction of reference actions removed by top-K pruning; task survival is reported at $R = 16$. Full $R \times K$ results are in Appendix E.

Stratum / R	Mean C_{path}	Mean reference rank	CPER $K = 4$	CPER $K = 6$	Task survival $K = 4$	Task survival $K = 6$
Control / 1	16.67	4.64	54.2%	27.1%	–	–
Control / 16	9.50	4.58	52.1%	24.0%	4/24	8/24
Conflict / 1	20.51	5.11	57.5%	43.7%	–	–
Conflict / 16	12.32	5.11	64.4%	39.1%	0/24	1/24

5 Experiments

5.1 Persistent Misranking under Repeated Scoring

Before the agent benchmarks, we isolate evaluator error on 24 Euclidean TSP50 instances (Kool et al., 2019). Exact binary MILP tours with iterative subtour elimination (Dantzig et al., 1954) define the reference next city, and a live Ministral-3-14B evaluator scores candidate continuations. We compare control frontiers with local-greedy conflict frontiers, where the reference action differs from the nearest-neighbor choice; CPER is the fraction removed by fixed-cardinality top-K pruning.

Increasing R from 1 to 16 substantially narrows the reference score gap, yet produces almost no rank recovery: control rank moves only from 4.64 to 4.58, while conflict rank remains 5.11. At $R = 16$, CPER is still 52.1% versus 64.4% for control versus conflict frontiers at $K = 4$, and 24.0% versus 39.1% at $K = 6$. Frontier-level errors also compound at the task level, as summarized by the reference-path survival table; full $R \times K$ results are in Appendix E.

Motivation. Repeated evaluation reduces score variation without reliably repairing pruning-critical misranking, motivating the bounded evaluator-bias model of Assumption 1.

5.2 Experimental Setup

Evaluation. We evaluate PAC-CF on Game24 arithmetic reasoning ($N = 100$), PlanBench Blocksworld planning ($N = 99$), and GTA structured tool planning ($N = 99$) (Yao et al., 2023a; Valmeekam et al., 2023; Wang et al., 2024). Utility is task success for Game24/Blocksworld and Avg3 (Tool/Argument/Plan F1) for GTA. We select five representative controllers that cover distinct search regimes: classical UCT-MCTS, A*-style ToolChain*, LATS with MCTS-based agent search, pruning-aware ToolTree, and policy-guided LevinTS (Zhuang et al., 2024; Zhou et al., 2024; Yang et al., 2026; Orseau et al., 2018). Prompts, candidate generation, evaluator, verifier, downstream search logic, and request budget are held fixed across Native and PAC-CF runs, so the paired comparison changes only the pruning rule in Eq. (7).

Calibration. For each controller–domain pair, calibration produces a scalar margin that is fixed across budgets. Verifier labels are used during calibration to construct the path scores. We prespecify $Q = .95$ and freeze every controller–domain margin before test-time evaluation. Budgets are $B \in \{100, 200, 300, 500\}$ requests per task. Utility and the primary capped-deployment workload estimand are evaluated on the full paired test set. We additionally report a natural-termination sensitivity estimand on paired tasks for which neither arm exhausts the hard budget. The latter is descriptive because membership in this cohort may be affected by the pruning intervention. Repeated scoring, frozen configurations, calibration ranks, and bootstrap details are in Appendix D–H.

Thresholds. Each controller–domain pair uses a separately calibrated $\widehat{\tau}_{.95}$ that remains fixed across budgets. Main experiments use $Q = .95$, $\varepsilon = 0$, and four evaluator observations per candidate; thresholds, calibration ranks, model settings, and seeds are in Appendix H.

5.3 MAIN RESULTS

Efficiency Estimands. We report three primary workload measures: physical requests, Net Graph nodes, and end-to-end tokens. For $X \in \{\mathrm{PReq}, \mathrm{Graph}, \mathrm{Token}\}$, let $\mathcal{I}^{\mathrm{all}}$ denote all paired test tasks and let $\mathcal{I}^{\mathrm{NBC}}$ denote the subset for which neither arm exhausts the hard budget. Our primary capped-deployment estimand is

$$R_X^{\mathrm{all}} := 1 - \frac{\sum_{i \in \mathcal{I}^{\mathrm{all}}} X_i^{\text{PAC-CF}}}{\sum_{i \in \mathcal{I}^{\mathrm{all}}} X_i^{\mathrm{Native}}}. \tag{8}$$

For comparison, the natural-termination sensitivity estimand replaces $\mathcal{I}^{\mathrm{all}}$ by $\mathcal{I}^{\mathrm{NBC}}$ and is denoted R_X^{NBC}. Across B100–B500, the all-task macro reductions are 6.14–9.47% for physical requests, 9.03–12.40% for Net Graph nodes, and 6.84–10.25% for tokens.

Physical Requests counts requests normalized to provider calls; larger values mean more request-level model usage. Net Graph measures the number of realized search nodes, excluding the root, after adaptive reallocation; larger values mean that more of the search graph is actually explored. E2E Token reports the combined prompt and completion tokens; larger values mean more total model-token usage.

We additionally report two gross metrics that reconstruct the observed Native subtree and the downstream candidate evaluations removed by the corresponding PAC-CF pruning decisions. Appendix D provides the full accounting procedure, including how overlapping subtrees are merged and how roots without a counterpart are excluded rather than extrapolated.

Table 2: Natural-termination sensitivity and path-preservation diagnostics. (a) Domain/macro results on uncensored pairs; utility uses the full test set. (b) Native coverage at frozen $Q = .95$ and B100 adaptive path survival. †LATS is 98.96% if the unresolved task is counted as failure; adaptive survival is descriptive. CIs for (a) are in Appendix Table 20.

(a) Natural-termination sensitivity

Domain	Budget	ΔU	PReq ↓	NetG ↓	Token ↓
Game24	B100	$-\mathbf{1.00}$ pp	10.73%	3.63%	11.41%
	B200	$-\mathbf{1.00}$ pp	12.31%	6.78%	12.76%
	B300	-1.60 pp	12.56%	7.01%	13.08%
	B500	-1.80 pp	**12.76%**	**7.06%**	**13.35%**
Blocksworld	B100	$+\mathbf{3.03}$ pp	1.69%	11.22%	2.38%
	B200	$+2.22$ pp	5.04%	14.17%	5.62%
	B300	$+1.41$ pp	5.43%	**14.67%**	6.12%
	B500	$+1.21$ pp	**5.74%**	**14.67%**	**6.46%**
GTA	B100	$+1.23$ pp	8.27%	13.90%	9.89%
	B200	$+1.20$ pp	9.38%	14.81%	10.92%
	B300	$+\mathbf{1.28}$ pp	10.12%	15.46%	12.35%
	B500	$+1.17$ pp	**10.96%**	**16.09%**	**12.85%**
Macro	B100	$+\mathbf{1.09}$ pp	6.90%	9.58%	7.89%
	B200	$+0.81$ pp	8.91%	11.92%	9.77%
	B300	$+0.37$ pp	9.37%	12.38%	10.52%
	B500	$+0.19$ pp	**9.82%**	**12.61%**	**10.89%**

(b) Coverage and path survival

Controller	Native coverage	B100 survival
UCT-MCTS	97.00%	97.00%
ToolChain*	95.88%	96.91%
LATS	100%†	100%
ToolTree	–	99.00%
LevinTS	93.75%	93.75%
Game24 pooled	–	**97.46%**
Blocksworld pooled	–	**99.80%**

Table 3: Paired hierarchical bootstrap 95% CIs. Utility uses all paired tasks; workload uses uncensored pairs. Bold marks the best standard-budget estimates; ToolTree B100 is shown separately.

Budget	Δ utility [95% CI]	P-Req. ↓ [95% CI]	Net Graph ↓ [95% CI]	E2E Token ↓ [95% CI]
B100	**+1.088** [0.251, 1.941]	+6.90% [5.76, 7.98]	+9.58% [8.09, 10.98]	+7.89% [6.74, 8.99]
B200	+0.807 [0.043, 1.585]	+8.91% [7.62, 10.14]	+11.92% [10.26, 13.44]	+9.77% [8.45, 11.01]
B300	+0.366 [−0.421, 1.152]	+9.37% [8.14, 10.58]	+12.38% [10.81, 13.87]	+10.52% [9.12, 11.84]
B500	+0.193 [−0.631, 1.025]	**+9.82%** [8.48, 11.11]	**+12.61%** [10.98, 14.15]	**+10.89%** [9.46, 12.23]
ToolTree B100	+4.381 [1.43, 7.41]	+18.95% [14.81, 22.96]	+12.90% [7.54, 17.77]	+23.76% [19.56, 27.85]

Natural-Termination Sensitivity and Path Preservation. Within the natural-termination cohort, workload CIs remain above zero at every budget, while macro utility is resolved positive only at B100 and B200. Panel (b) shows native Game24 coverage of 93.75–100% and B100 adaptive survival of 97.46% and 99.80% pooled on Game24 and Blocksworld, respectively; the adaptive rates are diagnostics, not extensions of the native-trajectory conformal guarantee.

Reallocation Tradeoffs. Aggregate workload reductions increase with budget, but controller responses differ. At B100, Game24 ToolTree cuts physical requests by 46.91% and tokens by 53.11% while losing three success points and increasing Net Graph by 19.13%; GTA ToolTree gains 7.05 Avg3 points while reducing requests by 14.24% and Net Graph by 26.29%. At B500, Game24

ToolTree still suppresses 11.42% of native physical nodes despite a 21.34% larger realized Net Graph, so gross pruning need not shrink the adaptive trajectory. Across B100–B500, ToolTree averages a $+4.38$-point utility change, with 18.94–18.95% request, 12.16–12.90% Net Graph, and 23.57–23.76% token reductions on the natural-termination cohort; full gross and controller-level results are in Appendix Table 19 and Appendix I.

Interpreting the Workload Measures. The three workload measures capture different effects of a changed search trajectory: physical requests count evaluator/model calls, Net Graph counts the retained structure after controller reallocation, and tokens capture prompt and completion traffic. Because saved budget may be reallocated to surviving branches, gross pruning need not reduce Net Graph. We therefore report the measures separately rather than as one compression or speedup factor.

5.4 Sensitivity across Evaluator Models

With Ministral-3-14B and separately recalibrated $Q = .95$ margins, macro reductions remain 6.22–7.86% for physical requests, 10.11–10.94% for Net Graph, and 6.80–9.20% for tokens, while utility changes range from $+0.08$ to -0.62 points. In this secondary-model study, the workload reductions remain positive while the utility changes do not. Using $Q = .90$ is more aggressive but worsens the utility–efficiency tradeoff; full results are in Appendix G.

5.5 Heterogeneity Across Controllers and Domains

Observed Efficiency without ToolTree. In this sensitivity analysis, removing ToolTree leaves positive workload reductions: from B100 to B500, physical-request, Net Graph, and token reductions increase from 3.88 to 7.54%, 8.75 to 12.72%, and 3.93 to 7.72%, while utility changes from $+0.265$ to -0.855 points. Thus ToolTree strengthens the utility result but is not the sole source of the observed workload savings.

Sensitivity to Budget Censoring. The uncensored cohort grows from 71.0% at B100 to 92.1% at B500. In the reported all-task analyses, workload reductions remain positive at every budget, so censoring changes their magnitude but not their sign; full results are in Appendix J.

The Guarantees Have Distinct Supports. The fixed-frontier results concern elimination under the bias bound L and its $\Delta - 4L$ certificate; Proposition 2 instead gives path-retention coverage on exchangeable native-controller trajectories. Closed-loop effects after PAC-CF changes admission are evaluated empirically.

Limitations and Scope. The calibrated margin is evaluator-, controller-, and domain-specific, so changing any of these components requires recalibration rather than reusing $\widehat{\tau}_Q$. Coverage applies to exchangeable tasks with nonempty protected exposure on native-controller trajectories, whereas deployed PAC-CF can induce different frontiers. Constructing protected labels also depends on domain-specific offline verification and may be costly or incomplete when exact reachability is unavailable. Finally, retaining a verifier-accepted continuation does not ensure optimal downstream utility, and the reported workload–utility tradeoffs remain benchmark-specific. These limitations motivate shift-aware calibration, cheaper protection labeling, and guarantees for adaptively induced frontiers.

6 Conclusion

PAC-CF treats irreversible pruning as a reliability decision. Bias-aware theory identifies when score separation supports elimination; conformal calibration turns native-controller path deficits into a finite-sample margin; deployment leaves the surrounding controller unchanged. In the evaluated cross-domain/controller macro averages, the point estimates for request, node, and token workload are lower at each budget. On ToolTree B100, the full-test-set utility difference is $+4.38$ points, while the natural-termination cohort uses 18.95% fewer physical requests and 23.76% fewer tokens. Coverage is conditional on protected exposure and exchangeable native-controller tasks; paired end-to-end experiments measure closed-loop effects after PAC-CF changes admission.

REPRODUCIBILITY STATEMENT

Section 4 and Algorithm 1 specify the PAC-CF calibration and deployment procedures. Appendix C documents protection-label construction and controller integration, while Appendices D and H record evaluator settings, request-budget and workload accounting, calibration sizes and ranks, model configurations, random seeds, and frozen deployment settings. Appendix K specifies the paired bootstrap and confidence-interval construction. The anonymous supplementary archive maps each numbered empirical table and figure to recovered source code or a provenance note; where a historical one-off plotting script was unavailable, the archive states that limitation rather than supplying a reconstructed substitute.

AI USE STATEMENT

Generative AI tools were used for literature search and related-work discovery, reference cross-checking, experimental implementation including code generation, and drafting, revising, and polishing parts of the manuscript. The authors manually reviewed AI-assisted text, code, citations, and reported results, and remain responsible for all claims and artifacts in the submission.

REFERENCES

Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, 2012. doi: 10.1109/TCIAIG.2012.2186810.

George B. Dantzig, Delbert R. Fulkerson, and Selmer M. Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, 2(4):393–410, 1954. doi: 10.1287/opre.2.4.393.

Nicolas Deutschmann, Marvin Alberts, and María Rodríguez Martínez. Conformal autoregressive generation: Beam search with coverage guarantees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11775–11783, 2024. doi: 10.1609/aaai.v38i10.29062. URL https://ojs.aaai.org/index.php/AAAI/article/view/29062.

Evrard Garcelon, Vashist Avadhanula, Alessandro Lazaric, and Matteo Pirotta. Top k ranking for multi-armed bandit with noisy evaluations. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 6242–6269, 2022.

Rajarshi Haldar and Julia Hockenmaier. Rating roulette: Self-inconsistency in llm-as-a-judge frameworks. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 24986–25004. Association for Computational Linguistics, 2025.

Shibo Hao, Yiyang Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In *Proceedings of EMNLP*, pp. 8154–8173, 2023.

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.

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.

Hadi Khalaf, Claudio Mayrink Verdun, Alex Oesterling, Himabindu Lakkaraju, and Flavio Calmon. Inference-time reward hacking in large language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Klaus-Rudolf Kladny, Bernhard Schölkopf, and Michael Muehlebach. Conformal generative modeling with improved sample efficiency through sequential greedy filtering. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=1i6lkavJ94.

Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *Machine Learning: ECML 2006*, volume 4212 of *Lecture Notes in Computer Science*, pp. 282–293. Springer, 2006. doi: 10.1007/11871842_29.

Jing Yu Koh, Stephen Marcus McAleer, Daniel Fried, and Ruslan Salakhutdinov. Tree search for language model agents. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=QF0N3x2XVm.

Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! In *International Conference on Learning Representations*, 2019.

Yunfan Li and Lin Yang. On the model-misspecification in reinforcement learning. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, pp. 2764–2772. PMLR, 2024.

Shuhang Lin, Chuhao Zhou, Xiao Lin, Zihan Dong, Kuan Lu, Zhencan Peng, Jie Yin, and Dimitris N. Metaxas. Conformal path reasoning: Trustworthy knowledge graph question answering via path-level calibration. *arXiv preprint arXiv:2605.08077*, 2026. URL https://arxiv.org/abs/2605.08077.

Jiawei Liu, Qisi Chen, Jianshu Zhang, Quan Liu, and Defu Lian. Equivpruner: Boosting efficiency and quality in llm-based search via action pruning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 2211–2226, 2026.

Thodoris Lykouris, Vahab Mirrokni, and Renato Paes Leme. Stochastic bandits robust to adversarial corruptions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 114–122, 2018.

Robert J. Moss, Arec Jamgochian, Johannes Fischer, Anthony Corso, and Mykel J. Kochenderfer. Constrainedzero: Chance-constrained pomdp planning using learned probabilistic failure surrogates and adaptive safety constraints. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 6752–6760, 2024. doi: 10.24963/ijcai.2024/746.

Arpan Mukherjee, Ali Tajer, Pin-Yu Chen, and Payel Das. Mean-based best arm identification in stochastic bandits under reward contamination. In *Advances in Neural Information Processing Systems*, volume 34, pp. 9651–9662, 2021.

Arpan Mukherjee, Marcello Bullo, Debabrota Basu, and Deniz Gündüz. Test-time verification via optimal transport: Coverage, roc, & sub-optimality. In *International Conference on Learning Representations*, 2026.

Laurent Orseau, Levi H. S. Lelis, Tor Lattimore, and Théophane Weber. Single-agent policy tree search with guarantees. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Young-Jin Park, Kristjan Greenewald, Kaveh Alimohammadi, Hao Wang, and Navid Azizan. Know what you don't know: Uncertainty calibration of process reward models. In *Advances in Neural Information Processing Systems*, 2025.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/28e50ee5b72e90b50e7196fde8ea260e-Abstract-Conference.html.

Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi S. Jaakkola, and Regina Barzilay. Conformal language modeling. In *The Twelfth International Conference on Learning Representations*, 2024.

Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. Robots that ask for help: Uncertainty alignment for large language model planners. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 661–682. PMLR, 2023.

Dhruv Rohatgi, Abhishek Shetty, Donya Saless, Yuchen Li, Ankur Moitra, Andrej Risteski, and Dylan J. Foster. Taming imperfect process verifiers: A sampling perspective on backtracking. In *International Conference on Learning Representations*, 2026.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551, 2023.

Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. Judging the judges: A systematic study of position bias in llm-as-a-judge. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 292–314. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.ijcnlp-long.18. URL https://aclanthology.org/2025.ijcnlp-long.18/.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

David Silver, Aja Huang, Chris J. Maddison, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 2016. doi: 10.1038/nature16961.

Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. Judging the judges: Evaluating alignment and vulnerabilities in llms-as-judges. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM^2 2025)*, pp. 404–430, 2025. doi: 10.18653/v1/2025.gem-1.33. URL https://aclanthology.org/2025.gem-1.33/.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. *Advances in Neural Information Processing Systems*, 36:38975–38987, 2023.

Harit Vishwakarma, Alan Mishler, Thomas Cook, Niccolo Dalmasso, Natraj Raman, and Sumitra Ganesh. Prune 'n predict: Optimizing llm decision-making with conformal prediction. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 61601–61634. PMLR, 2025.

Ziyu Wan, Xidong Feng, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. Alphazero-like tree-search can guide large language model decoding and training. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 49890–49920. PMLR, 2024.

Ante Wang, Linfeng Song, Ye Tian, Dian Yu, Haitao Mi, Xiangyu Duan, Zhaopeng Tu, Jinsong Su, and Dong Yu. Don't get lost in the trees: Streamlining llm reasoning by overcoming tree search exploration pitfalls. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23946–23959, 2025. doi: 10.18653/v1/2025.acl-long.1167. URL https://aclanthology.org/2025.acl-long.1167/.

Jize Wang, Zerun Ma, Yining Li, Songyang Zhang, Cailian Chen, Kai Chen, and Xinyi Le. GTA: A benchmark for general tool agents. In *Advances in Neural Information Processing Systems*, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022. doi: 10.52202/068431-1800.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, James Xu Zhao, Min-Yen Kan, Junxian He, and Michael Xie. Self-evaluation guided beam search for reasoning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems*, 2024.

Shuo Yang, Soyeon Caren Han, Yihao Ding, Shuhe Wang, and Eduard Hovy. Tooltree: Efficient llm agent tool planning via dual-feedback monte carlo tree search and bidirectional pruning. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Ef5O9gNNLE.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, 2023a.

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*, 2023b.

Xiao Yu, Baolin Peng, Vineeth Vajipey, Hao Cheng, Michel Galley, Jianfeng Gao, and Zhou Yu. Exact: Teaching ai agents to explore with reflective-mcts and exploratory learning. In *International Conference on Learning Representations*, 2025.

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 *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 62138–62160. PMLR, 2024.

Yuchen Zhuang, Xiang Chen, Tong Yu, Saayan Mitra, Victor Bursztyn, Ryan A. Rossi, Somdeb Sarkhel, and Chao Zhang. Toolchain*: Efficient action space navigation in large language models with a* search. In *International Conference on Learning Representations*, 2024.

A NOTATION AND PARAMETERS

Table 4 consolidates the notation used in the main text, methodology, experimental accounting, and theoretical proofs. Equivalent symbols used in search, analysis, and algorithmic views are grouped in the same row. Proof-local symbols are marked explicitly. In particular, the theoretical bias bound L is distinct from the calibrated score-gap threshold $\widehat{\tau}_Q$, and the fixed-frontier failure probability δ is distinct from conformal miscoverage $1 - Q$.

Table 4: Notation used throughout the paper. Equivalent notation from different views is grouped to avoid duplicate definitions.

Symbol	Meaning
Task and search process	
$x = (z_0, \mathcal{A}, P, T, \mathcal{V}, U)$	Agent task: initial state z_0, admissible actions $\mathcal{A}(z)$, response kernel $P(\cdot \mid z, a)$, transition $T(z, a, o)$, verifier $\mathcal{V}(h)$, and trajectory utility $U(h)$.
$a,\ o$	Generic action and the resulting reasoning-model, tool, or environment response.
$v,\ v_t,\ h(v),\ \ell$	Search-tree node, node selected at frontier t, its partial trajectory/history, and trajectory depth. The history symbol h avoids collision with the proof stopping time τ.
B	Per task request budget.
$Q_{\mathrm{MCTS}}(v, a)$	native MCTS action-value statistic; distinct from conformal coverage Q.
$p_t,\ \mathcal{C}(p_t),\ \mathcal{A}_t$	The t-th scored frontier and its candidate set. In fixed-frontier analysis, $\mathcal{A}_t \equiv \mathcal{C}(p_t)$.
$M_t,\ a_{t,j}$	Frontier size $M_t = \lvert\mathcal{C}(p_t)\rvert$ and candidate j at frontier t.
$R,\ Y_{t,a,r},\ Y_{m,r}$	Evaluator repetitions and the r-th observation. Calibration and deployment use fixed $R = 4$ observations per scored candidate with balanced presentation order; $Y_{m,r}$ is the fixed-frontier theory notation.
$s_p(a),\ s_t(a),\ s_t^*$	Mean score from the fixed $R = 4$ observations on a calibration frontier p, the corresponding online mean score at frontier t, and $s_t^* = \max_a s_t(a)$.
$\mathcal{K}^{\mathrm{N}}(p_t),\ \mathcal{K}_t$	Candidate sets admitted by the native rule and PAC-CF, respectively.
$\mathcal{P}(p)$	Protected candidates at frontier p: actions whose subtrees contain at least one completion accepted by the verifier.
Fixed-frontier bias-aware analysis	
$m,\ j,\ m^*$	Candidate/arm indices and the optimal candidate under true downstream value.
$\mu_m,\ \mu^*,\ \Delta_m$	True downstream value, optimum $\mu^* = \max_j \mu_j$, and suboptimality gap $\Delta_m = \mu^* - \mu_m$.
$\mathcal{F}_{r-1},\ \sigma^2$	History before observation r and conditional sub-Gaussian variance proxy of the centered residual.
L	Assumed evaluator-bias bound in Assumption 1; distinct from the calibrated score-gap threshold $\widehat{\tau}_Q$.
$b_m(n),\ b_m(t)$	Empirical mean after n evaluator samples; at inspected step t, $b_m(t) := b_m(n_m(t))$.
$u_{\mathrm{stat}}(n),\ u_{\mathrm{dist}}(n)$	Statistical radius and bias-robust radius, with $u_{\mathrm{dist}}(n) = u_{\mathrm{stat}}(n) + L$.
$\delta,\ \varepsilon$	Fixed-frontier failure probability and elimination tolerance/slack.
$\Delta_{\mathrm{eff},m}$	Effective certifiable gap $\Delta_m - 4L$; proofs may suppress the candidate index and write Δ_{eff}.
$n_m,\ N_m,\ N_m(\tau)$	Samples allocated to candidate m, its elimination sample-complexity bound/scaling, and the realized sample count by proof stopping time τ.
Conformal calibration on native-controller trajectories	
$\bar{Y}(a)$	Aggregated evaluator score for action a on a calibration frontier from the native controller, computed before verifier labeling.
$\mathcal{E}_{\mathrm{N}}(x),\ C_{\mathrm{path}}(p),\ C^{\mathrm{N}}_{\mathrm{task}}(x)$	Protected calibration frontiers for task x, frontier-level path-score gap and task-level nonconformity (the maximum deficit over $\mathcal{E}_{\mathrm{N}}(x)$).
$n,\ C_{(k)}$	Number of exchangeable calibration tasks with nonempty protected exposure and the k-th order statistic of their task scores.
$Q,\ k_Q$	Target conformal coverage and rank $k_Q = \lceil (n+1)Q \rceil$.
$\widehat{\tau}_Q$	Frozen operational score gap margin $\widehat{\tau}_Q = C_{(k_Q)}$; distinct from theoretical bias bound L.
X_{new}	Future exchangeable task in the calibration trajectory coverage statement.
Online PAC-CF deployment	
$\Pi_{\mathrm{N}},\ K$	Surrounding native controller and its fixed-cardinality top-K parameter when applicable.
κ	Running request count in provider call equivalents during deployment.
ε	Nonnegative slack in the direct score gap pruning rule.
τ_{pre}	native ToolTree eligibility gate retained before PAC-CF replaces the subsequent top-K admission step; top-K is not reapplied to PAC-CF survivors.
Proof-local notation in Appendix B	
$q_{m,r},\ \beta_{m,r},\ \xi_{m,r}$	Conditional evaluator mean, predictable bias $\beta_{m,r} = q_{m,r} - \mu_m$ with $\lvert\beta_{m,r}\rvert \le L$, and centered residual $\xi_{m,r} = Y_{m,r} - q_{m,r}$.
$n_{\min}$	Minimum sample count in the compared pair, $\min(n_m, n_{m^*})$.
$C_1,\ C_2,\ W_{-1}$	Auxiliary constants and the negative Lambert-W branch used in the upper-bound inversion.
$\tau,\ \mathrm{kl}(p, q)$	Stopping time and binary relative entropy used in lower-bound arguments.
$\Delta,\ c,\ g$	Generic two-child true-value gap, proof-local centering constant, and observable separation $g = \Delta - 2L$.

Continued on next page

Table 4 continued

Symbol	Meaning
$\mathcal{I},\ \gamma$	Set of separated alternatives and the arbitrarily small positive perturbation used in the additive lower bound.
Experimental workload/accounting notation	
$d,\ c,\ \mathcal{I}^{\mathrm{NBC}}_{d,c,B}$	Domain and controller indices, and the uncensored paired task subset at request budget B.
$X,\ R^{\mathrm{net}}_{X}$	Generic realized workload quantity (e.g., PReq, Graph, Token) and its net reduction from Native to PAC-CF.
$\mathrm{Token}_i,\ p_{ij},\ c_{ij}$	Exact task level prompt and completion token usage and its prompt and completion counts for each operation.
$\mathcal{P}^{\mathrm{M}}_i,\ \phi(u),\ \mathrm{Desc}_{\mathrm{N}}(\phi(u))$	Roots pruned by PAC-CF that are matched to the saved native controller tree, the native node, and its native descendants.
$\mathcal{S}_i$	Union of pruned Native roots and their observed Native descendants for task i.
$V^{\mathrm{N}}_i,\ E^{\mathrm{N}}_i,\ E_i(\mathcal{S}_i)$	Native instantiated nodes, Native scored candidate occurrences, and scored occurrences downstream of the pruned subtrees.
$R^{\mathrm{gross}}_{\mathrm{PNode}},\ R^{\mathrm{gross}}_{\mathrm{VNode}}$	Gross physical node and virtual node reductions reconstructed from Native subtrees.
$N_{\mathrm{eff}},\ \Delta U$	Effective paired task count for a workload cell and utility difference between PAC-CF and Native.
θ	Offline diagnostic score gap threshold in Appendix F; distinct from transition operator T.

B Theoretical Boundaries for Biased Exploration

Standing assumption (restated). Assumption 1 requires $|\mathbb{E}[Y_{m,r} \mid \mathcal{F}_{r-1}] - \mu_m| \le L$ for every active candidate and evaluation, with a conditionally σ^2-sub-Gaussian centered residual. The following statements and proofs use this same fixed-frontier model.

B.1 Lemma 1: statement and proof

Statement. With probability at least $1-\delta$, simultaneously for every $m \in \mathcal{A}_t$ and every $n \ge 1$,

$$|b_m(n) - \mu_m| \le u_{\mathrm{stat}}(n) + L = u_{\mathrm{dist}}(n),$$

where

$$u_{\mathrm{stat}}(n) = \sqrt{\frac{2\sigma^2 \ln(\pi^2 n^2 |\mathcal{A}_t|/(3\delta))}{n}}.$$

Thus repeated scoring shrinks only the statistical radius; the additive bias allowance L remains.

Proof. For evaluation r, define the predictable conditional mean

$$q_{m,r} := \mathbb{E}[Y_{m,r} \mid \mathcal{F}_{r-1}], \qquad \beta_{m,r} := q_{m,r} - \mu_m,$$

and the centered residual

$$\xi_{m,r} := Y_{m,r} - q_{m,r}.$$

Assumption 1 gives $|\beta_{m,r}| \le L$ pathwise, while $\{\xi_{m,r}\}_{r\ge1}$ is a conditionally σ^2-sub-Gaussian martingale-difference sequence. Hence

$$b_m(n) - \mu_m = \frac{1}{n}\sum_{r=1}^{n} \xi_{m,r} + \frac{1}{n}\sum_{r=1}^{n} \beta_{m,r}. \tag{9}$$

The second term is deterministically bounded:

$$\left|\frac{1}{n}\sum_{r=1}^{n} \beta_{m,r}\right| \le \frac{1}{n}\sum_{r=1}^{n} |\beta_{m,r}| \le L.$$

For the martingale term, conditional sub-Gaussianity implies for every fixed m and n,

$$\Pr\left(\left|\sum_{r=1}^{n} \xi_{m,r}\right| \ge n\, u_{\mathrm{stat}}(n)\right) \le 2\exp\left(-\frac{n\, u_{\mathrm{stat}}(n)^2}{2\sigma^2}\right) = \frac{6\delta}{\pi^2 n^2 |\mathcal{A}_t|}.$$

Summing over $n \ge 1$ and using $\sum_{n\ge1} n^{-2} = \pi^2/6$ gives failure probability at most $\delta/|\mathcal{A}_t|$ for one candidate over all inspected sample counts. A second union bound over the finite frontier therefore yields, with probability at least $1-\delta$,

$$\left|\frac{1}{n}\sum_{r=1}^{n} \xi_{m,r}\right| \le u_{\mathrm{stat}}(n) \qquad \forall m \in \mathcal{A}_t,\ \forall n \ge 1.$$

Combining this event with Eq. (9) proves Eq. (2). This argument is time-uniform over the sample counts inspected at the fixed frontier; it does not require the sample allocation across candidates to be uniform. □

B.2 THEOREM 1: STATEMENT AND PROOF

Statement. On the simultaneous concentration event of Lemma 1, a suboptimal candidate m is safely eliminated whenever

$$2u_{\mathrm{stat}}(n_m) + 2u_{\mathrm{stat}}(n_{m^*}) < \Delta_m - 4L - \varepsilon.$$

Hence $\Delta_{\mathrm{eff},m} = \Delta_m - 4L$. If $\Delta_{\mathrm{eff},m} > \varepsilon$, the leading pairwise evaluation complexity is

$$N_m = O\left(\frac{\sigma^2[\log(|\mathcal{A}_t|/\delta) + \log(\Delta_{\mathrm{eff},m}^{-2})]}{(\Delta_{\mathrm{eff},m} - \varepsilon)^2}\right).$$

Proof. The interval rule eliminates a suboptimal candidate m when

$$b_m(t) + u_{\mathrm{dist}}(n_m) < b_{m^*}(t) - u_{\mathrm{dist}}(n_{m^*}) - \varepsilon. \tag{10}$$

Work on the simultaneous event of Lemma 1. The empirical means can deviate toward one another by

$$b_m(t) \le \mu_m + u_{\mathrm{stat}}(n_m) + L, \qquad b_{m^*}(t) \ge \mu^* - u_{\mathrm{stat}}(n_{m^*}) - L.$$

Substituting these extremal directions into Eq. (10), a sufficient condition for elimination is

$$\begin{aligned}(\mu_m + u_{\mathrm{stat}}(n_m) + L) + (u_{\mathrm{stat}}(n_m) + L) < (\mu^* - u_{\mathrm{stat}}(n_{m^*}) - L)\\ - (u_{\mathrm{stat}}(n_{m^*}) + L) - \varepsilon.\end{aligned}$$

Using $\Delta_m = \mu^* - \mu_m$ gives

$$2u_{\mathrm{stat}}(n_m) + 2u_{\mathrm{stat}}(n_{m^*}) < \Delta_m - 4L - \varepsilon, \tag{11}$$

which is Eq. (3). This establishes the effective certifiable gap $\Delta_{\mathrm{eff},m} = \Delta_m - 4L$ and proves the safe-elimination part of the theorem.

The sample counts need not be balanced. Let $n_{\min} = \min(n_m, n_{m^*})$. The radius $u_{\mathrm{stat}}(n) = \sqrt{2\sigma^2 \ln(C_1 n^2)/n}$ is decreasing once n is beyond a constant-size prefix (equivalently, once $\ln(C_1 n^2) > 2$), and this prefix is absorbed by the asymptotic bound below. Hence, in the relevant regime,

$$u_{\mathrm{stat}}(n_m) \le u_{\mathrm{stat}}(n_{\min}), \qquad u_{\mathrm{stat}}(n_{m^*}) \le u_{\mathrm{stat}}(n_{\min}),$$

so the stronger condition

$$4u_{\mathrm{stat}}(n_{\min}) < \Delta_m - 4L - \varepsilon \tag{12}$$

is sufficient. In particular, if $\Delta_m \le 4L + \varepsilon$, the right-hand side is non-positive and additional repeated scoring alone cannot make this certificate fire.

To obtain the leading sample-complexity scaling, set

$$C_1 = \frac{\pi^2 |\mathcal{A}_t|}{3\delta}.$$

Equation (12) requires

$$4\sqrt{\frac{2\sigma^2 \ln(C_1 n_{\min}^2)}{n_{\min}}} < \Delta_m - 4L - \varepsilon.$$

Squaring and rearranging gives

$$\frac{n_{\min}}{\ln(C_1 n_{\min}^2)} > \frac{32\sigma^2}{(\Delta_m - 4L - \varepsilon)^2} =: C_2. \tag{13}$$

The boundary equation can be inverted with the negative branch W_{-1} of the Lambert W function. Using the standard small-argument expansion $-W_{-1}(-x) = \log(1/x) + \log\log(1/x) + o(1)$ gives, up to universal constants and lower-order iterated logarithms,

$$n_{\min} = O\left(\frac{\sigma^2[\log(|\mathcal{A}_t|/\delta) + \log(\Delta_{\mathrm{eff},m}^{-2})]}{(\Delta_{\mathrm{eff},m} - \varepsilon)^2}\right).$$

This is the leading asymptotic scaling reported in Eq. (4); it is not claimed as an exact finite-sample equality (Kaufmann et al., 2016). The argument also shows why the least-sampled member of the compared pair controls the frontier-wise certificate. □

B.3 Proof of Corollary 1

Proof. On the simultaneous event of Lemma 1, elimination implies

$$\mu_m \le b_m(n_m) + u_{\mathrm{dist}}(n_m) < b_j(n_j) - u_{\mathrm{dist}}(n_j) - \varepsilon \le \mu_j - \varepsilon \le \mu^* - \varepsilon.$$

Thus every eliminated candidate has gap strictly larger than ε. □

B.4 Theorem 3: statement and proof

Theorem 3 (Structural $2L$ lower boundary)**.** *Consider the two-child Gaussian subclass of the bounded-bias model with common variance σ^2, true gap $\Delta > 0$, and a learner required to return the unique ε-optimal child with probability at least $1-\delta$, where $\delta < 1/2$ and $\Delta > \varepsilon$.*

1. *If $\varepsilon < \Delta \le 2L$, there exist two admissible instances with opposite optimal children but identical adaptive observation laws. Uniform (ε, δ)-PAC identification is therefore impossible.*
2. *If $\Delta > 2L$, any uniformly (ε, δ)-PAC learner obeys, on a suitable admissible pair,*

$$\mathbb{E}[\tau] \ge \frac{2\sigma^2\ \mathrm{kl}(1-\delta, \delta)}{(\Delta - 2L)^2} = \Omega\left(\frac{\sigma^2 \log(1/\delta)}{(\Delta - 2L)^2}\right). \tag{14}$$

Proof. For the impossibility statement, consider two children. In world P let the true-value vector be

$$(c + \Delta/2,\ c - \Delta/2),$$

and in world Q swap the two coordinates. If $\Delta \le 2L$, choose bounded biases $(-\Delta/2, +\Delta/2)$ in world P and the swapped biases in world Q. Every observation from either queried child then has mean c in both worlds. Therefore, under any predictable adaptive sampling policy, the complete observation-history laws are identical. Yet when $\Delta > \varepsilon$, the sets of valid ε-optimal outputs are disjoint because the optimal child is different in the two worlds. No procedure can return the correct ε-optimal child with probability at least $1-\delta$ in both worlds for $\delta < 1/2$. This proves the first part.

Now assume $\Delta > 2L$. Construct two admissible Gaussian worlds with variance σ^2 and opposite optimal children, using saturated opposing biases $\pm L$. The observable mean separation relevant to distinguishing the worlds is

$$g = \Delta - 2L > 0.$$

Let A be the event that the learner returns child 1. Uniform (ε, δ)-PAC correctness implies

$$P(A) \ge 1 - \delta, \qquad Q(A) \le \delta.$$

By data processing for KL divergence,

$$D_{\mathrm{KL}}(P^H \| Q^H) \ge \mathrm{kl}(1-\delta, \delta),$$

where P^H, Q^H denote the full adaptive transcript laws. The adaptive KL chain rule decomposes the history-level divergence into expected sample counts times per-sample divergences. For this two-world Gaussian construction, each informative query contributes at most $g^2/(2\sigma^2)$, hence

$$D_{\mathrm{KL}}(P^H \| Q^H) \le \mathbb{E}_P[\tau] \frac{g^2}{2\sigma^2}.$$

Combining the last two displays yields

$$\mathbb{E}_P[\tau] \ge \frac{2\sigma^2\ \mathrm{kl}(1-\delta, \delta)}{(\Delta - 2L)^2},$$

which is Eq. (14). Since $\mathrm{kl}(1-\delta, \delta) = \Omega(\log(1/\delta))$ for $\delta < 1/2$, the stated asymptotic form follows. □

B.5 Proposition 4: statement and proof

Proposition 4 (Additive lower bound for separated alternatives)**.** *For a finite base frontier, let*

$$\mathcal{I} = \{m \neq m^* : \Delta_m > \varepsilon,\ \Delta_m + \varepsilon > 2L\}.$$

Assume that for every $m \in \mathcal{I}$ the parameter space contains an alternative that raises candidate m above the original optimum by $\varepsilon + \gamma$ for arbitrarily small $\gamma > 0$. Within the Gaussian bounded-bias subclass, any learner uniformly (ε, δ)-PAC over the base instance and these alternatives must satisfy

$$\mathbb{E}[\tau] \geq \sum_{m \in \mathcal{I}} \frac{2\sigma^2\ \mathrm{kl}(\delta, 1-\delta)}{(\Delta_m + \varepsilon - 2L)^2}. \tag{15}$$

No finite lower bound from this construction is asserted when $\Delta_m + \varepsilon \leq 2L$.

Proof. Fix a base instance and one $m \in \mathcal{I}$. Construct an alternative instance that leaves every candidate except m unchanged and raises the true value of candidate m above the original optimum by $\varepsilon + \gamma$, where $\gamma > 0$ can be arbitrarily small. Assign bias $+L$ to m in the base instance and bias $-L$ to m in the alternative. The observation-mean shift on candidate m is

$$\Delta_m + \varepsilon + \gamma - 2L,$$

which is strictly positive by the definition of $\mathcal{I}$ for sufficiently small γ.

Let A_m be the event that the learner returns candidate m. Uniform (ε, δ)-PAC correctness implies that under the base instance $P(A_m) \leq \delta$, while under the m-alternative $Q_m(A_m) \geq 1 - \delta$. By data processing and the adaptive KL chain rule,

$$\mathbb{E}_P[N_m(\tau)] \frac{(\Delta_m + \varepsilon + \gamma - 2L)^2}{2\sigma^2} \geq \mathrm{kl}(\delta, 1-\delta).$$

Therefore

$$\mathbb{E}_P[N_m(\tau)] \geq \frac{2\sigma^2\ \mathrm{kl}(\delta, 1-\delta)}{(\Delta_m + \varepsilon + \gamma - 2L)^2}.$$

Letting $\gamma \downarrow 0$ gives the per-candidate necessary sample count. Because each alternative changes only its corresponding candidate, the inequalities hold simultaneously for every $m \in \mathcal{I}$ under the same base instance. Summing them and using $\tau = \sum_j N_j(\tau)$ yields

$$\mathbb{E}_P[\tau] \geq \sum_{m \in \mathcal{I}} \frac{2\sigma^2\ \mathrm{kl}(\delta, 1-\delta)}{(\Delta_m + \varepsilon - 2L)^2},$$

which proves Proposition 4. When $\Delta_m + \varepsilon \leq 2L$, this particular separated-alternative construction does not provide a positive observable shift, so no finite bound from this construction is asserted. □

B.6 Summary of Theoretical Regimes

Table 5: Theoretical regimes for a two-child true-value gap Δ. The intermediate region is not labeled impossible because the lower bound does not rule out identification there, while the interval certificate does not guarantee deletion.

Gap regime	Consequence
$\Delta \leq \varepsilon$	Distinguishing the children is unnecessary.
$\varepsilon < \Delta \leq 2L$	PAC identification can be impossible in the worst case.
$\max\{\varepsilon, 2L\} < \Delta \leq 4L + \varepsilon$	Identification not ruled out; deletion not certified.
$\Delta > 4L + \varepsilon$	Finite reliable elimination is available.

B.7 Conditional graceful degradation under severe bias

An unconditional $4L + \varepsilon$ return quality guarantee does not follow from Assumption 1 alone: the remaining statistical radius must also be controlled. The following fixed-frontier corollary states the corresponding guarantee supported by Lemma 1.

Corollary 3 (Graceful degradation for empirically best selection)**.** *Work on the concentration event of Lemma 1, and let*

$$\widehat{m} \in \arg\max_{m \in \mathcal{A}_t} b_m(n_m)$$

be the empirically best candidate at a fixed frontier. Then

$$\mu^* - \mu_{\widehat{m}} \le u_{\mathrm{stat}}(n_{m^*}) + u_{\mathrm{stat}}(n_{\widehat{m}}) + 2L. \tag{16}$$

Consequently, if $u_{\mathrm{stat}}(n_{m^*}) \le L + \varepsilon/2$ *and* $u_{\mathrm{stat}}(n_{\widehat{m}}) \le L + \varepsilon/2$*, then*

$$\mu^* - \mu_{\widehat{m}} \le 4L + \varepsilon.$$

Proof. On the concentration event,

$$\mu^* \le b_{m^*}(n_{m^*}) + u_{\mathrm{stat}}(n_{m^*}) + L.$$

Because $\widehat{m}$ maximizes the empirical mean, $b_{m^*}(n_{m^*}) \le b_{\widehat{m}}(n_{\widehat{m}})$, while again by concentration,

$$b_{\widehat{m}}(n_{\widehat{m}}) \le \mu_{\widehat{m}} + u_{\mathrm{stat}}(n_{\widehat{m}}) + L.$$

Combining the two inequalities gives Eq. (16). The stated $4L+\varepsilon$ consequence follows by substituting the two radius bounds. □

Scope of the severe bias corollary. Corollary 3 is deliberately conditional and fixed-frontier. It does not claim that finite repeated scoring automatically reaches the stated radius condition, nor does it extend the conformal guarantee on calibration trajectories to frontiers induced by PAC-CF. It is included only to retain the valid part of the earlier severe bias intuition without overstating what the current assumptions prove.

C Method and Protocol Details

C.1 Calibration Construction and Controller Integration

For calibration, the native controller first runs without PAC-CF and without solution or future information. Each exposed frontier and its candidates are recorded. Every scored candidate receives exactly $R = 4$ evaluator observations with balanced presentation order, which are averaged before any calibration statistic is formed. After search terminates, offline verifier or solution verification information is used to label

$$\mathcal{P}(p) = \{a \in \mathcal{C}(p) : \text{the subtree under } a \text{ contains at least one solution accepted by the verifier}\}.$$

Tasks with no protected exposure are recorded as missing rather than assigned a zero protection score. The operational score is computed from the evaluator scores and labels assigned after search. For Game24, the reconstruction recovers the protected set using exact symbolic reachability; Blocksworld and GTA use the verifier or solution verification information specific to each domain retained by the calibration pipeline. These labels are used only during calibration and are never available to the online pruning rule. Accordingly, PAC-CF assumes access to an offline procedure for assigning protection labels appropriate to each domain on calibration traces, not to protection labels at deployment. The cost and completeness of this offline procedure are domain dependent; our guarantee is therefore stated with respect to the protected labels available in each benchmark rather than a universal exact reachability oracle. The calibration statistic and pruning rule are shared across domains and controllers, while the offline protection labels are constructed using domain-specific verification information. The theoretical bias allowance L in Assumption 1 is not replaced by or identified with the empirical $\widehat{\tau}_Q$.

At test time PAC-CF uses Eq. (7) only. Candidate generation, native priority/selection, rollout, backup, termination, and task verification remain with the surrounding controller. For ToolTree, PAC-CF is inserted at the same point as the native pruning rule: the τ_{pre} eligibility gate is retained, while the subsequent fixed-cardinality top-K step is replaced by Eq. (7). All other ToolTree logic is unchanged. PAC-CF does not evaluate u_{stat}, estimate the theoretical bias bound L, construct an online confidence interval, adaptively resample candidates, or take δ as a deployment input.

Proof of Corollary 2. If $C^{\mathrm{N}}_{\mathrm{task}}(x) \le \widehat{\tau}_Q$, every exposed native frontier has $C_{\mathrm{path}}(p) \le \widehat{\tau}_Q$. Its best protected action therefore has score deficit at most $\widehat{\tau}_Q$ and belongs to the keep set in Eq. (7).

Protection Target and Controller Interface. The calibration target is existential: on a covered calibration task from the native controller it is enough that at least one valid continuation survives at each protected frontier, rather than every viable action. This is a feasibility/path survival guarantee, not a guarantee that the retained continuation is optimal or maximizes graded utility. For UCT-MCTS and LATS the filter is applied to the complete scored child set before native rollout selection; for ToolChain* and LevinTS it is applied before the native priority/Levin ordering; for ToolTree it replaces only the top-K admission after τ_{pre}, while subsequent evaluation and backup remain unchanged. A candidate removed at one frontier occurrence may still reappear if a different search path generates it again.

Main Text Algorithm. Algorithm 1 gives the complete frontier level PAC-CF procedure. This appendix therefore records only calibration, controller integration, and accounting details rather than repeating the pseudocode.

D Experiment Protocol Details

Balanced Repeated Scoring. Each scored candidate receives exactly four evaluator observations. Candidate order is a deterministic shuffle determined by the seed with paired forward/reverse ordering within microbatches. The four scores are aggregated before the frontier decision. This reduces sensitivity to position effects and stochastic variation without assuming that evaluator bias averages away; the implementation does not adaptively request additional evaluator samples until a pruning decision becomes certifiable.

Efficiency Metrics. The per task budget is a hard cap measured in provider call equivalents. We report physical requests measured in provider call equivalents, Net Graph nodes measuring the realized search structure after adaptive reallocation, and end-to-end prompt and completion tokens. For realized quantities X, reduction is $1 - \sum_i X_i^{\mathrm{PAC}} / \sum_i X_i^{\mathrm{Native}}$ before the declared controller/domain averaging.

The two gross node metrics are reconstructed from native ancestry. If $\mathcal{P}_i^{\mathrm{M}}$ is the set of roots pruned by PAC-CF whose action paths can be matched to Native, then $\mathcal{S}_i = \bigcup_{u \in \mathcal{P}_i^{\mathrm{M}}} (\{\phi(u)\} \cup \mathrm{Desc}_{\mathrm{N}}(\phi(u)))$. Gross physical node reduction is $\sum_i |\mathcal{S}_i| / \sum_i |V_i^{\mathrm{N}}|$; overlapping subtrees are unioned once. Gross virtual node reduction counts scored descendant candidate occurrences in these subtrees, excluding the directly pruned root because PAC-CF scores it before deciding to prune. Unmatched roots are not assigned hypothetical descendants. The resulting metrics therefore quantify observed gross structural suppression and gross downstream evaluation suppression, respectively. Net graph node reduction instead compares the complete Native and PAC-CF trajectories after adaptive reallocation and thus quantifies realized structural benefit. The saved GTA records do not retain complete parent–child ancestry, so gross subtree metrics are not reconstructed for GTA.

Calibration Size and Rank Bound. All reported controller–domain calibrations use exactly 60 task scores from tasks with protected exposure. Tasks with no protected exposure remain missing, and the prespecified calibration stream continues until 60 valid task scores are collected. Thus, for $Q = .95$, $k_{.95} = \lceil .95(60+1) \rceil = 58$, giving the exact rank-based conformal miscoverage bound $1 - 58/61 = 4.92\%$. Technical failures use the same prespecified completion rule without Q retuning.

Bootstrap Analysis. For each condition, utility intervals use 200,000 bootstrap resamples over paired tasks. Domain averages use 300,000 resamples with task identities shared across the five controllers; summaries across the three domains and for ToolTree use 300,000 independent paired resamples within each domain followed by averaging across domains with equal weights. Physical request and net graph node reductions are ratios of totals within each resample before controller/domain averaging. Node point estimates are reconstructed directly from the saved search records and saved counters. This analysis introduces no new LLM calls.

Table 6: Exact TSP50 catastrophic reference path elimination rate (CPER) on the fixed-cardinality matched cohort. Each row uses the same frontiers for every retained cardinality K; ties at the Kth score are retained. Lower is better.

Stratum	R	$K=1$	$K=2$	$K=4$	$K=6$	$K=8$
Control	1	91.7%	76.0%	54.2%	27.1%	0.0%
	2	86.5%	75.0%	52.1%	22.9%	0.0%
	4	87.5%	79.2%	53.1%	22.9%	0.0%
	8	90.6%	81.2%	56.2%	28.1%	0.0%
	16	86.5%	77.1%	52.1%	24.0%	0.0%
Local-greedy conflict	1	86.2%	79.3%	57.5%	43.7%	0.0%
	2	89.7%	79.3%	66.7%	43.7%	0.0%
	4	92.0%	81.6%	62.1%	41.4%	0.0%
	8	87.4%	79.3%	62.1%	43.7%	0.0%
	16	83.9%	74.7%	64.4%	39.1%	0.0%

E EXACT TSP50 STRESS TEST FOR PERSISTENT EVALUATOR MISRANKING

Exact-Reference Test Bed. We generate 24 Euclidean TSP50 instances following the standard random Euclidean routing setting (Kool et al., 2019) and solve each instance exactly using a binary MILP with iterative subtour-elimination constraints (Dantzig et al., 1954) until a single Hamiltonian cycle remains. A fixed orientation of the optimal tour defines the optimal reference next city. For each instance, we select four uniformly distributed control frontiers and four local greedy conflict frontiers before observing any evaluator outputs. Conflict frontiers are positions at which the globally optimal continuation differs from the local nearest-neighbor choice. A live Ministral-3-14B evaluator scores the candidate next cities with $R \in \{1, 2, 4, 8, 16\}$ repeated evaluations. The primary matched cohort includes only frontiers with exactly eight candidates and all 16 evaluator observations, resulting in 96 control frontiers and 87 conflict frontiers.

For a protected reference continuation a^* with repeated score mean $\bar{Y}_R(a^*)$, let $\mathrm{Retain}_{R,K}$ contain all candidates whose mean score is at least the Kth largest mean score. We define the catastrophic path elimination rate as

$$\mathrm{CPER}_{R,K} = \Pr[a^* \notin \mathrm{Retain}_{R,K}] \,.$$

Ranking Persistence under Repeated Evaluation. Pruning aggressiveness has a clear effect on catastrophic elimination, whereas additional evaluator samples do not yield a systematic reduction in CPER. At $R = 16$, retaining four of eight candidates gives CPERs of 52.1% on control frontiers and 64.4% on local greedy conflict frontiers. Even when six of eight candidates are retained, the corresponding rates remain 24.0% and 39.1%. Increasing R from 1 to 16 therefore does not move either stratum into a low CPER regime under fixed-cardinality pruning.

Local–Global Conflict Risk. The conflict stratum isolates frontiers at which the local geometric preference disagrees with the exact globally optimal continuation. At $R = 16$, CPER on conflict frontiers is 12.3 percentage points higher than on control frontiers at $K = 4$ and 15.1 percentage points higher at $K = 6$. The difference is less informative at $K = 1$ and $K = 2$, where elimination rates are already high in both strata. The effect of local–global disagreement is therefore most visible under moderate irreversible truncation.

Pathwise Bias Accumulation. Frontier-level elimination can accumulate across an instance because the exact reference continuation must survive every inspected frontier.

Table 7: Task-level reference path survival at $R = 16$. A task survives when the exact reference continuation is retained at every inspected frontier.

Stratum	$K=1$	$K=2$	$K=4$	$K=6$	$K=8$
Control	0/24	0/24	4/24 (16.7%)	8/24 (33.3%)	24/24
Conflict	0/24	0/24	0/24	1/24 (4.2%)	24/24

At $K = 4$, only $4/24$ control instances preserve the reference path, while no conflict instance does; at $K = 6$, the corresponding counts are $8/24$ and $1/24$.

F Certificate Coverage and Deployment Diagnostics

Calibration–Deployment Scope. All analyses in this section are reconstructed from saved experimental records; no additional LLM/API calls are made. The calibration trajectory analysis evaluates the object in Proposition 2, whereas the deployment analysis examines trajectories generated after PAC-CF changes admission. The latter is an empirical deployment diagnostic and is not covered by the conformal coverage statement.

F.1 Native-Trajectory Coverage and Pruning Aggressiveness

Frozen-Trajectory Coverage Reconstruction. For each reconstructable Game24 controller, sibling candidates and their four saved evaluator observations are restored from saved native controller records. Exact symbolic reachability labels the protected set, after which Eq. (5) is recomputed task by task. Reconstruction is accepted only when the recovered protected frontier/action counters agree with the saved oracle diagnostics. ToolTree is reported as N/A rather than mixing its distinct archived judge channel with the score channel used by the other controllers.

Controller Coverage Heterogeneity. Table 8 shows that, among reconstructable Game24 controllers, observed task level coverage is 97.00% for UCT-MCTS, 95.88% for ToolChain*, and 100% on the 95 reconstructed LATS tasks; counting the unresolved LATS task as a failure still gives 98.96%. LevinTS is lower at 93.75% on 80 exposed tasks. This finite realization should therefore not be summarized as uniformly exceeding 95% across controllers. The reported Binomial$(80, .95)$ reference probability of approximately 0.371 for observing at most 75 successes also indicates that the LevinTS realization is not by itself in tension with a nominal .95 marginal target.

Table 8: Game24 native-trajectory coverage and pruning aggressiveness at the frozen $Q = .95$ operating points. Pruning rates are offline fixed-trajectory diagnostics, not realized end-to-end savings.

Controller	$\widehat{\tau}_{.95}$	Task coverage	Candidate pruning	Protected-action pruning
UCT-MCTS	30	$97/100\,(97.00\%)$	13.27%	9.69%
ToolChain*	25	$93/97\,(95.88\%)$	12.88%	12.26%
LATS	30	$95/95\,(100\%)^{\dagger}$	8.17%	1.52%
LevinTS	50	$75/80\,(93.75\%)$	4.61%	7.27%
ToolTree	20	N/A‡	–	–

†One additional exposed LATS task is unresolved; counting it as a failure gives $95/96 = 98.96\%$. ‡ToolTree is not reconstructed because its archived judge channel differs from the score channel used by the other controllers.

Coverage–Aggressiveness Diagnostic. The pruning rates are reconstructed at the prespecified frozen margins on native-controller trajectories and therefore do not measure realized end-to-end savings, since changing admission can alter subsequent frontiers. LevinTS attains $75/80 = 93.75\%$ observed coverage; observing at most 75 successes out of 80 has probability approximately 0.371 under an independent Binomial$(80, .95)$ reference and is therefore not by itself in tension with the nominal .95 marginal target. No test-set value is used to retune the operating point.

F.2 Oracle Diagnostic on Adaptive Trajectories

Post-Intervention Path-Survival Diagnostic. On the actual PAC-CF trajectory we ask a different question: among controller–task pairs that expose protected frontiers, did any inspected frontier lose all continuations accepted by the oracle? This event is downstream of intervention and therefore is not covered by Proposition 2.

Table 9: B100 oracle diagnostic on adaptive trajectories. "Survival" means that no inspected protected frontier loses all actions accepted by the oracle. Game24 ToolTree uses the canonical $C = 20$ replacement run, not the superseded $C = 32.5$ run.

Domain	Controller	Exposed	No loss of all protected actions	Survival
Game24	UCT-MCTS	100	97	97.00%
Game24	ToolChain*	97	94	96.91%
Game24	LATS	96	96	100%
Game24	ToolTree	100	99	99.00%
Game24	LevinTS	80	75	93.75%
Game24	pooled	473	461	97.46%
Blocksworld	pooled five controllers	495	494	99.80%

Observed Path Survival. As shown in Table 9, at B100, 461 of 473 exposed Game24 controller–task pairs retain at least one continuation accepted by the oracle at every inspected protected frontier, giving 97.46% pooled survival. Blocksworld records 494 of 495, or 99.80%. The Game24 rates for individual controllers range from 93.75% for LevinTS to 100% for LATS. These figures support the deployment diagnostic but do not extend the calibration trajectory finite-sample guarantee to frontiers induced by PAC-CF.

F.3 Sensitivity without ToolTree

Sensitivity Excluding ToolTree. Because ToolTree contributes a large positive utility effect, we remove it from the macro average across controllers while leaving the other four controllers unchanged. The latest accounting on the common set of completed pairs still gives positive reductions for all three realized workload metrics at every budget, whereas the utility shift moves from slightly positive at B100 to negative at larger budgets.

Table 10: Macro average across the three domains excluding ToolTree. Positive reduction means lower PAC-CF workload.

Budget	Δ utility	P-Req. $\downarrow$	Net Graph $\downarrow$	Token $\downarrow$
B100	$+0.265$	3.88%	8.75%	3.93%
B200	-0.086	6.41%	11.86%	6.32%
B300	-0.638	6.98%	12.43%	7.25%
B500	-0.855	7.54%	12.72%	7.72%

ToolTree Ablation Effects. Table 10 shows that, without ToolTree, physical request reduction rises from 3.88% at B100 to 7.54% at B500, Net Graph reduction from 8.75% to 12.72%, and token reduction from 3.93% to 7.72%. Over the same budgets, utility changes from $+0.265$ to -0.855 points. The robustness check therefore isolates two effects: the computation saving is distributed across controllers, while the positive cross-domain macro-average utility result is more sensitive to the inclusion of ToolTree.

F.4 Realized Workload Visualization

F.5 Controlled Gaussian BAI Boundary Validation

Empirical Boundary Transition. We evaluate the fixed-frontier confidence rule in a controlled Gaussian BAI environment with opposing bias at the maximum allowed magnitude. The frontier contains $M = 20$ candidates with gap $\Delta = 1$, noise level $\sigma = 0.2$, confidence level $\delta = 0.05$, and $\varepsilon = 0$. The optimal candidate receives constant bias $-L$, while every suboptimal candidate receives constant bias $+L$. We use the same time-uniform statistical radius as in the theoretical analysis and predeclare $10{,}000$ Monte Carlo trials per condition.

Table 11: Controlled Gaussian BAI validation around the sufficient separation boundary. Pruning is the fraction of suboptimal candidates eliminated at the predeclared $n = 5000$ evaluations per candidate.

L/Δ	$\Delta - 4L > 0$	Finite Cert.	Pruning	PAC-CF PCS
0.05	Yes	Yes	100.0%	1.00
0.15	Yes	Yes	100.0%	1.00
0.25	Boundary	No	0.0%	1.00
0.40	No	No	0.0%	1.00

Finite-Sample Boundary Shift. For balanced sampling, the sufficient finite-sample condition can be written as

$$4u_{\mathrm{stat}}(n) < \Delta - 4L - \varepsilon,$$

or equivalently,

$$\frac{L_{\mathrm{crit}}(n)}{\Delta} = \frac{1}{4} - \frac{u_{\mathrm{stat}}(n)}{\Delta} - \frac{\varepsilon}{4\Delta}.$$

With $\varepsilon = 0$, the predicted critical ratios are 0.166, 0.209, and 0.230 for $n = 200$, 1000, and 5000, respectively, approaching 0.25 as the statistical radius contracts.

Dense-Grid Transition. Across the predeclared bias grid, the empirical transition shifts toward the asymptotic boundary as the number of evaluations increases. At $n = 200$, pruning drops from 94.07% at $L/\Delta = 0.20$ to 0.72% at 0.22 and reaches 0.00% at 0.23. At $n = 1000$, pruning is 99.998% at 0.22, 41.29% at 0.23, and 0.00% at 0.24. At $n = 5000$, pruning remains 100.0% at 0.23, decreases to 56.03% at 0.24, and reaches 0.00% at 0.25. Thus the empirical transition moves toward $L/\Delta = 1/4$ as statistical uncertainty decreases.

Boundary Sufficiency versus Necessity. Empirical pruning can occur slightly outside the region certified by the uniform finite-sample condition. For example, at $n = 5000$ and $L/\Delta = 0.24$, the finite-sample certificate is negative while PAC-CF still eliminates 56.03% of suboptimal candidates. This does not contradict the theory: the condition $\Delta > 4L + \varepsilon$ is sufficient for reliable elimination rather than necessary for elimination on every realized sample. Across all predeclared conditions, the optimal candidate is never pruned, and PAC-CF PCS remains 1.00.

G Sensitivity across Models

Secondary-Model Recalibration. The secondary model experiment keeps the five controllers, budgets $B \in \{100, 200, 300, 500\}$, four evaluator observations per candidate, frozen $Q = .95$, and the same utility and uncensored paired workload accounting as the primary evaluation, while replacing the evaluator with `Ministral-3-14B-Instruct-2512`. Because the score gap distribution is model dependent, margins are recalibrated and frozen separately rather than transported from the primary model. Table 12 gives the score-gap thresholds used by the reported secondary model runs.

Table 12: Frozen secondary model score-gap thresholds $\widehat{\tau}_{.95}$ on the common $[0, 100]$ score scale.

Domain	UCT-MCTS	ToolChain*	LATS	ToolTree	LevinTS
Game24	85.0	85.0	85.0	45.0	75.0
Blocksworld	85.0	90.0	82.5	27.5	90.0
GTA	75.0	77.5	52.5	47.5	30.0

G.1 Calibration Quantile Sensitivity

Quantile–Margin Relation. To test whether the observed behavior can be reproduced by arbitrarily tightening the operational pruning margin, we recalibrate the same native-controller trajectories at $Q = .90$ while retaining $Q = .95$ as the primary operating point. Table 13 shows that lowering Q

can substantially reduce the calibrated margin $\widehat{\tau}_Q$. This sensitivity changes the operational conformal threshold rather than the theoretical bias allowance L.

Table 13: Secondary model score-gap thresholds at $Q = .90$ and $Q = .95$. Each entry reports $\widehat{\tau}_{.90}/\widehat{\tau}_{.95}$. Smaller margins permit more aggressive pruning.

Domain	UCT-MCTS	ToolChain*	LATS	ToolTree	LevinTS
Game24	80/85	80/85	77.5/85	32.5/45	65/75
Blocksworld	70/85	15/90	10/82.5	0/27.5	65/90
GTA	50/75	35/77.5	20/52.5	42.5/47.5	25/30

Quantile Utility–Efficiency Tradeoff. The smaller $Q = .90$ margins induce more aggressive pruning but do not uniformly preserve utility. Table 14 reports the endpoint budgets for Game24 and Blocksworld.

Table 14: Endpoint deployment sensitivity under the more aggressive $Q = .90$ calibration. Positive workload values denote reductions relative to Native search.

Domain	Budget	ΔU	P-Req. ↓	Net Graph ↓	Token ↓
Game24	B100	$+3.41$ pp	7.90%	17.36%	8.59%
	B500	$+0.41$ pp	16.10%	22.82%	16.93%
Blocksworld	B100	$+0.00$ pp	16.54%	34.18%	17.37%
	B500	-7.80 pp	22.33%	37.14%	23.40%

The more aggressive operating point increases workload reduction but can substantially degrade utility, as seen for Blocksworld at B500. Thus the utility–efficiency behavior depends materially on the calibrated operating point rather than on an arbitrary score-gap threshold. GTA $Q = .90$ margins are reported in Table 13, but a complete matched end-to-end $Q = .90$ GTA evaluation is unavailable.

Cross-Model Utility–Efficiency Contrast. Table 15 reports the complete secondary model budget sweep. Game24 occupies the strongest region: utility improves by $+0.80$ and $+0.40$ points at B100 and B200, then changes by -1.00 and -0.40 points at B300 and B500, while physical request, Net Graph, and token reductions remain between 13.37%–15.09%, 16.52%–17.84%, and 14.07%–15.77% across the four budgets. GTA shows a different frontier: utility changes remain within -0.15 to -0.59 points while the three realized reductions increase with budget, reaching 9.81%, 13.34%, and 11.97% at B500. Blocksworld is the least favorable domain for the secondary model: utility changes by -0.40 points at each budget, physical requests are nearly unchanged, and the principal effect appears in search structure, with Net Graph reduction decreasing from 3.81% at B100 to 2.75% at B500. The resulting macro average across the three domains retains positive workload reductions at every budget but moves from a $+0.08$-point utility change at B100 to -0.38 points at B500.

Matched Calibration Size. For the secondary model robustness study, each controller–domain margin is calibrated from 60 tasks with protected exposure at the prespecified $Q = .95$, giving $k_{.95} = 58$. The margin is frozen before deployment and is not selected from test outcomes. Gross subtree metrics remain unavailable for GTA because complete ancestry is not retained; realized request, Net Graph, and token accounting are unaffected.

H Frozen Configuration and Evaluation Scope

Frozen Configuration. Table 16 summarizes the implementation, calibration target, evaluator protocol, budgets, and paired analysis sets fixed before the matched Native/PAC-CF comparisons.

Table 15: Secondary model results at the frozen $Q = .95$ operating point across all budgets. Utility changes use the full paired test set; workload reductions use uncensored paired rows.

Domain	Budget	ΔU	P-Req. ↓	Net Graph ↓	E2E Token ↓
Game24	B100	$+0.80$ pp	13.37%	17.84%	14.07%
	B200	$+0.40$ pp	13.63%	16.52%	14.31%
	B300	-1.00 pp	15.09%	17.06%	15.77%
	B500	-0.40 pp	13.76%	16.72%	15.37%
Blocksworld	B100	-0.40 pp	-0.17%	3.81%	0.39%
	B200	-0.40 pp	-0.26%	3.18%	0.29%
	B300	-0.40 pp	-0.29%	2.83%	0.27%
	B500	-0.40 pp	-0.30%	2.75%	0.26%
GTA	B100	-0.15 pp	5.47%	9.49%	5.93%
	B200	-0.59 pp	6.60%	10.63%	7.70%
	B300	-0.46 pp	8.79%	12.41%	11.40%
	B500	-0.34 pp	9.81%	13.34%	11.97%
Macro average	B100	$+0.08$ pp	6.22%	10.38%	6.80%
	B200	-0.20 pp	6.66%	10.11%	7.43%
	B300	-0.62 pp	7.86%	10.77%	9.15%
	B500	-0.38 pp	7.76%	10.94%	9.20%

Table 16: Frozen implementation and evaluation configuration. These settings are fixed across the matched Native/PAC-CF comparisons unless a row explicitly states otherwise.

Item	Frozen setting
Controllers	UCT-MCTS, ToolChain*, LATS, ToolTree, LevinTS
Provider model	`deepseek-v4-flash`; seed 42
Budgets	$B \in \{100, 200, 300, 500\}$, measured in provider call equivalents per task
Main calibration	Direct conformal calibration for each domain–controller pair with $n = 60$ tasks with protected exposure per cell, $Q = 0.95$; deployed $\varepsilon = 0$
Frontier scoring	Four evaluator observations per candidate; matched evaluator temperature 0.2; deterministic shuffle set by the seed with paired forward/reverse order
Calibration labels	Exact verifier / solution verification information assigned after search; never exposed to online search
Online intervention	Eq. (7) only; native generator, downstream priority, rollout/backup, termination, and verifier unchanged
Efficiency accounting	Three primary metrics: physical requests, net graph nodes, and end-to-end tokens; gross physical and virtual nodes are supplementary; workload metrics use all paired tasks as the primary capped-deployment estimand; uncensored paired rows provide a natural-termination sensitivity analysis
Primary paired analysis sets	Utility and primary workload: Game24 $N = 100$, Blocksworld/GTA $N = 99$; natural-termination sensitivity: uncensored paired subsets within each domain–controller–budget cell.

Trace-Compatible Identifiers. The released CSVs retain the internal controller identifiers `vanilla`, `astar`, and `lts` for trace compatibility; they correspond to UCT-MCTS, ToolChain*, and LevinTS, respectively. The identifiers `lats` and `tooltree` already match their public names.

Three-Domain Macro. The primary macro evaluation uses Game24, Blocksworld, and GTA.

Calibration Rank Summary. Every controller–domain calibration uses $n = 60$ task scores with protected exposure. Tasks without protected exposure are treated as missing and the prespecified stream continues until 60 valid scores are collected. Hence $k_{.95} = 58$ and the exact rank-based miscoverage bound is $1 - 58/61 = 4.92\%$ for every cell. Table 17 reports only the quantity that varies across cells, the frozen score-gap threshold $\widehat{\tau}_{.95}$.

Table 17: Frozen $Q = .95$ score-gap thresholds $\widehat{\tau}_{.95}$. All cells use $n = 60$, $k_{.95} = 58$, and the same 4.92% rank-based miscoverage bound.

Domain	UCT-MCTS	ToolChain*	LATS	ToolTree	LevinTS
Game24	30.0	25.0	30.0	20.0	50.0
Blocksworld	45.0	50.0	62.5	0.0	10.0
GTA	40.0	40.0	40.0	30.0	30.0

The zero Blocksworld ToolTree margin means that its best protected action never fell below the frontier maximum on calibration tasks with protected exposure; it does not imply that other controller–domain pairs require no protection.

I Results by Controller

Controller Heterogeneity. Table 18 reports every controller included in the macro averages in Table 2. Utility deltas are PAC-CF minus Native, and positive reductions mean lower PAC-CF workload.

Table 18: Results by controller at the frozen $Q = .95$ operating point for B100 and B500. Utility deltas use the full test set; N_{eff} and workload reductions use uncensored paired rows.

Domain	Controller	B100					B500				
		ΔU	N_{eff}	PReq	NetG	Token	ΔU	N_{eff}	PReq	NetG	Token
Game24	UCT-MCTS	+1.00	75	+2.07%	+11.64%	+1.81%	+0.00	88	+2.41%	+15.72%	+2.24%
	ToolChain*	+0.00	57	+4.80%	+16.21%	+2.53%	-2.00	100	+8.69%	+19.73%	+6.38%
	LATS	+0.00	68	+0.00%	+4.49%	+0.00%	+0.00	100	+4.81%	+15.36%	+4.77%
	ToolTree	-3.00	99	+46.91%	-19.13%	+53.11%	-3.00	100	+46.89%	-21.34%	+52.53%
	LevinTS	-3.00	61	-0.12%	+4.95%	-0.38%	-4.00	100	+1.01%	+5.85%	+0.85%
Blocksworld	UCT-MCTS	+1.01	30	+2.26%	+8.53%	+2.29%	+3.03	47	+5.16%	+8.67%	+4.87%
	ToolChain*	+2.02	69	+4.14%	+3.68%	+4.01%	+0.00	99	+3.59%	+3.11%	+3.34%
	LATS	+0.00	28	+0.00%	+1.59%	+0.00%	-1.01	57	+0.00%	+0.31%	+0.00%
	ToolTree	+9.09	99	-4.30%	+31.53%	+0.04%	+9.09	99	-4.30%	+31.53%	+0.04%
	LevinTS	+3.03	61	+6.34%	+10.76%	+5.57%	-5.05	97	+24.26%	+29.74%	+24.06%
GTA	UCT-MCTS	+0.21	73	+22.55%	+34.27%	+25.91%	+0.03	93	+33.85%	+41.67%	+38.82%
	ToolChain*	+0.00	85	+0.94%	+1.69%	+1.46%	+0.00	97	+0.92%	+1.59%	+1.23%
	LATS	-0.32	68	+0.00%	+0.00%	+0.00%	-0.31	97	+1.34%	+3.38%	+1.47%
	ToolTree	+7.05	99	+14.24%	+26.29%	+18.14%	+7.05	99	+14.24%	+26.29%	+18.14%
	LevinTS	-0.77	86	+3.60%	+7.24%	+3.93%	-0.95	99	+4.46%	+7.52%	+4.59%

Utility–Efficiency Heterogeneity. At B500, GTA UCT-MCTS reduces physical requests by 33.85%, Net Graph nodes by 41.67%, and tokens by 38.82% with a $+0.03$ utility shift, whereas Game24 ToolTree reduces requests by 46.89% and tokens by 52.53% but loses three success points and increases Net Graph nodes by 21.34%. Blocksworld ToolTree shows yet another regime: utility improves by 9.09 points and Net Graph nodes fall by 31.53%, while physical requests increase by 4.30% and token use is nearly unchanged. No single workload metric therefore captures all deployment effects.

J Budget Censoring Sensitivity and Gross Subtree Accounting

Budget-Censoring Sensitivity. Exhausting the hard budget truncates the observed trajectory at the experiment cap rather than at its natural termination. Our primary capped-deployment analysis includes all paired tasks and therefore does not condition on an event that may be affected by the intervention. We additionally report the uncensored paired cohort as a descriptive natural-termination sensitivity analysis. Because cohort inclusion requires both runs to terminate before the cap and may therefore be affected by the intervention, we do not assign the uncensored-cohort comparison a causal interpretation. As shown in Table 2, across B100–B500, the reductions in the analysis over all tasks are 6.14–9.47% for physical requests, 9.03–12.40% for Net Graph nodes, and 6.84–10.25% for tokens; the corresponding uncensored paired values are uniformly larger but keep the same positive sign.

Table 19: B500 matched pruning and realized graph reduction on the uncensored paired cohort. Physical Nodes is gross matched native subtree suppression; Net Graph is realized end-to-end reduction after adaptive reallocation. "Matched roots" is the fraction of PAC-CF pruning roots whose Native ancestry is recoverable. GTA is omitted because complete node ancestry was not recorded.

Domain	Controller	N_{eff}	Physical Nodes ↓	Net Graph ↓	Matched roots
Game24	UCT-MCTS	88	15.79%	15.72%	99.66%
	ToolChain*	100	21.61%	19.73%	98.96%
	LATS	100	15.34%	15.36%	100.00%
	ToolTree	100	11.42%	−21.34%	81.44%
	LevinTS	100	7.60%	5.85%	95.51%
	Macro	–	14.35%	7.06%	95.11%
Blocksworld	UCT-MCTS	47	11.91%	8.67%	89.47%
	ToolChain*	99	6.95%	3.11%	100.00%
	LATS	57	0.31%	0.31%	100.00%
	ToolTree	99	57.18%	31.53%	56.25%
	LevinTS	97	30.58%	29.74%	100.00%
	Macro	–	21.38%	14.67%	89.14%

K Paired Statistical Analysis

Paired Bootstrap. All statistics in this section are offline analyses of the frozen task level artifacts; they introduce no new LLM or API calls. The primary analysis uses Game24 $N = 100$ paired tasks and Blocksworld/GTA $N = 99$ paired tasks. Utility intervals for individual conditions use 200,000 nonparametric bootstrap resamples over paired tasks. Domain level intervals use 300,000 resamples with task identities shared across the five controllers before averaging controllers with equal weights. Summaries across the three domains and for ToolTree use 300,000 independent paired resamples within each domain followed by averaging across domains with equal weights. Workload reductions are ratios of totals over uncensored paired rows inside each resample before controller/domain averaging; end-to-end token usage is resampled in the same way. Utility remains on the full task set. Rescue/regression/tie (R/G/T) counts compare PAC-CF and Native task level utility, with exact ties defined at numerical tolerance 10^{-12}.

Table 20: Paired bootstrap averages by domain. Utility intervals use the full paired test set; workload intervals use uncensored paired rows. Positive reduction means lower PAC-CF workload.

Domain	Budget	Δ utility [95% CI]	PReq [95% CI]	NetG [95% CI]	Token [95% CI]
Game24	B100	−1.00 [-2.60,0.60]	+10.73% [8.92,12.34]	+3.63% [0.40,6.50]	+11.41% [9.66,12.95]
Game24	B200	−1.00 [-2.60,0.40]	+12.31% [10.58,13.90]	+6.78% [3.26,9.87]	+12.76% [11.08,14.27]
Game24	B300	−1.60 [-3.00,-0.40]	+12.56% [10.98,14.01]	+7.01% [3.71,9.91]	+13.08% [11.50,14.50]
Game24	B500	−1.80 [-3.20,-0.60]	+12.76% [11.19,14.22]	+7.06% [3.80,9.95]	+13.35% [11.76,14.81]
Blocksworld	B100	+3.03 [1.41,4.85]	+1.69% [-0.73,4.11]	+11.22% [9.06,13.46]	+2.38% [0.07,4.70]
Blocksworld	B200	+2.22 [0.81,3.64]	+5.04% [2.06,7.97]	+14.17% [11.58,16.70]	+5.62% [2.72,8.45]
Blocksworld	B300	+1.41 [-0.20,3.03]	+5.43% [2.70,8.20]	+14.67% [12.28,17.13]	+6.12% [3.46,8.83]
Blocksworld	B500	+1.21 [-0.61,3.23]	+5.74% [2.92,8.59]	+14.67% [12.25,17.17]	+6.46% [3.71,9.24]
GTA	B100	+1.23 [0.33,2.24]	+8.27% [6.75,9.75]	+13.90% [11.74,16.02]	+9.89% [8.04,11.67]
GTA	B200	+1.20 [0.29,2.20]	+9.38% [7.63,11.05]	+14.81% [12.49,17.07]	+10.92% [8.84,12.89]
GTA	B300	+1.28 [0.37,2.29]	+10.12% [8.22,11.97]	+15.46% [13.05,17.81]	+12.35% [9.61,14.95]
GTA	B500	+1.17 [0.26,2.17]	+10.96% [8.58,13.23]	+16.09% [13.36,18.74]	+12.85% [9.98,15.45]

Domain-Level Utility–Efficiency Heterogeneity. GTA has positive utility intervals at all four budgets, from $+1.23$ $[0.33, 2.24]$ at B100 to $+1.17$ $[0.26, 2.17]$ at B500, together with positive intervals for all workload metrics. Blocksworld is resolved positive in utility at B100 and B200 but becomes unresolved at B300 and B500. Game24 instead has resolved negative utility at B300 and B500, with intervals $[-3.00, -0.40]$ and $[-3.20, -0.60]$, even though all three workload intervals remain positive. The aggregate result should therefore be interpreted as an observed efficiency pattern with a utility tradeoff that varies by domain.

Table 21: Independent summary across the three domains for the already pruning-aware ToolTree controller. Utility uses the full test set; workload intervals use uncensored paired rows.

Budget	Δ utility [95% CI]	PReq [95% CI]	NetG [95% CI]	Token [95% CI]
B100	$+4.38$ [1.43,7.41]	$+18.95\%$ [14.81,22.96]	$+12.90\%$ [7.54,17.77]	$+23.76\%$ [19.56,27.85]
B200	$+4.38$ [1.43,7.41]	$+18.94\%$ [14.84,22.93]	$+12.16\%$ [6.62,17.20]	$+23.57\%$ [19.39,27.65]
B300	$+4.38$ [1.43,7.40]	$+18.94\%$ [14.84,22.93]	$+12.16\%$ [6.67,17.19]	$+23.57\%$ [19.39,27.65]
B500	$+4.38$ [1.44,7.41]	$+18.94\%$ [14.83,22.94]	$+12.16\%$ [6.66,17.20]	$+23.57\%$ [19.40,27.65]

ToolTree Tradeoff by Budget. Table 21 shows that, for the already pruning-aware ToolTree controller, the average utility gain across domains remains $+4.38$ points with a strictly positive 95% interval at every budget. Physical request reduction stays at about 18.94–18.95%, Net Graph reduction at 12.16–12.90%, and token reduction at 23.57–23.76%. This aggregate pattern does not imply that every ToolTree domain improves: the table by controller shows the negative Game24 utility case and the positive Blocksworld/GTA cases that combine into the average across domains.