SELECTIVE CERTIFICATION FOR SAFE UPGRADE PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

An upgrade planner should spend evaluation on plausible routes, distinguish a rejected candidate from a certified obstruction, and justify any reuse of evidence. We instantiate explore–verify in a known finite graph whose edits share unknown gain signs. The algorithm screens only labels on optimistic candidate routes, then certifies either a useful positive route or a negative blocking set. It stops with finite expected cost on every sign assignment, including infeasible ones, and attains the minimum positive-route information cost on each fixed feasible instance. Its screening cost depends on labels reached by correct optimistic search. A checkpoint contract extends the same procedure to irreversible diagnostic access. A separate packet rule permits adaptive route selection after partially validated sharing, with an explicit combined safety budget. Experiments include 160 random mixed-sign graphs, irrelevant-route and ordering controls, checkpointed deployment, and solver-certified optima on larger graphs. Selective screening removes large all-label overhead, but simpler testing often wins at moderate confidence; an optional rule retains that baseline. A code study that charges all work and matches candidate order reduces executions by 22.1% while taking longer on these cheap programs. These guarantees apply to checked upgrades, not general autonomous self-improvement.

1 INTRODUCTION

A proposed software upgrade can be useful, unsafe, or too uncertain to evaluate economically. A sequence of upgrades adds a planning problem: a fact checked for one edit may support several later edits, while a deployment can remove the data or interface needed for future checks. We study this component with a fixed objective and a protected acceptance rule. The inputs are a finite candidate graph and a stated evidence model. Discovering the graph or learning a transfer relation is a separate task.

Self-modifying artificial intelligence (AI) systems motivate checked upgrades, from proof-based agents to the Self-Taught Optimizer (STOP), the Darwin Gödel Machine and Hyperagents (Schmidhuber, 2003; Zelikman et al., 2024; Zhang et al., 2025; 2026). Janus already co-evolves candidate programs and proxy evaluators with external validation for promotion (Liu et al., 2026). Here the evaluator accumulates observations within a fixed statistical model. Calling that general evaluator co-evolution would overstate the abstraction. Our question is: once possible upgrades are specified, which evidence must be collected before deploying them?

Choosing a route and safely deploying it need different confidence levels. A mistaken route guess can be discarded; a harmful deployment cannot be undone by collecting evidence later. This distinction is established by the explore–verify framework of Karnin (2016), including its high-confidence exploration/verification cost split. Multiple-answer exploration also provides general asymptotically optimal methods (Degenne & Koolen, 2019), with recent non-asymptotic analyses (Poiani et al., 2026). Our work is an explicit realization for utility-qualified label paths with a one-sided deployment constraint. It does not introduce explore–verify or a new general sample-complexity principle.

The graph raises two questions: which labels to test, and what to do when no route remains. Screening every label may waste most of the budget on irrelevant alternatives. Permanently rejecting an edit after a cheap test can miss a useful route; restarting indefinitely fails to handle the common outcome that no safe upgrade exists. We instead screen optimistic candidate routes, retain screening advice

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.

within an attempt, and use a high-confidence negative blocking certificate to justify abstention. This preserves the positive-route leading cost while making infeasible inputs terminate.

The same procedure supports two interfaces. At a checkpoint, all diagnostics for the next upgrade block are available before commitment; under separable blocks, local screening attains the access-constrained cost. A validated sharing packet supplies actual member identities, not permission to reuse a certificate on whichever unchecked edits the optimizer likes. Its risk combines with the sign-testing budget even after adaptive planning. These interfaces clarify which components compose and which guarantees remain restricted.

The experiments test the practical limits of this construction. We compare selective and full screening, symmetric testing, and cheap rejection on random mixed-sign graphs across four confidence levels, with both feasible and infeasible instances. We measure planning gaps against certified optima. Finally, we compare source verification against direct execution on exactly the same returned edits and gain. The resulting case for selective certification is conditional: it avoids unnecessary exploration and handles abstention correctly, but it is not uniformly the cheapest finite-confidence policy.

2 A PRECISE UPGRADE CONTRACT

Graph, labels, and queries. Let $G = (V, E)$ be a known finite directed acyclic graph (DAG) with parallel edges with start s and terminal set T. An edge e deploys a persistent change with gain $b_e\theta_{j(e)}$, where $b_e > 0$, its evidence label $j(e) \in [p]$, and $\theta_j \in \{-\gamma_j, +\gamma_j\}$ with known $\gamma_j > 0$. Edges sharing a label share an unknown gain sign. This can be represented by $x \leftarrow x + b_e e_{j(e)}$ under $R(x) = \theta^\top x$. Exact sharing is part of the statistical contract, not a fact inferred from similar names or outputs on a few examples.

A trusted query of j returns an independent $Y \sim \mathcal{N}(\theta_j, \sigma^2)$, with known σ. All observations count, including screening, failed attempts and obstruction checks. The policy sees only its declared inputs and charged observations; evaluation harnesses alone see true signs and actual gain. Global access permits every diagnostic before deployment. With local access, j can be queried only at a current node v satisfying $j \in Q(v)$. A collected certificate can be retained; a past opportunity to query cannot.

Safety, gain, and an explicit negative answer. Write $g_e = b_e\gamma_{j(e)}$ for the edge gain if its sign is positive. A route is eligible if it ends in T and $\sum_{e\in P} g_e \geq u$, for a prescribed $u \geq 0$. A positive eligible route is a useful upgrade. The policy must control the probability of any negative-gain deployment, not just final net loss. In this paper its final answer is either a certified positive eligible route or a certificate that no such route exists. A budget interruption is a third, *undecided*, outcome. Safe completion, a terminal flag, and retained actual gain are reported separately.

For distinct path labels $U(P)$, define

$$\ell = \log(1/\delta), \quad c_j = \frac{\sigma^2}{2\gamma_j^2}, \quad C_\theta(u) = \min_{P \text{ positive eligible route}} \sum_{j\in U(P)} c_j. \tag{1}$$

The minimum is infinite on an infeasible instance. The information per observation distinguishing the two allowed signs is $1/c_j$. On a positive route a label is paid once, even if used repeatedly. Varying u changes which routes qualify and therefore gives a utility–query tradeoff.

Worked decision. In Figure 1, when both signs are positive, target $u = 0.7$ permits the cheaper lower route with leading cost $(1/2)\ell$. Target $u = 0.9$ instead costs $(3/2)\ell$. If a screens negative, the lower route remains an option only for the smaller target. For the larger target, a high-confidence negative a-certificate proves infeasibility even if b is positive. This is a utility-qualified blocking set, not necessarily an ordinary graph cut that disconnects every terminal. If the upper route is selected, both certificates can be obtained at s before any change, so the missing interface at v does not prevent this particular execution.

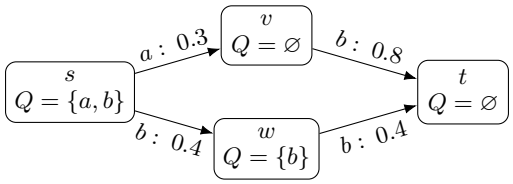

Figure 1: Edges show label and positive-sign gain; nodes show diagnostic access. With $c_a = 1, c_b = 1/2$, the lower route needs only label b and gains 0.8. A target $u = 0.9$ requires the upper route and both signs. Its b-evidence must be collected before leaving s. A screened negative is advice to replan, not yet a certificate that no eligible positive route exists.

3 SELECTIVE EXPLORE–VERIFY WITH CERTIFIED ABSTENTION

A sign test uses $Z_{j,n} = 2\gamma_j \sum_{i \le n} Y_i/\sigma^2$. A fresh test at $(+A, -B)$ falsely accepts a negative sign with probability at most e^{-A} and rejects a positive sign with probability at most e^{-B}. These are classical likelihood-ratio guarantees (Wald, 1945; Wald & Wolfowitz, 1948). Let $\ell_0 = \max\{2, \ell\}$ and set

$$B = \log(4p\ell_0), \qquad A_h = \log \frac{p\pi^2 h^2}{6\delta}, \quad h = 1, 2, \ldots. \tag{2}$$

Each epoch has fresh streams. Screening information is retained within the epoch, but never authorizes deployment.

Selective–Certify, epoch h
1. Plan the cheapest eligible route avoiding screened-negative labels. Keep original c_j costs for screened positives.
2. Screen only its unscreened labels at $\pm B$. Replan immediately after a negative screen. Each label is screened at most once per epoch.
3. Once the route is fully screened positive, test its labels on fresh streams at $(+A_h, -B)$. Deploy only if every verdict is positive; otherwise restart.
4. If no eligible route remains, prune the screened negatives to a blocking set. Test its members on fresh streams at $(+B, -A_h)$. Output infeasible only if all verdicts are negative; otherwise restart.

Theorem 1 (Global access: positive route or verified obstruction)**.** *In the fixed known-gap model,* SELECTIVE–CERTIFY *stops with finite expected query count on* every *sign assignment. With probability at least* $1 - \delta$*, all deployed edges are positive and the final answer is correct. On each fixed feasible* θ*, it attains*

$$\mathbb{E}_\theta N = C_\theta(u)\ell + O_\theta(\log \ell), \qquad \delta \downarrow 0. \tag{3}$$

More precisely, let S_θ *be the labels reached by optimistic route screening with correct signs and the fixed ordering rule. Then*

$$\mathbb{E}_\theta N \le C_\theta(u)\ell + \left(\sum_{j \in S_\theta} c_j \right) \log(4p\ell_0) + O_\theta(1). \tag{4}$$

Random ordering replaces the parenthesized quantity by its expectation. On infeasible instances the expected cost is $O_\theta(\ell + \log \ell)$*; no optimal infeasibility constant is claimed.*

Why the changes preserve the guarantee. In an epoch, any wrong screen has probability at most $1/(4\ell_0)$. On correct screens, the optimistic candidate set contains every positive eligible route. Thus the first fully positive route has cost exactly $C_\theta(u)$, without testing unrelated labels. If no route remains, the screened negatives meet every eligible route and can be verified as an obstruction. A false deployment or false abstention requires a wrong strong verdict. At most p strong tests occur per epoch, so their allowances sum to $\sum_h pe^{-A_h} = \delta$ over *both* answer types. An epoch gives a correct terminating answer with probability at least $1 - 1/(2\ell_0)$, whether the instance is feasible or not.

On a feasible instance, expensive obstruction verification occurs only after a wrong screen. Its probability is $O(1/\ell)$ and cost $O_\theta(\ell)$, adding only $O_\theta(1)$ expected work. A label outside S_θ is

queried only after search has already deviated through a wrong screen. Independence of its still-unused stream yields the route-local term in (4). Appendix B.3 gives the full argument, including finite expected stopping. The lower bound flips each deployed positive label and applies the standard stopped-transcript information argument (Kaufmann et al., 2016); it is a graph specialization of multiple-answer exploration, not a new general lower bound.

Screening is optional, not automatically cheap. At moderate confidence, B can be comparable to A_1. Our implementation chooses symmetric testing at $\log(p/\delta)$ whenever $B \geq A_1/2$, before seeing observations. Otherwise it uses selective screening. In the skip region this is exactly the symmetric baseline, including its sample path under paired streams. For every fixed instance, $B/A_1 \to 0$, so the rule preserves the asymptotic theorem. The threshold factor was fixed before the experiments. It is not an optimal tuning rule and does not imply additive-$O(1)$ dominance outside the skip region.

Three possible outcomes from the same observations. Consider target $u = 0.9$ in Figure 1. Suppose a screen says that a is negative, so the only eligible route is excluded. The planner first checks whether that one negative label blocks every eligible route. It then runs a *new* test of a with the strong boundary on the negative side. A negative certificate ends the entire procedure with a checked obstruction. A positive verdict instead rejects this obstruction and starts another attempt. Neither the screen nor the contradiction itself is deployed as an edit. Under global access there is no deployment until a full route has passed its positive gates; under checkpoint access this rule applies to the current block.

This separation matters when a label is hard to test. A short, unlucky screen can cause extra search, but cannot permanently delete a useful route. Conversely, when the edit really is negative, repeated screening need not continue indefinitely: one reliable negative certificate is enough for that blocking set. If testing runs out of its allotted budget before either verdict, the output is undecided. That distinction lets a deployment system retain its current version without claiming to know that all safe improvements are impossible.

What is gained by selective rather than full screening? The factor $\sum_{j \in S_\theta} c_j$ describes an explicit search trace, not just the winning route. It includes labels examined on cheaper routes that are eventually rejected, so route selection is not free. It excludes a more expensive alternative that never becomes an optimistic minimum. Adding many such alternatives still increases the logarithmic familywise boundary and can affect exceptional-event constants, but does not force their ordinary screening costs into every attempt. This is the property tested by the expensive-distractor experiment. The theorem fixes the graph and nonzero gaps; it should not be extrapolated to a growing graph by ignoring those constants.

Where the method fits in prior theory. Theorem 3 and Corollary 4 of Karnin (2016) already yield an exploration term proportional to $\log\log(1/\delta)$ and a high-confidence verification term. Sticky Track-and-Stop covers multiple acceptable answers (Degenne & Koolen, 2019). Here the explicit answer is a utility-qualified path or a verified obstruction, the exploration oracle is lazy label routing, and only certified positive answers are deployed. Infeasibility can also be encoded in a structured answer space; it is not claimed to be beyond those frameworks. This scope is more precise than presenting the screen/verify separation as a new principle.

4 One procedure under access and sharing contracts

A route is *diagnostic-admissible* if every required label is available at a visited prefix before its first deployment. Restricting (1) to such routes defines $C_\theta^{\mathrm{acc}}(u)$. General irreversible access does not guarantee progress: two initially indistinguishable branches can each expose only their own diagnostic, with the positive and negative continuations exchanged between worlds. One branch choice then fails in at least one world. Appendix D proves this obstruction and the access-dependent information lower bound.

A sufficient condition is more useful than a universal claim. Consider a chain of *checkpoint blocks*: all routes through a block rejoin the same next checkpoint; label sets are disjoint across blocks; and every block label can be checked at its entry before deployment. Intermediate access may be empty.

Eligibility either factors into specified block gain quotas, or all routes in each block have the same positive-sign gain and the global target is no larger than their total. The latter is a special case of the original utility constraint, not a change of objective. Run the same selective algorithm at each entry with error allowance δ/K for K blocks, then deploy the certified block route. A later obstruction preserves earlier positive gain.

4.1 What the optimizer may reuse

Exact labels may be supplied by a verified relation. A partial check needs a different interface. Fix a family of F members before checking. Inspect m uniformly sampled members without replacement. If every inspection validates a sufficient transfer relation, issue r uniformly sampled unchecked identities as a *packet*; otherwise issue no unchecked identity. If there are a fixed unknown v invalid members, the joint risk of an invalid packet is

$$R(F, m, r; v) = \frac{\binom{F-v}{m}}{\binom{F}{m}} - \frac{\binom{F-v}{m+r}}{\binom{F}{m+r}}. \tag{5}$$

Choose the package and allowance β_t before its sample, with $\max_v R \le \beta_t$ and total $\sum_t \beta_t \le \beta$. After the draw, the route planner may use any subset of the issued packet, but may not substitute preferred unchecked identities. The whole packet's validity event protects every subset, including adaptively selected ones.

Theorem 2 (Supported composition)**.** *Under exact sharing and the checkpoint contract, the blockwise selective policy has finite expected stopping and joint correct-output/non-regression probability at least $1 - \delta$. On every fixed feasible instance it attains $\mathbb{E}_\theta N = C_\theta^{\mathrm{acc}} \ell + O_\theta(\log \ell)$, where C_θ^{acc} is the sum of the block costs; in the equal-gain special case it equals $C_\theta^{\mathrm{acc}}(u)$.*

More generally, with validation packets as above, sound deterministic proofs, and sign tests using total allowance δ, any adaptive planner restricted to the authorized members has harmful-deployment probability at most $\delta + \beta$. This composes safety, not the exact query constant or completion when validation removes routes.

The checkpoint cost follows from label-disjoint factorization; local observability alone does not give that factorization. The packet claim follows by conditioning on the history *before* each draw and taking a union bound. Both proofs, the independence requirements for diagnostic noise, and the planner interface are in Appendix C. The planner can compare direct checks, an applicable proof, and a risk-feasible package using explicit verification costs. Our package example minimizes nominal all-valid cost for exchangeable slots; named edits must instead be represented by their actual issued identities.

A joint allowance is not a conditional certificate. For $F = 64, m = 32, r = 3, v = 1$, the joint risk is $3/64 \approx 4.69\%$, but conditional on clean inspections it is $3/32 \approx 9.38\%$. Redrawing until inspections pass without allocating another allowance is invalid. Further, an invalid sufficient relation need not mean negative utility. The unchecked-release coupling bound $\mathbb{E}U_0 \le F\beta$ applies to invalid-release control; interpreting it as a harm lower bound requires indistinguishable single-member alternatives that can be harmful. Appendix F states these alternatives and a confidence-inversion alternative for conditional use.

Concrete scope of access and code verification. A no-snapshot migration can delete pre-migration rows needed for diagnostics even if source code is rolled back. Our SQLite fixture checks that behavior by executing the migration and pre-migration queries. Declared query/data dependencies determine its access sets; retaining a snapshot changes the contract. Separately, our integer-expression checker proves exact output equivalence, a stronger sufficient relation than shared gain sign. It does not establish transfer between distinct, non-equivalent fixes. Both examples are controlled fixtures, not measurements of an autonomous upgrade loop.

5 Experiments

We use independent graph-seed clusters and paired per-label observation streams; parameter settings generated from the same seed stay together in uncertainty calculations. Proposed policies never

Table 1: Random global-access graphs: mean total queries on the 120 feasible graphs (eight streams each). Lower is better, with the same eligibility target. Optional equals symmetric at $\delta = 0.05$; its other values reflect the prespecified skip rule, not selection using outcomes.

δ	Symmetric	Cheap reject	Full	Selective	Optional
0.05	16.25	16.95	89.44	29.25	16.25
10^{-4}	29.79	28.00	111.77	41.82	34.96
10^{-8}	49.56	43.64	134.50	58.44	58.44
10^{-16}	89.55	75.30	172.11	90.06	90.06

Table 2: Mean queries on the 40 infeasible random instances, eight streams each. Every output in this subset is either a separately checked obstruction or an explicitly counted error. Selective screening is not always cheapest, but no longer restarts indefinitely.

δ	Symmetric	Cheap	Full	Selective	Optional
0.05	23.73	37.36	88.38	36.04	23.73
10^{-4}	43.48	57.36	111.13	50.25	47.28
$1e-08$	72.73	85.73	133.21	68.47	68.47
$1e-16$	131.18	140.29	171.16	102.19	102.19

receive true signs. Every query counts; budget or solver exhaustion is retained rather than reported as success. Confidence intervals below are pointwise paired bootstrap intervals over graph seeds, averaging repeated observation streams within each graph. Runtime is descriptive. The main comparisons use the same exact route objective and the same gain target; differences in the testing policy are not differences in available information.

5.1 Random mixed signs and irrelevant alternatives

We generate 160 graph instances: 80 at each of $p = 12$ and $p = 24$, depth six and width three. The two label-count settings reuse the same 80 seed clusters. Independent signs are negative with probability 0.35. Edge gains vary; half the instances require $u = 0$ and half $u = 2.5$. No positive route is planted. There are 120 feasible and 40 infeasible graphs. Each graph uses eight independent observation streams, the four δ values in Table 1, and randomized label-test order. Full and selective screening share the same fresh verification and negative-certificate stopping rule. Cheap rejection tests candidate labels directly with a cheap negative boundary and fresh-epoch safeguards, rather than first screening the route.

Selective screening sharply reduces the full-screen overhead, but cheap rejection is cheaper on these random feasible graphs across the displayed range. At $\delta = 10^{-16}$, selective uses 90.06 queries versus 172.11 for full screening, 89.55 for symmetric testing and 75.30 for cheap rejection. This is evidence against a universal practical ranking. On the 40 infeasible graphs, selective uses 102.19 queries versus 131.18 for symmetric testing at the same confidence. All runs terminate within the explicit implementation caps. At $\delta = 0.05$, selective and full screening have no observed wrong output; symmetric and optional have two harmful outputs and one false abstention among 1,280 runs each. At the smaller displayed δ values, no method has an observed incorrect final answer. These counts do not empirically certify tiny error probabilities.

Table 2 separates the common "no useful route" outcome from successful upgrade search. At 10^{-16}, selective saves 22.10% [19.07,24.91] relative to symmetric testing on these instances. On feasible instances, its savings relative to *full screening* are 67.29% [64.77,69.74] at 0.05 and 47.67% [45.16,50.18] at 10^{-16}. Those comparisons identify the overhead removed by route restriction; they do not establish that the screening phase is always worth purchasing. In particular, the optional policy deliberately gives up those reductions relative to full screening at moderate confidence in order to match the less costly symmetric baseline exactly.

The error and cost questions also differ. High-confidence negative verification spends additional queries to distinguish a genuinely blocked instance from one with a mistakenly rejected positive route. Counting an unverified negative screen as a cheap correct abstention would improve its measured

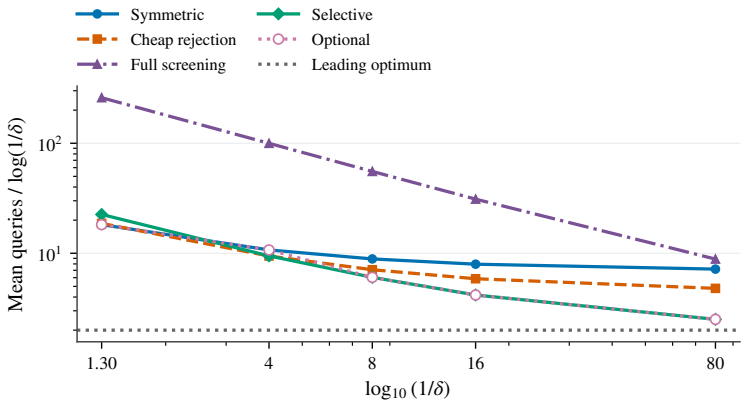

Figure 2: Twenty expensive alternatives that correct route-restricted screening never needs. All query phases count; 300 paired streams, positive-first stress order. The optional rule selects symmetric testing at moderate confidence. Both axes use logarithmic scales to retain the full-screen cost without cropping it.

cost by changing the output contract. Our comparisons hold that contract fixed. Appendix J reports the per-phase query counts, error events and paired protocol so that this distinction is visible.

The controlled stress test adds 20 expensive positive alternatives to five rejected two-edit routes and one useful route. Their costs exceed the useful route's cost, so correct optimistic search never screens them. At $\delta = 10^{-16}$, mean queries divided by ℓ are 31.04 for full screening, 7.96 for symmetric testing, 5.87 for cheap rejection, and 4.18 for selective screening. At 10^{-80} the selective value is 2.51 versus a leading constant of two. There are 300 paired streams per condition. Randomizing test order reduces the trap's advantage but does not remove the distinction: without distractors at 10^{-80}, selective and cheap rejection have normalized costs 2.38 and 3.49. The original positive-first order is therefore a stress condition, not the only evidence.

Choosing a finite-confidence testing policy. The optional rule is a reproducible default for avoiding the obvious case in which a screening boundary is as demanding as deployment verification. It is not a learned selector trained on these graphs. A deployment system with a known query budget may instead precommit to symmetric testing or another safe method, keeping the same negative-answer contract. Selecting a policy from a completed benchmark after observing its performance does not justify reusing that benchmark as fresh evidence. The policy comparisons here use paired data to measure costs, not to allocate uncharged confidence adaptively.

Screening helps when ruling out a route early saves enough later verification to cover its cost. The rejected-route construction amplifies that effect, while many random graphs have a useful cheap route and little wasted positive certification to avoid. In that setting the extra screen and fresh gate can remain more expensive even after irrelevant labels are removed. The theorem describes growing confidence on a fixed instance; it does not identify the cheapest policy at a fixed budget.

5.2 Local-access deployment and risk composition

The checkpoint suite uses 120 independently generated four-block instances, with eight label-disjoint diagnostics per block and mixed signs. All block routes have known gains $0.5, 0.75, 1, 1.25$, respectively; a completed route gains 3.5. Every diagnostic is available at the block entry and unavailable in its interior. These are the sufficient-condition instances of Theorem 2, not arbitrary random access sets. There are 26 globally feasible instances, and no route is inserted to guarantee feasibility. Each uses eight streams, two confidence levels, and budgets of 20, 60 and 200,000 queries.

At $\delta = 0.05$ and budget 60, optional and symmetric policies retain mean gain 1.15 and safely complete 14.7% of all instances; selective retains 0.75 and full screening 0.35. At $\delta = 10^{-8}$ without an effective budget restriction, all policies preserve the same mean gain 1.26, including useful prefixes

Table 3: Global-access route-cost gaps to solver-certified optimum, 100 graphs per label count (50 each at widths three and six). Entries are mean percentage gaps; brackets are paired 95% intervals over 50 seed clusters, keeping the two widths together. Linear programming (LP) gives a lower bound, not a competing deployable policy. Complete access-specific results and search-budget sensitivity are in the appendix.

Labels	Beam-32	Greedy	Shortlist-32
32	0.07 [0.00, 0.17]	21.30 [17.78, 24.85]	23.35 [19.37, 27.62]
64	0.20 [0.00, 0.53]	23.47 [19.95, 27.30]	26.04 [22.63, 29.47]
96	0.32 [0.09, 0.59]	20.01 [16.57, 23.54]	21.81 [17.99, 25.67]

on infeasible instances. Selective spends 110.75 queries versus 172.06 for full screening and 96.68 for symmetric testing. This suite validates the common access/testing contract and exposes its finite-confidence costs; it does not claim that screening wins merely because access is restricted. Appendix K.2 evaluates a separate symmetric-ledger replanning baseline under arbitrary access; it does not inherit the checkpoint theorem.

For transfer composition, fixed 64-member families receive packages $(m, r) = (32, 3)$ at $\beta = 0.05$, followed by sign certification at $\delta = 0.01$. The planner selects two preferred identities *within* the packet; dirty inspections instead permit checked members. We vary representative sign and violation count, with 3,000 draws each. With one violation and a positive representative, harmful release occurs in 4.93% of runs, below the combined 6% allowance; clean inspections occur in 49.8%. The exact packet risk is 4.69%, whereas its risk conditional on clean inspections is 9.38%. The experiment checks composition and selection, not a new estimation rate. A nominal package optimizer chooses direct checking for demands of three or eight members, and only three unchecked slots for larger demands. Sparse validation does not justify bulk reuse.

5.3 Planning against certified optima

Distinct-label routing is computationally difficult: the classical label-path and weighted-cover connections imply exponential exact state expansions and inherited logarithmic approximation hardness (Carr et al., 2000; Hassin et al., 2007; Dinur & Steurer, 2014). Our statistical theorem assumes exact planning. A feasible approximate route retains the same certification safety, but not the same query coefficient. We therefore measure route-cost gaps against a certified mixed-integer programming (MIP) optimum. The formulation, linear programming (LP) relaxation, feasibility checks and solver status are provided in Appendix G.

The study includes label counts 32, 64 and 96, depth ten, widths three and six, and global or local access. Beam widths 8, 32 and 128 and shortlist sizes 32 and 128 make search budget explicit. Across 600 instances, the solver certifies 583 optima and 17 infeasibilities, with no unresolved status. Mean solving time is 1.06 seconds and the maximum is 5.34 seconds. We retain every failed heuristic search. Table 3 reports paired gaps to these optima. The global LP bound lies 30.86%, 25.53% and 20.84% below optimum on average at the three label counts. Beam-128 reaches the optimum on all measured global instances; this is empirical, not a general approximation guarantee. An ablation using finite-confidence expected-test-length weights saves only about 0.07% at $\delta = 0.05$ (Appendix K.3).

The solver comparison measures the cost of a chosen evidence set, before random query lengths are realized. A 1% route-cost gap therefore means a 1% gap in the asymptotic verification objective, not automatically a 1% finite-sample wall-clock penalty. When two methods select the same label set they share the same statistical query problem. When they choose different sets, boundary overshoot, failure ordering and actual signs can matter, as the random mixed-sign study shows. We keep those testing questions separate from the deterministic optimization gaps rather than treating an all-positive planning experiment as evidence about selective screening.

5.4 Order-matched verification from actual source execution

The code study contains four fixed arithmetic tasks and 64 candidate expressions per task. A four-input fingerprint proposes a family. A restricted integer-ring checker can prove some candidates exactly equal to a verified representative; unproved candidates are executed on the full 256-input

Table 4: Primary code comparison: identical returned edits, order and gain on each task/ordering. Both methods use 256 cached reference answers. Abstract syntax tree (AST) work is reported separately. Counts and total runtime include the same fingerprint and representative/baseline work; the hybrid additionally pays for on-demand proofs. Means over four fixed tasks and 500 orderings per task. Orderings are not independent programming tasks.

Policy	Executions	AST nodes	Gain	Time (ms)
Direct	3258.88	0.00	3.24	2.46
Order-matched hybrid	2538.24	208.77	3.24	2.98

domain. The checker treats non-ring expressions as opaque, requires justified totality, and does not infer equivalence from the fingerprint (Necula, 1997). Boundary mutations create actual execution differences. This tests a narrow sufficient relation, not non-equivalent edits that happen to share a utility sign.

The timed procedure compiles and executes each requested candidate, obtains the common reference answers once, and computes proofs on demand. The hybrid returns exactly the same eight edits and gain as direct validation in every ordering. It reduces candidate executions from 3,258.88 to 2,538.24, or 22.1%, while adding 208.77 examined abstract syntax tree (AST) nodes on average. On these cheap programs its measured total runtime is *higher*, not lower. The deployment protocol and task bank are controlled fixtures; reusing one verified implementation everywhere would be a different, often simpler engineering contract.

6 Scope and conclusion

The graph and label map, nonzero two-point magnitudes, and Gaussian noise law are strong inputs. The checkpoint result requires common exits, disjoint labels and factored eligibility; it is not an optimality theorem for arbitrary lost access. Statistical sharing packets control joint release risk and may remove useful routes, while the code checker establishes exact equivalence only. Experiments remain synthetic or controlled source/database fixtures rather than a measured autonomous improvement loop. Optional previewing and unknown-mean testing are documented in the appendices as separate extensions; the unknown-mean result requires $u = 0$ and changes the information constant. No theorem composes all extensions into a universal procedure.

Within this scope, we provide: optimistic route-restricted exploration, strict deployment verification, a verifiable negative answer, and explicit access/transfer interfaces. The strongest practical lesson is not that screening always wins. It is that search advice, a deployment certificate, and a sharing claim are different objects whose costs and failure allowances must remain visible to the planner.

AI USE STATEMENT

In this work, we used generative AI tools to draft sections of the paper and polish writing. Additionally, we used generative AI tools for editing code. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Online least squares estimation with self-normalized processes: An application to bandit problems. *arXiv preprint arXiv:1102.2670*, 2011. URL https://arxiv.org/abs/1102.2670.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016. URL https://arxiv.org/abs/1606.06565.

Robert D. Carr, Srinivas Doddi, Goran Konjevod, and Madhav V. Marathe. On the red-blue set cover problem. In *ACM–SIAM Symposium on Discrete Algorithms*, pp. 345–353, 2000. URL https://organicorigami.com/thrackle/papers/polyhedral_bib.html.

Shouyuan Chen, Tian Lin, Irwin King, Michael R. Lyu, and Wei Chen. Combinatorial pure exploration of multi-armed bandits. In *Advances in Neural Information Processing Systems*, volume 27, pp. 379–387, 2014. URL https://papers.nips.cc/paper_files/paper/2014/hash/d3ea0f3316d2da934d79b8b344eafee4-Abstract.html.

Herman Chernoff. Sequential design of experiments. *The Annals of Mathematical Statistics*, 30(3): 755–770, 1959. doi: 10.1214/aoms/1177706205.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1706.03741.

Rémy Degenne and Wouter M. Koolen. Pure exploration with multiple correct answers. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1902.03475.

Amol Deshpande, Lisa Hellerstein, and Devorah Kletenik. Approximation algorithms for stochastic Boolean function evaluation and stochastic submodular set cover. In *ACM–SIAM Symposium on Discrete Algorithms*, pp. 1453–1466, 2014. doi: 10.1137/1.9781611973402.107.

Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *ACM Symposium on Theory of Computing*, pp. 624–633, 2014. doi: 10.1145/2591796.2591884. URL https://arxiv.org/abs/1305.1979.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. Generalization in adaptive data analysis and holdout reuse. *arXiv preprint arXiv:1506.02629*, 2015. URL https://arxiv.org/abs/1506.02629.

Tom Everitt, Victoria Krakovna, Laurent Orseau, Marcus Hutter, and Shane Legg. Reinforcement learning with a corrupted reward channel. *arXiv preprint arXiv:1705.08417*, 2017. URL https://arxiv.org/abs/1705.08417.

Tanner Fiez, Lalit Jain, Kevin Jamieson, and Lillian Ratliff. Sequential experimental design for transductive linear bandits. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1906.08399.

Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart Russell, and Anca Dragan. Inverse reward design. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1711.02827.

Refael Hassin, Jérôme Monnot, and Danny Segev. Approximation algorithms and hardness results for labeled connectivity problems. *Journal of Combinatorial Optimization*, 14(4):437–453, 2007. doi: 10.1007/s10878-007-9044-x.

Zohar S. Karnin. Verification based solution for structured MAB problems. In *Advances in Neural Information Processing Systems*, volume 29, pp. 145–153, 2016. URL https://papers.nips.cc/paper_files/paper/2016/hash/65b9eea6e1cc6bb9f0cd2a47751a186f-Abstract.html.

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. URL https://www.jmlr.org/papers/v17/kaufman16a.html.

Abbas Kazerouni, Mohammad Ghavamzadeh, Yasin Abbasi-Yadkori, and Benjamin Van Roy. Conservative contextual linear bandits. *arXiv preprint arXiv:1611.06426*, 2016. URL https://arxiv.org/abs/1611.06426.

Robert W. Keener. Second order efficiency in the sequential design of experiments. *The Annals of Statistics*, 12(2):510–532, 1984. doi: 10.1214/aos/1176346503.

Neeraj Kumar, Daniel Lokshtanov, Saket Saurabh, and Subhash Suri. A constant factor approximation for navigating through connected obstacles in the plane. In *ACM–SIAM Symposium on Discrete Algorithms*, 2021. URL https://arxiv.org/abs/2011.14463.

Steven P. Lalley and Gary Lorden. A control problem arising in the sequential design of experiments. *The Annals of Probability*, 14(1):136–172, 1986. doi: 10.1214/aop/1176992620.

David Lindner, Sebastian Tschiatschek, Katja Hofmann, and Andreas Krause. Interactively learning preference constraints in linear bandits. In *International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 13505–13527, 2022. URL https://proceedings.mlr.press/v162/lindner22a.html.

Ximeng Liu, Qianlong Wang, Yingming Mao, Annan Li, Yatao Li, Shizhen Zhao, Jianmin Wu, Dawei Yin, and Dou Shen. Janus: An algorithm–evaluator co-evolution framework for LLM-driven discovery under expensive evaluation budgets. *arXiv preprint arXiv:2608.08189*, 2026. URL https://arxiv.org/abs/2608.08189.

Gary Lorden. On excess over the boundary. *The Annals of Mathematical Statistics*, 41(2):520–527, 1970. doi: 10.1214/aoms/1177697092.

George C. Necula. Proof-carrying code. In *ACM Symposium on Principles of Programming Languages*, pp. 106–119, 1997. doi: 10.1145/263699.263712.

Jerzy Neyman. Outline of a theory of statistical estimation based on the classical theory of probability. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 236(767):333–380, 1937. doi: 10.1098/rsta.1937.0005.

Sirin Nitinawarat and Venugopal V. Veeravalli. Controlled sensing for sequential multihypothesis testing with controlled Markovian observations and non-uniform control cost. *Sequential Analysis*, 34(1):1–24, 2015. doi: 10.1080/07474946.2014.961864. URL https://arxiv.org/abs/1310.1844.

Sirin Nitinawarat, George K. Atia, and Venugopal V. Veeravalli. Controlled sensing for multihypothesis testing. *IEEE Transactions on Automatic Control*, 58(10):2451–2464, 2013. doi: 10.1109/TAC.2013.2261188. URL https://arxiv.org/abs/1205.0858.

Juan C. Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. *arXiv preprint arXiv:2002.06673*, 2020. URL https://arxiv.org/abs/2002.06673.

Riccardo Poiani, Martino Bernasconi, and Andrea Celli. Non-asymptotic analysis of (sticky) track-and-stop. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/6c3490b3616832331d4e2224bb6414c2-Abstract-Conference.html.

Gregg Rothermel and Mary Jean Harrold. A safe, efficient regression test selection technique. *ACM Transactions on Software Engineering and Methodology*, 6(2):173–210, 1997. doi: 10.1145/248233.248262.

Daniel Russo and James Zou. How much does your data exploration overfit? controlling bias via information usage. *arXiv preprint arXiv:1511.05219*, 2015. URL https://arxiv.org/abs/1511.05219.

Jürgen Schmidhuber. Gödel machines: Self-referential universal problem solvers making provably optimal self-improvements. *arXiv preprint arXiv:cs/0309048*, 2003. URL https://arxiv.org/abs/cs/0309048.

Robert J. Serfling. Probability inequalities for the sum in sampling without replacement. *The Annals of Statistics*, 2(1):39–48, 1974. doi: 10.1214/aos/1176342611.

Marta Soare, Alessandro Lazaric, and Rémi Munos. Best-arm identification in linear bandits. In *Advances in Neural Information Processing Systems*, volume 27, 2014. URL https://arxiv.org/abs/1409.6110.

Yanan Sui, Alkis Gotovos, Joel Burdick, and Andreas Krause. Safe exploration for optimization with gaussian processes. In *International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 997–1005, 2015. URL https://proceedings.mlr.press/v37/sui15.html.

Yanan Sui, Vincent Zhuang, Joel W. Burdick, and Yisong Yue. Stagewise safe Bayesian optimization with gaussian processes. *arXiv preprint arXiv:1806.07555*, 2018. URL https://arxiv.org/abs/1806.07555.

Philip S. Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High confidence policy improvement. In *International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2380–2388, 2015. URL https://proceedings.mlr.press/v37/thomas15.html.

Matteo Turchetta, Felix Berkenkamp, and Andreas Krause. Safe exploration in finite Markov decision processes with gaussian processes. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://arxiv.org/abs/1606.04753.

Tonguç Ünlüyurt. Sequential testing of complex systems: A review. *Discrete Applied Mathematics*, 142(1–3):189–205, 2004. doi: 10.1016/j.dam.2002.08.001.

Abraham Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16 (2):117–186, 1945. doi: 10.1214/aoms/1177731118.

Abraham Wald and Jacob Wolfowitz. Optimum character of the sequential probability ratio test. *The Annals of Mathematical Statistics*, 19(3):326–339, 1948. doi: 10.1214/aoms/1177730197.

Eric Zelikman, Eliana Lorch, Lester Mackey, and Adam Tauman Kalai. Self-taught optimizer (STOP): Recursively self-improving code generation. In *Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2310.02304.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin Godel machine: Open-ended evolution of self-improving agents. *arXiv preprint arXiv:2505.22954*, 2025. URL https://arxiv.org/abs/2505.22954.

Jenny Zhang, Bingchen Zhao, Wannan Yang, Jakob Foerster, Jeff Clune, Minqi Jiang, Sam Devlin, and Tatiana Shavrina. Hyperagents. *arXiv preprint arXiv:2603.19461*, 2026. URL https://arxiv.org/abs/2603.19461.

A NOTATION, ASSUMPTIONS, AND QUERY ACCOUNTING

Symbol	Meaning
$G = (V, E), s, T$	Known directed acyclic upgrade graph, start, allowed terminal nodes.
$p, j(e), b_e$	Number of evidence labels, edge label, positive edge amplitude.
$\theta_j, \gamma_j, \sigma$	Unknown signed effect, known positive magnitude in the two-point model, known Gaussian noise scale.
$g_e = b_e\gamma_{j(e)}, u$	Edge gain under its positive sign and required terminal gain.
$Q(v), U(P)$	Diagnostics currently available at node v; distinct labels on path P.
$c_j, C_\theta(u), C_\theta^{\mathrm{acc}}(u)$	Inverse information, optimal positive-route cost, optimal positive admissible-route cost.
N, ℓ	Total charged sign queries; $\log(1/\delta)$.
$Z_{j,n}, A_h, B$	Sign log likelihood, epoch-h deployment boundary, cheap screening/rejection boundary.
$M, \lambda_{\mathrm{pr}}, I, \mu_{\mathrm{pr}}$	Preview count, price, information per preview, preview observation mean magnitude.
e_{pr}	Probability of choosing the expensive branch in the symmetric preview model.
F, m, v, r	Family size, number inspected, fixed violation count, number of unchecked releases.
α, β	Confidence-inversion allowance (in that subsection); packet release-risk allowance.
D	Number of input values in a finite code-evaluation domain.

The symbols ℓ, N, and λ_{pr} are not reused for log-likelihood statistics, domain sizes, or magnitude ratios. All asymptotics fix the graph, gaps, and noise unless explicitly stated otherwise. A family-sharing label identifies equal *signs* in the graph model; equality of full output behavior in the code checker is a stronger sufficient condition.

Protocol and information. An algorithm observes only its own randomization, declared graph/access information, and charged query outcomes. Conditional on its past and its chosen query j, the next query has distribution $\mathcal{N}(\theta_j, \sigma^2)$. It may maintain arbitrary internal state. The sign model has no uncharged feedback from deployment, including a revealed true utility; actual gain is evaluated by the reporting harness. Edges are irreversible unless a rollback edge is explicitly included. Under global access, a complete route can be tested before deployment. Under local access, tests and deployments must be interleaved. Earlier raw data or certified signs may be retained, but earlier *access* is not itself an observation.

A uniform safety statement controls a harmful *edge*, not just final aggregate return. A negative edge followed by a larger positive edge remains a failure. Completion for target u requires a positive route with gain at least u; on an unsafe run, the code's terminal flag need not imply this mathematical event. The global selective algorithm of Theorem 1 stops with finite expected cost on feasible and infeasible instances. Budget and epoch guards are implementation interruptions, never infeasibility certificates. The separate unknown-mean extension below retains its stated feasible-instance scope.

Why exact sign reuse is an assumption. If $R(x) = \theta^\top x$ and each update adds $b_e e_{j(e)}$, the sign of its increment depends only on $j(e)$. This algebra makes shared-label testing sound by construction. It does not establish an unknown family map for arbitrary source edits. Section 4.1 treats that map separately; only after verification can its members be placed under an exact shared label.

B GLOBAL MIXED-SIGN COMPLEXITY

B.1 INFORMATION LOWER BOUND

Lemma 3 (A deployed positive label requires information)**.** *Fix θ and a label j with $\theta_j = \gamma_j > 0$. For every uniformly δ-safe policy, let q_j be the probability under θ of ever deploying a j-edge, and let N_j be the number of its trusted queries. Then*

$$\mathbb{E}_\theta N_j \geq c_j\{q_j\ell - \log 2\}.$$

Proof. Let $\theta^{(j)}$ differ only in coordinate j, now equal to $-\gamma_j$. The conditional Kullback–Leibler (KL) divergence contribution of a query of j is $\mathrm{KL}(\mathcal{N}(\gamma_j, \sigma^2), \mathcal{N}(-\gamma_j, \sigma^2)) = 2\gamma_j^2/\sigma^2 = 1/c_j$. Other

queries, action choices conditional on history, and internal randomization contribute no additional conditional KL. Apply the chain rule to the finite transcript through time t and data processing to the event that a j-edge has been deployed by then. The event has probability at most δ in the alternative, by safety. For any event probabilities q and $a \le \delta$, binary relative entropy satisfies

$$\mathrm{kl}(q, a) = q \log(1/a) + (1-q) \log(1/(1-a)) - h(q) \ge q\ell - \log 2.$$

This inequality does not require $q \ge \delta$. Let $t \to \infty$; monotone convergence gives the expected query count and event probability. Localization avoids assuming termination in the alternative, which may have no positive route. This is the standard stopped-transcript method of Kaufmann et al. (2016). □

Lower bound in Theorem 1. Let $\mathcal{E}$ be safe completion with terminal gain at least u. Safety and the stipulated completion imply $\mathbb{P}_\theta(\mathcal{E}) \ge 1 - 2\delta$ (if completion already includes positivity, the weaker bound still holds). On $\mathcal{E}$ the deployed route's distinct-label cost is at least $C_\theta(u)$ and every deployed label is positive. Therefore

$$\sum_{j:\theta_j>0} c_j q_j = \mathbb{E}_\theta \sum_{j:\theta_j>0} c_j \mathbf{1}\{j \text{ deployed}\} \ge (1-2\delta) C_\theta(u).$$

Sum Lemma 3 over positive labels; querying negative labels only increases the total. This proves the finite lower bound. □

B.2 Testing at different confidence levels

For known magnitude γ_j, the log-likelihood statistic after n fresh queries is

$$Z_{j,n} = \frac{2\gamma_j}{\sigma^2} \sum_{i=1}^{n} Y_{j,i}.$$

Under its positive sign the increments have mean $\mu_j = 1/c_j$ and variance $2\mu_j$; under its negative sign the distribution is reflected. A fresh two-boundary test stops at $Z_{j,n} \ge A$ or $Z_{j,n} \le -B$. Likelihood-ratio martingales imply

$$\mathbb{P}_-(\text{positive verdict}) \le e^{-A}, \qquad \mathbb{P}_+(\text{negative verdict}) \le e^{-B}. \tag{6}$$

These bounds continue to hold conditional on adaptively deciding to start a fresh test. The positive expected length is at most $c_j(A + K_{\mu_j})$ and the negative expected length at most $c_j(B + K_{\mu_j})$, where

$$K_\mu = \mu + 2\sqrt{\mu/\pi}.$$

The Gaussian overshoot proof is given in Appendix H. These bounds suffice without an asymptotic approximation to the expected length.

B.3 Selective screening and a verified alternative to endless restarts

A screened sign is search advice, not a deployment certificate. All search calls in this subsection charge each positive-screened label its original cost c_j; zeroing its cost would change the selected route and invalidate the optimality argument. Only the symmetric-baseline ablation zeroes already *certified* labels. Let $\ell_0 = \max\{2, \ell\}$ and $B = \log(4p\ell_0)$. In epoch h, set

$$A_h = \log \frac{p\pi^2 h^2}{6\delta}, \qquad \alpha = pe^{-B} = \frac{1}{4\ell_0}, \qquad q = 2\alpha \le \frac{1}{4}.$$

Start an epoch with no screened signs. Find a minimum original-cost eligible route avoiding screened negatives. Screen its unqueried labels at $\pm B$, in a fixed order; a negative result triggers replanning, not deployment. If every label on a route has screened positive, test its labels on fresh streams at $(+A_h, -B)$. A rejection starts the next epoch, and all positive verdicts authorize this route. If no eligible route remains, the screened-negative set hits every eligible route. Delete redundant members to obtain a blocking set K_h, and test each on fresh streams at $(+B, -A_h)$. All negative verdicts authorize an infeasibility output; a positive verdict starts the next epoch. The deletion order in the implementation is decreasing cost, then label index. This gives an inclusion-minimal blocking set, not a minimum-weight one.

Correct output and finite expected stopping in Theorem 1. In each epoch the algorithm performs at most p strong verification tests: either for one route or for one blocking set. The relevant wrong verdict has probability at most e^{-A_h}, conditional on any preceding screening and planning, by the fresh-stream bound (6). A false deployment requires a false positive sign; a false infeasibility output on a feasible instance requires a false negative sign in the blocking set. The sum of the allowances over both possible answer types is

$$\sum_{h\geq 1} pe^{-A_h} = \delta.$$

There is no additional factor of two: the epoch verifies *either* a route *or* a blocking set. If all strong verdicts are correct, every deployment is positive and every infeasibility output is true. Query caps return "undecided", never "infeasible".

For each label, pre-sample an independent screening-test stream even if it will not be used. With probability at least $1-\alpha$, every screening verdict is correct. Lazy search then either finds a positive eligible route or, on an infeasible instance, constructs a truly negative blocking set. At most p distinct labels are screened, so search ends. Each proposed strong verdict has correct sign and a probability at most e^{-B} of being rejected at its cheap boundary. Thus a fresh epoch gives the correct terminating answer with probability at least $1-q$, regardless of feasibility. Conditional on reaching epoch h, its work has finite expectation bounded by

$$C_\Sigma\{B + \max(A_h, B)\} + 2K_\Sigma, \quad C_\Sigma = \sum_j c_j, \quad K_\Sigma = \sum_j c_j K_{1/c_j}.$$

The reaching probability is at most q^{h-1}. Since $A_h = O_\theta(\ell + \log h)$, summing the bound proves finite expected stopping on *every* sign assignment. On a feasible instance the correct answer is a positive eligible route; on an infeasible instance it is a blocking certificate. Together with the error allowance, this proves the claimed joint output guarantee. □

Leading cost and route-local screening term. First fix a deterministic label order and tie rule. Run the optimistic search with noiseless correct screening signs, and denote its queried set by S_θ. The set is defined only for analysis; the policy never observes it. Put $C_{S_\theta} = \sum_{j\in S_\theta} c_j$ and $K_{S_\theta} = \sum_{j\in S_\theta} c_j K_{1/c_j}$. Deciding whether to query label j occurs before its screening stream is inspected. Therefore its expected screening work factors into its query probability and the expected fresh-test length. For $j \notin S_\theta$, querying j requires an earlier wrong screen, of probability at most α. Consequently one epoch's expected screening cost is at most

$$C_{S_\theta} B + K_{S_\theta} + \alpha(C_\Sigma B + K_\Sigma). \tag{7}$$

The independence used here concerns the unused stream of j, not the event that *all* screens are correct. Conditioning a test length on its own correct verdict would not justify the same calculation.

On the all-screens-correct event of a feasible instance, every truly positive eligible route remains an optimistic candidate. The selected route is positive and minimizes cost among a superset containing all positive eligible routes. Its cost is therefore both at most and at least $C_\theta(u)$. This is why positively screened labels must retain their original costs in the planner. Strong verification streams are fresh, giving expected route-verification work at most $C_\theta(u)A_h + K_\Sigma$ on that event. On its complement, an arbitrary route or blocking set costs at most $C_\Sigma \max(A_h, B) + K_\Sigma$. For fixed instance and sufficiently small δ, $A_h \geq B$ for all h; hence one epoch costs at most

$$C_\theta(u)A_h + C_{S_\theta}B + O_\theta(1) + \alpha C_\Sigma(A_h + B).$$

Multiply by q^{h-1} and sum. Since $q = O(1/\ell)$, αA_h contributes $O_\theta(1)$ after summation, and

$$\sum_{h\geq 1} q^{h-1}A_h = \ell + O_\theta(1), \qquad \sum_{h\geq 1} q^{h-1}B = B + o(1).$$

It follows that

$$\mathbb{E}_\theta N \leq C_\theta(u)\ell + C_{S_\theta}\log(4p\ell_0) + O_\theta(1). \tag{8}$$

A blocking test on a feasible instance is triggered only after a wrong screen. Its expected contribution is bounded by $\sum_h q^{h-1}\alpha(C_\Sigma A_h + K_\Sigma) = O_\theta(1)$. The lower bound in Lemma 3 completes the leading-constant result. For independent random test ordering, condition on the ordering seeds first and replace C_{S_θ} by its expectation; the finite set of orderings has uniform fixed-instance constants. Irrelevant labels still enter the confidence allocation and the exceptional-event constant, but no longer the main screening term. The bounds fix p and gaps; they are not uniform as distractor count or inverse gaps grow with $1/\delta$. □

What is and is not optimal on infeasible instances. The above policy has expected cost $O_\theta(\ell + \log \ell)$ there. It is not claimed to attain the optimal infeasibility constant. A label blocking set is sufficient, but an optimal test may combine information about several negative labels without certifying a minimum integral cut. The upper statement therefore asserts reliable, finite abstention, not a second universal efficiency characterization.

Optional screening at moderate confidence. The implementation uses the deterministic rule $B \geq A_1/2$ to select a symmetric absorbing ledger at threshold $\log(p/\delta)$, before looking at data. Its two-sided familywise sign error is at most δ. It halts on every fixed nonzero instance and outputs a positive route or a certified blocking set, using the same exact utility-qualified planner. Under the same streams and tie/order seeds, this branch equals the symmetric baseline pathwise. At each fixed instance, $B/A_1 \to 0$, so eventually the rule always selects the lazy algorithm and preserves (8). The factor $1/2$ was fixed before experiments; it is not a Bellman-optimal choice. Outside the skip region there is no theorem of universal additive-$O(1)$ dominance over symmetric testing or cheap rejection. Even redundant fresh screening on an already determined route can add $\Theta(\log \ell)$ work. The empirical comparisons retain such finite-confidence losses.

Positioning against explore–verify. Karnin (2016), Algorithm 1, Theorem 3 and Corollary 4, already separate moderate-confidence exploration from high-confidence verification, including an exploration $\log\log(1/\delta)$ term. Degenne & Koolen (2019) gives asymptotic optimality for multiple acceptable answers. Our construction instantiates these ideas in a utility-qualified label-path answer set, with a route-restricted exploration oracle and a checked negative alternative that makes infeasible inputs operationally well-defined. Adding an infeasibility answer also gives a structured identification problem; we do not claim that existing frameworks cannot express it. The new graph-specific accounting is (7), the blocking-certificate stopping rule, and their composition under the contracts below.

C Checkpoint access and transfer composition

C.1 A sufficient local-access condition

A graph is a *checkpoint chain* if it is the concatenation of K finite DAG blocks with entries s_k and common exits s_{k+1}. Different blocks have disjoint label sets J_k. At s_k, every label of block k can be checked repeatedly before any block deployment. All intermediate nodes may have empty access. All block routes rejoin the same exit. We consider either prescribed block gain quotas u_k, or the original global-gain problem when every route in block k has the same positive-sign gain G_k and $0 \leq u \leq \sum_k G_k$. In the latter case every positive complete concatenation qualifies, so it is a literal special case of the original answer set, not a quota relaxation.

Checkpoint part of Theorem 2. Run the global policy at each entry with allowance δ/K, and deploy the selected block route only after its full certificate. Every query is legal: no old block's access or future block's access is needed. Sum the strong-test error allowances over the blocks. A local infeasibility certificate is also a global obstruction under the stated answer contract, because every complete route must pass through that block and meet its local constraint. A budget stop has no such interpretation. All already certified prefix deployments remain retained.

On every feasible fixed sign assignment, each block has a qualifying positive route. Disjointness of label sets gives the factorization

$$C_\theta^{\mathrm{acc}} = \sum_{k=1}^{K} C_{\theta,k}(u_k).$$

For the equal-gain special case, the same expression equals $C_\theta^{\mathrm{acc}}(u)$ in the original model. Apply (8) separately and use $\log(K/\delta) = \ell + \log K$. Multiplying a block's work by the probability of reaching it only reduces the expected upper bound, because future query streams are fresh. The sum is $C_\theta^{\mathrm{acc}}\ell + O_\theta(\log \ell)$. The general access lower bound (9) applies to this instance and has the same leading constant. Finite expected stopping follows from at most K finite-expectation local calls. Joint correct output and non-regression hold with probability at least $1 - \delta$. □

Why the sufficient condition is stronger than local observability. Querying the immediate branches before a decision is not by itself enough to minimize global query cost. One branch may reuse a label in a later block while another may force an additional distinct label, or future attainable gain may depend on the present choice. Disjoint block labels, a common next checkpoint, and factored eligibility remove those couplings. Without them, locally optimal route choice need not attain $C_\theta^{\mathrm{acc}}(u)$ even when it is safe. Proposition 4 remains an obstruction without informative pre-commitment access. The checkpoint theorem does not cover independently sampled arbitrary access sets.

C.2 A PACKET RULE THAT SURVIVES OPTIMIZATION

Fix the family, violation set, and package parameters (F, m, r, β_t) before its random member checks. Conditional on the entire pre-package history, sample m members uniformly without replacement; if all match a sufficient transfer relation, issue the next r uniformly sampled unchecked members as a *packet*. Previously inspected and verified members can also be used. Choose β_t predictably, with $\sum_t \beta_t \le \beta$, and require $\max_v R(F, m, r; v) \le \beta_t$ before sampling. After seeing the packet, the planner may choose *any subset of that packet*. It may not enlarge it, replace its members by preferred unchecked identities, or repeatedly redraw under an uncharged allowance.

Transfer-budget part of Theorem 2. For any fixed violation pattern, the permutation argument in Theorem 9 bounds the conditional-on-prehistory probability that the issued packet contains an invalid member by β_t. It is not a conditional-on-clean-inspection bound. A subsequently selected subset has an invalid member only if the original packet has one. Taking conditional expectations and summing gives an invalid-packet probability at most β even with adaptive family choice and future planning. The claim requires the finite family and its pattern to be fixed before that family's checks, conditional on the history; changing members after the draw violates it.

Use sign certificates with total allowance δ, either globally or within the checkpoint chain, and restrict statistical reuse to materialized packet members or members proved valid by a sound checker. Assume a valid relation is sufficient for the claimed sign transfer and that the diagnostic noise law continues to hold conditional on validation history. No independence between transfer and sign errors is required. Any harmful deployment lies in the union of a wrong sign certificate and an invalid released member, of total probability at most $\delta + \beta$. Stopping and positive-route efficiency for exact sharing do *not* automatically transfer: validation may remove a route, and its checks have an additional cost. The theorem composes safety, not the exact C_θ constant for random family validation. □

Planner interface and cost objective. A sound symbolic proof creates deterministic member-to-label links; only these can enter the exact-label planner without a new risk allowance. A packet instead creates a finite list of randomized authorized links, after which route selection is free to optimize among them. For named edits not in the packet, direct validation remains available. Before drawing a package, a planner can compare inspection cost plus estimated downstream sign-query cost against individual member checks. Our integrated example minimizes nominal all-valid inspection cost over $m + r = q$ for exchangeable slots and the explicit constraint $\max_v R \le \beta$. This is a declared planning surrogate, not an unknown-population expected-cost optimum. It offers individual checking and often selects it. Identity-specific utility or prerequisites must be evaluated on actual issued members; a random release slot cannot silently substitute for a required named edit.

Joint and conditional probabilities. For $F = 64, m = 32, r = 3, v = 1$, $R = 3/64 \approx 4.69\%$, while conditional invalid-packet probability given clean checks is $3/32 \approx 9.38\%$. Selecting the package before its random checks permits the former guarantee; calling the clean result a 5% conditional certificate is incorrect. If conditional certification is required, use the hypergeometric confidence-inversion rule in Proposition 10, with its separate confidence and release allowances. Invalidity denotes failure of a sufficient relation. For a lower bound on *harm*, the indistinguishable single-member alternatives in (10) must be allowed to have negative utility; otherwise that coupling bounds invalid-release control only.

D STATE-DEPENDENT DIAGNOSTIC ACCESS

For any fixed instance on which safe eligible completion has probability at least $1 - 2\delta$, the stopped-information argument below gives

$$\mathbb{E}_\theta N \geq [1 - (p+2)\delta] C_\theta^{\mathrm{acc}}(u)\ell - (\log 2) \sum_{j:\theta_j>0} c_j. \tag{9}$$

This lower bound does not assert that a uniformly complete policy exists for every access pattern.

Proposition 4 (Indistinguishable irreversible commitments)**.** *There is a two-branch graph with neutral irreversible commitments, indistinguishable pre-commitment observations, and a positive admissible continuation in each of two worlds, such that every policy has safe-completion probability at most $1/2$ in one world. A uniformly δ-safe policy has terminal-arrival probability at most $1/2 + \delta$ there.*

Definition 5 (Temporal admissibility)**.** Write a terminal path as $(v_0, e_1, v_1, \ldots, e_H, v_H)$, and let h_j be the index of the first j-edge. It is diagnostic-admissible if, for each used label j, some $k < h_j$ has $j \in Q(v_k)$. At a replanning state, already certified positive labels may be added to this initial condition, but previously visited access sets may not.

Lemma 6 (Unqueried deployment is rare)**.** *For any positive label j under θ, the probability of first deploying a j-edge before any j-query is at most δ for a uniformly δ-safe policy.*

Proof. Stop the transcript immediately before the first j-query or at the first j-deployment, whichever occurs first. Under θ and its j-sign flip, every observation up to the relevant event has the same conditional distribution. The unqueried-deployment event consequently has the same probability in both worlds. In the flipped world it is harmful and hence has probability at most δ. Finite-time localization handles paths that never reach either event. □

Access lower bound (9). With probability at least $1 - 2\delta$ the policy completes safely on the fixed feasible θ. Exclude also the union of unqueried first deployments of positive labels, losing at most $p\delta$. On the remaining event, the positive deployed route is eligible and every first-used label had an earlier query. Such a query was legal only at a visited prefix mode, so the route is diagnostic-admissible. Its union cost is at least $C_\theta^{\mathrm{acc}}(u)$. Summing Lemma 3 gives (9). If there is no admissible positive route, this joint event is empty; the two guarantees cannot coexist for $\delta < 1/(p+2)$ at that instance. This is a necessary condition, not a sufficient uniform-completion characterization. □

Proposition 7 (All-positive access constant)**.** *On a fixed finite all-positive instance with an eligible admissible route, the optimal leading expected query cost among uniformly safe policies that complete on that instance is $C_{\theta^+}^{\mathrm{acc}}(u)$. A single minimizing-route policy, ignorant of the true signs, achieves this on the all-positive instance. If no eligible admissible route exists, safety and all-positive completion are incompatible for sufficiently small δ.*

Proof. The lower bound follows above. For the upper bound, compute a minimum-cost eligible admissible route using the known magnitudes. Schedule each distinct required label's symmetric sign test at its earliest available visited mode, using threshold $\log(p/\delta)$. Test before leaving that mode and deploy an edge only after its label is positively certified. Stop on any negative verdict. At most p tests are used, each with error at most δ/p; the policy is uniformly safe and completes on the all-positive instance with probability at least $1 - \delta$. Its expected query count is bounded by the sum of the positive expected lengths on the chosen route, which equals $C_{\theta^+}^{\mathrm{acc}}(u)\ell + O(1)$ by the overshoot bound. No query is made retrospectively. The resulting upper statement is explicitly all-positive; it is not mixed-sign instance optimality. □

Proposition 4. Allow two objective-neutral initial edges to configurations A and B. No sign diagnostic is available before this choice. Only the chosen branch's sign diagnostic is then available, and its single deployment edge leads to a terminal. World 1 has signs $(+, -)$ and world 2 has signs $(-, +)$. Both worlds have a positive diagnostic-admissible route. Let q be the common probability of selecting branch A; pre-commitment transcripts have the same law, so it is common to both worlds. Safe completion in world 1 has probability at most q, and in world 2 at most $1 - q$. Their minimum is at most $1/2$. Unsafe terminal deployment can add at most δ under uniform safety, giving the other assertion. Abstaining before commitment cannot improve these bounds. □

Arbitrary-access replanning baseline. The following symmetric-ledger policy is a separate arbitrary-access baseline. It is not the checkpoint selective procedure and does not inherit its mixed-sign efficiency theorem. The implemented comparison uses a shared symmetric absorbing ledger at threshold $\log(p/\delta)$, not the global fresh-screen algorithm. On encountering a negative certificate, it excludes that label and plans again from the *current* node, with all positive certificates retained. For a proposed continuation it queries currently available labels that the continuation will need, then deploys its next positively certified edge. At future nodes it repeats this schedule. A planning failure, deterministic search cap, or query-budget exhaustion retains the already deployed prefix. The paired stopping baseline uses the same planner and schedule but ends at the first rejection. Therefore the measured difference is replanning, not an unequal safety threshold or free access to old modes. Any finite run remains safe on the one simultaneous sign-correctness event. Permanent negative certificates can incorrectly exclude a useful route; no general completeness theorem is claimed for this access heuristic.

Migration fixture and the scope of known access. The SQLite fixture constructs four records with a pre-migration stratum field, registers a diagnostic that groups by that field, migrates to an aggregate table, and drops the original records. A bounded actual query probes the available interface; syntax-only query-plan checks can be cached and are insufficient for checking whether diagnostic rows still exist. Recreating the pre-migration schema after deletion produces zero rows, not the four original records. The fixture's transition contract prohibits snapshots. Declared Structured Query Language (SQL) and data dependencies plus the transition give the exact access map for this fixture, and tests verify the observed map. The graph experiments randomize access independently; they are not calibrated to the frequency of such migrations in deployed systems. A retained service, copy, or other recovery action changes Q or the graph. Statistical validity after training on a benchmark requires a separate model of adaptive data use (Dwork et al., 2015; Russo & Zou, 2015); mere database capability does not certify that validity.

E UNKNOWN NONZERO GAUSSIAN MEANS

Here the hypothesis class is $\theta \in (\mathbb{R} \setminus \{0\})^p$, with known σ and unknown magnitudes. We require any positive terminal route ($u = 0$), because a nonzero exact gain target introduces a separate mean-estimation boundary. Define

$$d_j = \frac{2\sigma^2}{\theta_j^2}, \qquad D_\theta = \min_{P \text{ positive terminal route}} \sum_{j \in U(P)} d_j.$$

Proposition 8 (Changing the hypothesis class changes the constant)**.** *For fixed feasible θ and global access, the optimal leading expected query count is $D_\theta \ell$. In particular, $d_j = 4c_j$ when comparing a true magnitude $|\theta_j|$ with the known two-point model. A pilot-plus-composite-test policy attains this leading constant while remaining uniformly safe over all nonzero means.*

Proof. For a positive coordinate, replace its mean by $-\zeta$, where $\zeta > 0$ can be arbitrarily small. The conditional KL is $(\theta_j + \zeta)^2/(2\sigma^2)$. Repeating Lemma 3 and letting $\zeta \downarrow 0$ gives

$$\mathbb{E}_\theta N_j \geq d_j\{q_j \ell - \log 2\}.$$

Summing along safe positive completions gives the leading lower bound D_θ. Replacing the known magnitude by a fitted value in a two-point SPRT would not justify this lower-bound class or a uniform false-positive guarantee.

For the upper bound, use fresh epochs. Each starts with $n_0 = \lceil \ell_0^{2/3} \rceil$ observations of every label. Let $\widehat{\theta}_j$ be their averages, screen out negative averages, and plan with weights

$$\widehat{d}_j = \frac{2\sigma^2}{\max\{|\widehat{\theta}_j|, \ell_0^{-1/8}\}^2}.$$

If there is no route, restart. Otherwise test its labels on fresh streams using the anytime boundary

$$w_n(\alpha_h) = \sqrt{\frac{2\sigma^2}{n} \log \frac{2n(n+1)}{\alpha_h}}, \qquad \alpha_h = \frac{6\delta}{p\pi^2 h^2}.$$

Accept when $\overline{Y}_n > w_n(\alpha_h)$ and reject when $\overline{Y}_n < -w_n(\alpha_h)$. A rejection restarts the epoch; all positive gates authorize the route. Gaussian tails and $\sum_{n\geq 1} 1/[n(n+1)] = 1$ give a two-sided simultaneous error probability at most α_h for each test. Summing over epochs and labels proves uniform safety.

For any fixed nonzero θ, all pilot signs are correct with probability $1 - O_\theta(e^{-a_\theta \ell^{2/3}})$. On the event $\max_j |\widehat{\theta}_j - \theta_j| \leq \ell^{-1/6}$, which fails with probability at most $O_\theta(e^{-b_\theta \ell^{1/3}})$, all weights converge uniformly to d_j. Because the finite graph has finitely many terminal paths, the selected route's true union cost is at most $D_\theta + o(1)$. More explicitly, its cost is no more than the best route's cost plus twice the maximal pathwise sum of weight errors.

For a fixed positive mean, a composite test has expected length at most $(d_j + o(1)) \log(1/\alpha_h)$ as its error allowance vanishes. To verify this, fix $\varepsilon_1 > 0$. Beyond $n = (1+\varepsilon_1) d_j \log(1/\alpha_h)$, the $2 \log n$ term is lower order; the boundary is below the true mean by a fixed positive fraction for sufficiently small α_h. The event of not yet exiting implies $\overline{Y}_n \leq w_n$, whose Gaussian tail is then exponentially summable in n. The tail-sum expression for expected stopping gives the bound, and then $\varepsilon_1 \downarrow 0$ gives its coefficient. Reflection proves the same length statement for negative means. A coarse bound $O_\theta(1 + \log(1/\alpha_h))$ holds for every epoch.

For each fixed δ, there is a strictly positive probability of correct pilot signs and successful gates in a fresh epoch, so completion is almost sure on feasible instances. As $\delta \downarrow 0$, the epoch failure probability tends to zero faster than any inverse power of ℓ (the gate failure adds $O(\delta)$). Pilot work is $pn_0 = o(\ell)$, exceptional pilot events have negligible expected gating cost by the coarse uniform path bound, and repeated epochs contribute $o(\ell)$. Consequently the total expected count is at most $(D_\theta + o(1))\ell$. Together with the lower bound this proves the proposition. The result is pointwise: constants and convergence deteriorate as a true mean approaches zero. □

F Partial validation and structural transfer

F.1 Exact finite-population guarantees

Theorem 9 (Fixed-family packet risk and unchecked-release obstruction)**.** *For a fixed family with v invalid members, uniform inspection followed by a precommitted uniform packet has the exact joint risk in* (5)*. A valid prior bound on v may restrict its maximization. Conversely, suppose single-member invalid alternatives are observationally indistinguishable until that member is checked. Every policy uniformly controlling invalid releases at level β satisfies, in the all-valid world,*

$$\mathbb{E}_0 U_0 \leq F\beta, \tag{10}$$

where U_0 counts distinct members released unchecked. For a harm lower bound, assume those permitted single-member alternatives can have negative utility while the representative observations remain unchanged.

The family is fixed, and v of its F members violate the proposed transfer relation. A member check identifies a violation without error; a fallible check requires its own allocated error allowance. Uniform sampling is over distinct members, without replacement. Violations here mean disagreement with a sufficient transfer condition, so a violation need not actually have negative utility. Bounds on invalid releases therefore also bound harmful releases once the representative is safely certified, but can be conservative.

Exact risk in Theorem 9. Generate a uniform random permutation of all F members. Inspect the first m; if they are valid, release the next r. The event of an invalid issued packet under this rule is the difference between two nested events: no violation among the first m, and no violation among the first $m + r$. Their probabilities are the two hypergeometric zero-count probabilities in (5). The formula is exact for the *joint* event of clean inspection and invalid release. It is not a posterior probability conditional on observing a clean inspection. Maximizing over every fixed v yields the uniform bound, and the gate chooses the largest permitted r up to its requested release count. With a known bound $v \leq v_{\max}$, maximizing over only that range is valid. A guessed violation prior cannot be used as such a bound. □

Coupling lower bound in Theorem 9. For each member i let A_i be the event that it is released before it has received an independent check in the all-valid world. Couple that run and the world with only member i invalid, using the same internal randomness. On A_i, all inspection responses before its release are identical; hence the algorithm makes the same release in the alternative. The alternative's invalid-release probability is at least $\mathbb{P}_0(A_i)$, so uniform invalid-release control requires $\mathbb{P}_0(A_i) \le \beta$. Summing over the F members gives $\mathbb{E}_0 U_0 = \sum_i \mathbb{P}_0(A_i) \le F\beta$. The argument allows adaptive checking and stopping. Members are counted once, and verification after release cannot justify that release. If every unchecked member is released and at most m were inspected, $U_0 \ge F - m$, proving the near-census requirement. □

Sharpness and interpretation. For one violation, (5) equals r/F, independent of m. For fixed r, inspecting $F - r$ members and releasing all remaining members only after clean inspection attains worst-case risk r/F, since for $v \ge 1$ the chance all violations lie in the last r positions is $\binom{r}{v}/\binom{F}{v} \le r/F$. Thus the one-violation obstruction is sharp for this full-inspection limit. Cheap sampling does not become a certificate for unrestricted individual edits merely because many representatives agree. This complements the familiar distinction between estimating a population property and certifying every individual outcome; finite-population sampling itself is classical (Serfling, 1974).

Proposition 10 (A confidence-inversion release rule)**.** *Let S be the number of violations in m uniform inspections and put*

$$V^+(S) = \max\left\{ v' : \Pr_{F,v',m}(S' \le S) \ge \alpha \right\},$$

where feasible counts are $S \le v' \le F - m + S$ and S' is hypergeometric. Let $r(S)$ be any requested count, selected from the observed inspections, satisfying

$$1 - \frac{\binom{F-m-(V^+(S)-S)}{r(S)}}{\binom{F-m}{r(S)}} \le \beta.$$

Uniformly releasing $r(S)$ unchecked members has probability of an invalid release at most $\alpha + \beta$. A separately certified representative with error probability α_{rep} changes the bound to $\alpha_{\mathrm{rep}} + \alpha + \beta$.

Proof. Exact one-sided test inversion gives $\Pr_v(v > V^+(S)) \le \alpha$: on this event the true hypergeometric lower-tail cumulative distribution function (CDF) at the observed count is less than α, an event of probability at most α. This is the standard confidence-inversion construction (Neyman, 1937). Conditional on an inspection transcript with $v \le V^+(S)$, at most $V^+(S) - S$ of the $F - m$ remaining members are invalid. A uniform sample of size $r(S)$ misses them all with the stated combinatorial probability; monotonicity in the violation count makes the displayed expression an upper bound. Integrate over good inspection transcripts and add the coverage-failure probability. The representative event is combined by a union bound and need not be independent. □

The inversion rule can release after a nonzero observed violation count and provides a conditional sampling-risk control on a high-probability event. The clean-inspection gate controls only the joint release event and often permits more unchecked members at the same numerical allowance. We do not treat the two guarantees as identical conditional statements. Checked valid members may always be released under either rule. Across multiple families or adaptive release batches, these single-batch bounds must be allocated within a total failure budget.

F.2 Source-derived equivalence for a restricted expression grammar

The patch experiment uses an integer variable x. Constants and x are integer-valued; additions, subtractions, unary signs, and products are interpreted in the ring of unbounded integers. A finite list of pure integer-returning functions and other integer-valued subexpressions may be treated as opaque atoms. An atom is identified by its exact abstract syntax tree, not by matching outputs on a few test inputs.

The checker performs two checks before comparing normal forms. First, both complete expressions must lie in the recognized pure integer grammar. Floating-point arithmetic is not rewritten using ring identities. Second, every opaque candidate atom must already occur as an evaluated atom of the

representative, apart from x itself. The representative's totality on the declared domain is checked directly. Eagerly evaluated products such as $0 \cdot (1//0)$ cannot bypass this atom condition, even if their polynomial coefficients would cancel. Unsupported syntax, newly introduced atoms, normal-form resource caps, or unequal normal forms return `UNKNOWN`, never an inferred negative utility sign.

Proposition 11 (Soundness of the restricted checker)**.** *Fix a domain of integer inputs. Suppose the representative and every allowed opaque atom terminate with an integer on that domain. If the checker accepts a candidate, its output equals the representative's output on every input in the domain. Consequently, under the same baseline and per-input objective, the two patches have the same utility gain.*

Proof. Associate a formal indeterminate with each syntactically distinct opaque atom and with x. Recursively normalize additions and subtractions by combining coefficients, and products by multiplying monomials and collecting like terms. Structural induction proves that each normalized polynomial evaluates to its expression for every assignment of integers to its indeterminates: the induction uses only the ring identities for integer addition and multiplication. Equal normal forms therefore evaluate equally for every assignment, including the actual outputs of the opaque atoms at any domain input. The atom restriction and the totality premise guarantee that the candidate does not introduce a new undefined operation. Thus equality includes successful termination rather than merely a formal expression identity. Identical candidate outputs under the same baseline give identical per-input loss differences and identical aggregate gain. □

The checker is not a verified theorem prover; its implementation is tested against exhaustive finite-domain execution and adversarial exception examples. Its proved fragment is narrower than the entire set of equivalent programs: for example, two differently written square-root computations remain different opaque atoms unless directly checked. That incompleteness is intentional. Proof-carrying code provides the general architecture of a producer supplying checkable evidence (Necula, 1997); the present normalizer is a restricted executable instance of that idea.

G Planning formulations, approximations, and hardness

G.1 Exact state expansion

For global access, a state (v, S) records a node and the already used label set. Traversing an edge with label j adds cost c_j if $j \notin S$ and zero otherwise. Path projection preserves this cost in both directions, proving equivalence to the minimum distinct-label path problem. With a gain target, keep the maximum prefix gain for each (v, S), as justified in Appendix B. Availability requires also a set A of diagnostics seen along a *planned* prefix, yielding (v, S, A) and the condition $j(e) \in A$ for its first use. This is an offline existence check: the actual execution must schedule queries while those modes are still current. It does not grant retrospective queries. Bounds $|V|2^p$ and $|V|4^p$ are loose worst-case counts.

G.2 A compact mixed-integer formulation

Add a zero-cost super-sink t_* and an unlabeled artificial arc from each allowed terminal to it. Let $x_e \in \{0, 1\}$ be arc-use variables and $y_j \in \{0, 1\}$ label activation variables. The global formulation is

$$\min \sum_j c_j y_j, \qquad \sum_{e \in \mathrm{out}(v)} x_e - \sum_{e \in \mathrm{in}(v)} x_e = \mathbf{1}\{v = s\} - \mathbf{1}\{v = t_*\}, \qquad x_e \le y_{j(e)}$$

for original labeled arcs, together with $\sum_{e \in E} g_e x_e \ge u$ when a gain target is requested. Set $x_e = 0$ for forbidden negative labels and zero out already certified label costs. In a DAG, an integral unit flow is one path: it cannot contain a separate circulation, and induction from the source excludes branching flow larger than one. Thus projection gives a feasible original route. Unused positive-cost activations can be deleted.

For known access, fix a topological order and let $z_v = \sum_{e \in \mathrm{in}(v)} x_e$ for $v \neq s$, with $z_s = 1$. For every original edge e with uncertified label j, impose

$$x_e \le \sum_{v:\ v \preceq \mathrm{source}(e),\ j \in Q(v)} z_v.$$

Any visited node counted on the right precedes the edge along the chosen path, by topological ordering. The inequality therefore requires an earlier available diagnostic. Conversely an admissible path supplies such a node for every used edge, so it satisfies all the inequalities. These constraints prove that the integer formulation is exact for temporal admissibility. Fractional relaxation gives a lower bound; a fractional solution is not an executable upgrade. Returned routes are independently checked for unit flow, cost, and admissibility, and unneeded active labels are removed before reporting their feasible cost.

G.3 Practical bounded searches

Beam search propagates the same distinct-label prefix states but retains at most w states per node, with $w \in \{8, 32, 128\}$, ordered by accumulated union cost and deterministic path tie-breaking. Its bound is a computational budget, not a theorem-level approximation factor. Marginal-label greedy activates the label with largest new-reachable-vertex benefit per cost and searches the induced graph; complementarities can defeat that score. Shortlist search generates 32 or 128 edge-additive shortest routes and reranks them by distinct-label cost. Under sparse access it filters these candidates for admissibility; it is not a sophisticated access-aware shortest-path oracle. We therefore report its coverage separately and use the global-access comparison for the principal matched-quality table.

G.4 Inherited inapproximability

Given a weighted set-cover instance on H elements and sets $S_1, \dots, S_p$ of weights c_j, construct nodes $0, \dots, H$. Between nodes $i-1$ and i, include a parallel edge labeled j whenever element i belongs to S_j. Distinct labels of any terminal path form a cover. Conversely, a cover induces a path by selecting a covering set at each layer; its used label cost is no larger than the cover cost. Applying both directions to optima proves exact equality, and a path approximation would give a cover approximation with the same ratio. The fixed-ε_0 $(1-\varepsilon_0)\log H$ hardness under $\mathrm{P} \neq \mathrm{NP}$ (deterministic versus nondeterministic polynomial time) thus follows from Dinur & Steurer (2014). This connects to classical labeled connectivity and red-blue covering (Carr et al., 2000; Hassin et al., 2007); Kumar et al. (2021) studies a more specialized geometric setting, not the origin of this combinatorial structure.

G.5 Finite-boundary test costs

Let a positive-sign log-likelihood increment have density f_μ for $\mathcal{N}(\mu, 2\mu)$, with $\mu = 1/c_j$. For a symmetric boundary A, the expected remaining query count $t(z)$ satisfies

$$t(z) = 1 + \int_{-A}^{A} t(y) f_\mu(y - z)\, dy, \quad |z| < A; \qquad t(z) = 0 \quad |z| \geq A.$$

Gauss–Legendre quadrature gives a finite linear system, evaluated again at $z = 0$. The implementation increases quadrature order with interval width divided by transition standard deviation. A fixed small order on a large interval can seriously underresolve the transition density; the adaptive order and the analytic bound $1 \leq t(0) \leq (A + K_\mu)/\mu$ are checked. Doubling the order is a numerical sensitivity check, not a proof of a uniform quadrature error bound. The finite-cost planner uses $t_j(0)$ rather than c_j as a distinct-label weight, valid for the all-positive fixed-path test protocol. This is not a solution to optimal finite-budget adaptive route selection.

H Previewing diagnostic difficulty: proofs and finite stopping

Theorem 12 (Exact second-order preview cost)**.** *Fix $0 < \epsilon < 1$, $\lambda_{\mathrm{pr}} > 0$, $I > 0$, and $\sigma > 0$. In the two-stage preview contract of Section H, the minimax all-positive cost among uniformly δ-safe policies with all-positive completion at least $1 - \delta$ in both worlds is $c_{\mathrm{easy}}\ell + (\lambda_{\mathrm{pr}}/I)\log\ell + O(1)$. The constants are allowed to depend on the fixed model parameters.*

This section proves Theorem 12. The two magnitude worlds are A and B: in A, branch 0 has magnitude 1 and branch 1 has magnitude ϵ; in B the magnitudes are exchanged. Both branch signs can vary independently. Before commitment, the only observations are independent previews with

distribution $\mathcal{N}(\mu_{\mathrm{pr}}, \tau^2)$ in A and $\mathcal{N}(-\mu_{\mathrm{pr}}, \tau^2)$ in B. They reveal nothing about either sign. After an almost-surely finite preview stage, the algorithm must choose exactly one branch. Its magnitude is disclosed, but its sign is not. The algorithm can now query that sign and either deploy or abstain, but cannot change branches or obtain further previews. Commitment itself is objective-neutral, unlike the labeled deployment edges in Section 2. This is an explicit extension of that model, not an unannounced zero-amplitude edge.

Write M for the preview count, N for the downstream sign-query count, $\lambda_{\mathrm{pr}} > 0$ for the cost per preview, $I = 2\mu_{\mathrm{pr}}^2/\tau^2$, $\ell = \log(1/\delta)$, $c_{\mathrm{easy}} = \sigma^2/2$, and $c_{\mathrm{hard}} = \sigma^2/(2\epsilon^2)$. All these model parameters except δ are fixed, with $I > 0$ and $0 < \epsilon < 1$. Let $\mathfrak{A}_\delta$ be the policies with finite expected counts, safety at level δ for every world and every sign pair, and all-positive completion probability at least $1 - \delta$ in both worlds. The minimax cost refers only to these two all-positive instances:

$$K_\delta = \inf_{\mathcal{A} \in \mathfrak{A}_\delta} \max_{w \in \{A,B\}} \mathbb{E}_{w,+}[\lambda_{\mathrm{pr}} M + N].$$

There is no requirement to find the positive branch on every mixed-sign instance. Such a requirement would contradict irreversible commitment based on sign-independent previews.

H.1 A BOUNDED GAUSSIAN OVERSHOOT

Lemma 13 (An elementary overshoot bound)**.** *Let Z_i be independent $\mathcal{N}(\mu, 2\mu)$ increments, $\mu > 0$, and $T_a = \inf\{n \geq 1 : \sum_{i=1}^n Z_i \geq a\}$, $a > 0$. Then*

$$\mathbb{E}T_a \leq \frac{a + K_\mu}{\mu}, \qquad K_\mu = \mu + 2\sqrt{\mu/\pi}.$$

The same upper bound holds for the first crossing of either a or $-a$.

Proof. The event $\{T_a > n\}$ implies $\sum_{i=1}^n Z_i < a$. For $n \geq 2a/\mu$, a Gaussian tail bound is exponentially decreasing in n, so $\mathbb{E}T_a < \infty$. Let $Z \sim \mathcal{N}(\mu, s^2)$, $s = \sqrt{2\mu}$. Its mean residual $r_{\mathrm{res}}(d) = \mathbb{E}[Z - d \mid Z \geq d]$ decreases with d. For completeness, the standard-normal hazard $h(z) = \varphi(z)/(1 - \Phi(z))$ satisfies $h'(z) = h(z)(h(z) - z) > 0$: for $z > 0$, $\int_z^\infty \varphi(u)\, du < \varphi(z)/z$ by integrating $u\varphi(u)$; for $z \leq 0$, $h(z) > 0 \geq z$. An increasing hazard makes the tail ratio $\mathbb{P}(Z > d + t)/\mathbb{P}(Z > d)$ decrease with d for every $t \geq 0$. Integrating this ratio proves the claimed monotonicity of r_{res}.

Immediately before any crossing, the residual distance d is positive. Conditional on crossing with that residual, the expected overshoot is at most

$$r_{\mathrm{res}}(0) = \mu + s\frac{\varphi(\mu/s)}{\Phi(\mu/s)} \leq \mu + s\sqrt{2/\pi} = K_\mu.$$

Averaging over the crossing time preserves this bound. Wald's identity applies because $\mathbb{E}T_a < \infty$ and the increments have finite absolute mean; hence $\mu\mathbb{E}T_a = \mathbb{E}\sum_{i=1}^{T_a} Z_i \leq a + K_\mu$. The two-sided exit occurs no later than the upper crossing. This explicit argument is sufficient here; more general sequential-testing and overshoot theory is classical (Wald, 1945; Wald & Wolfowitz, 1948; Lorden, 1970). □

H.2 A FINITE-CONFIDENCE LOWER BOUND

Proposition 14 (Cost of identifying a cheaper route)**.** *For the preceding model,*

$$K_\delta \geq c_{\mathrm{easy}}\ell - c_{\mathrm{hard}}\delta\ell - (c_{\mathrm{easy}} + c_{\mathrm{hard}})\log 2 \tag{11}$$

$$+ \min_{0 < e_{\mathrm{pr}} \leq 1/2} \left\{ \frac{\lambda_{\mathrm{pr}}}{I}\, \mathrm{kl}(1 - e_{\mathrm{pr}}, e_{\mathrm{pr}}) + (c_{\mathrm{hard}} - c_{\mathrm{easy}})\ell e_{\mathrm{pr}} \right\}. \tag{12}$$

The right-hand side can also be replaced by its maximum with zero.

Proof. First symmetrize any policy: use a fair coin to run either the policy or its reflected version, exchanging branch names and negating previews. Reflection exchanges worlds A and B and also relabels their signs. Uniform safety and all-positive completion are preserved. Each all-positive world

now has cost equal to the average of the original two costs, no larger than their maximum. Thus a lower bound for every symmetric policy is a minimax lower bound.

For a symmetric policy let e_{pr} be its probability of choosing the hard branch and m its expected preview count. Before commitment, the two worlds differ only in the preview distribution; internal randomization has the same law. Stopped change of measure and data processing through the chosen branch give

$$Im \geq \mathrm{kl}(1 - e_{\mathrm{pr}}, e_{\mathrm{pr}}). \tag{13}$$

If $m = \infty$ the result is immediate; otherwise the identity follows by truncation or Wald's identity for the preview log likelihood. Zero branch-selection error in both worlds requires infinite information.

Fix one magnitude world and its all-positive instance. For either branch j, change only its sign. The complete transcript KL is $\mathbb{E}_+ N_j / c_j$, since previews, commitment, disclosed magnitudes, and observations on the other branch have unchanged conditional laws. If q_j is the probability of deploying branch j on the all-positive instance, uniform safety under this alternative gives

$$\mathbb{E}_+ N_j \geq c_j \,\mathrm{kl}(q_j, p_j) \geq c_j(q_j \ell - \log 2), \qquad p_j \leq \delta.$$

Here p_j is the deployment probability under the sign-flipped alternative. The entropy bound $\mathrm{kl}(q, p) \geq q\ell - \log 2$ for $p \leq \delta$ does not require $q \geq \delta$. Summing the second inequalities, using $q_0 + q_1 \geq 1 - \delta$ and $q_{\mathrm{hard}} \geq e_{\mathrm{pr}} - \delta$, yields

$$\mathbb{E}_+ N \geq c_{\mathrm{easy}} \ell + (c_{\mathrm{hard}} - c_{\mathrm{easy}}) e_{\mathrm{pr}} \ell - c_{\mathrm{hard}} \delta \ell - (c_{\mathrm{easy}} + c_{\mathrm{hard}}) \log 2.$$

Combining with (13) proves the stated bound. Minimizing over $e_{\mathrm{pr}} \in (0, 1)$ is equivalent to minimizing over $(0, 1/2]$: reflecting an $e_{\mathrm{pr}} > 1/2$ preserves its binary KL but reduces the linear penalty. □

H.3 Exact second-order cost

Proof of Theorem 12. Set $w_{\mathrm{pr}} = \lambda_{\mathrm{pr}} / I$ and $\Delta_c = c_{\mathrm{hard}} - c_{\mathrm{easy}} > 0$. For $0 < e_{\mathrm{pr}} \leq 1/2$,

$$\mathrm{kl}(1 - e_{\mathrm{pr}}, e_{\mathrm{pr}}) = (1 - 2e_{\mathrm{pr}}) \log((1 - e_{\mathrm{pr}})/e_{\mathrm{pr}}) \geq \log(1/e_{\mathrm{pr}}) - 2/e - \log 2,$$

since $2e_{\mathrm{pr}} \log(1/e_{\mathrm{pr}}) \leq 2/e$ and $(1 - 2e_{\mathrm{pr}}) \log(1 - e_{\mathrm{pr}}) \geq -\log 2$. For sufficiently large ℓ, minimizing $w_{\mathrm{pr}} \log(1/e_{\mathrm{pr}}) + \Delta_c \ell e_{\mathrm{pr}}$ gives $e_{\mathrm{pr}} = w_{\mathrm{pr}}/(\Delta_c \ell)$ and minimum $w_{\mathrm{pr}} \log \ell + O(1)$. Proposition 14 therefore implies $K_\delta \geq c_{\mathrm{easy}} \ell + (\lambda_{\mathrm{pr}}/I) \log \ell - O(1)$.

For the upper bound, run a preview likelihood-ratio test with boundaries $\pm b$, where $b = \log \ell$ for $\ell > 1$. Its wrong-branch probability is at most $e^{-b} = 1/\ell$ by the likelihood-ratio martingale bound, and Lemma 13 gives $\mathbb{E}M \leq (b + K_I)/I$. Commit to the branch it identifies as easy. After observing that branch's magnitude γ, run an independent sign test with boundaries $\pm \ell$. Conditional on either selected branch, an incorrect sign classification has probability at most $e^{-\ell} = \delta$. Therefore the deployment error bound and all-positive completion bound hold after averaging over the branch choice. There is no need to spend deployment-error probability on selecting the cheaper branch: both branches are checked after selection.

Under a positive selected sign, its likelihood-ratio increments have mean $\mu_\gamma = 2\gamma^2/\sigma^2$ and variance $2\mu_\gamma$. Applying Lemma 13 to the sign test gives

$$\mathbb{E}N \leq (c_{\mathrm{easy}} + \Delta_c e_{\mathrm{pr}})\ell + \max_{\gamma \in \{1, \epsilon\}} c_\gamma K_{\mu_\gamma} \leq c_{\mathrm{easy}} \ell + O(1).$$

Adding preview cost proves the matching upper bound. The forced-preview heuristic threshold

$$b = \log(\max\{2, \Delta_c I \ell / \lambda_{\mathrm{pr}}\})$$

differs from $\log \ell$ by a fixed constant asymptotically and has the same guarantee. It balances a leading proxy for preview expenditure, $\lambda_{\mathrm{pr}} b / I$, against the hard-route penalty $\Delta_c \ell e^{-b}$; it is not claimed to be exactly optimal at finite confidence. □

Boundaries of the result. For $I = 0$, previews identify neither route and blind commitment is sufficient. For $\epsilon = 1$, routes have equal difficulty and previews are unnecessary. The $O(1)$ constants need not be uniform as $\epsilon \downarrow 0$, $I \downarrow 0$, or $\lambda_{\mathrm{pr}} \to \infty$. In particular, the asymptotic result does not imply that previews are worthwhile at a given moderate confidence level. A sign-independent preview is an efficiency signal, not a substitute for a trusted deployment query. Controlled sensing already formalizes adaptive choice of observations (Chernoff, 1959; Nitinawarat et al., 2013; Nitinawarat & Veeravalli, 2015); the result here isolates the additional cost of a reversible information stage preceding an irreversible, separately certified decision.

H.4 BLIND COMMITMENT AND THE COMPARATOR'S INFORMATION

If no informative preview is available, the branch-choice distribution must be identical in the two magnitude worlds. If the probability of selecting branch 0 is q, the two all-positive leading costs are $qc_{\text{easy}} + (1-q)c_{\text{hard}}$ and $qc_{\text{hard}} + (1-q)c_{\text{easy}}$. The larger is minimized at $q = 1/2$, yielding $(c_{\text{easy}} + c_{\text{hard}})\ell/2 + o(\ell)$. An informed branch selector, which knows the easy branch for free, instead has leading coefficient c_{easy}. The ratio $(1+\epsilon^{-2})/2$ is therefore an information-value comparison, not an efficiency failure relative to a same-information algorithm. We do not use its unboundedness as the main novelty claim.

H.5 BELLMAN OPTIMAL STOPPING WITH AN EXPLICIT SKIP ACTION

The numerical comparison fixes the downstream sign test. With $\pi(z) = (1+e^{-z})^{-1}$ and easy/hard test lengths $t_{\text{easy}}, t_{\text{hard}}$, define

$$g(z) = t_{\text{easy}} + \min\{\pi(z), 1-\pi(z)\}(t_{\text{hard}} - t_{\text{easy}}).$$

For $W_+ \sim \mathcal{N}(I, 2I)$ and $W_- \sim \mathcal{N}(-I, 2I)$,

$$\begin{aligned} V_0(z) &= g(z), \qquad V_h(z) = \min\{g(z), \lambda_{\text{pr}} + H_{h-1}(z)\}, \\ H_{h-1}(z) &= \pi(z)\mathbb{E}V_{h-1}(z+W_+) + (1-\pi(z))\mathbb{E}V_{h-1}(z+W_-). \end{aligned} \tag{14}$$

The skip action is $g(z)$; hence $V_h(0) \le g(0)$ for the exact recursion. Grid/quadrature approximations have separately reported sensitivity, not an exact numerical error certificate.

Fix the downstream sign-testing rule at boundaries $\pm\ell$. Its expected length under either reflected sign is the same for a given magnitude. Let the easy and hard mean lengths be t_{easy} and t_{hard}, evaluated by the integral equation in Appendix G. Under the symmetric prior over magnitude worlds, the sufficient statistic z has posterior $\pi(z)$. Stopping chooses the posterior-more-likely easy branch and incurs $g(z)$ in Section H. Continuing costs λ_{pr} and yields a log-likelihood increment with law $\mathcal{N}(I, 2I)$ in the positive world and $\mathcal{N}(-I, 2I)$ in the negative world. Conditioning on that next observation and applying backward induction proves (14) exactly for every finite horizon. The symmetric solution also has equal costs in the two worlds and is minimax within this fixed-terminal-test finite-horizon model.

Stopping immediately is feasible, hence $V_h(0) \le g(0)$ for every h. Moreover $V_{h+1} \le V_h$, and the nonnegative sequence has a limit. We do not assert an error rate for approximating that infinite-horizon limit by a fixed cap. The theoretical asymptotic construction in Theorem 12 has an unbounded stopping time; a fixed cap is a separate finite-confidence baseline.

Numerical solutions use a grid on $[-16, 16]$ with spacing 0.025 and horizon 128, then spacing 0.0125 and horizon 256 for sensitivity. Normal transition bin masses are computed from the Gaussian CDF and applied by convolution. Outside the grid the value is replaced by t_{easy}; the stopping-value difference there is at most $(t_{\text{hard}} - t_{\text{easy}})e^{-16}$. This pointwise tail observation alone is not a global discretization certificate. The returned policy interpolates the continuation values and can stop without any preview. Numerical Bellman values are reported separately from Monte Carlo executions and from the universal finite-confidence lower bound.

I ASSUMPTION-LEVEL COMPARISON WITH NEARBY RESULTS

Explore–verify and finite-confidence exploration. Karnin (2016) already gives moderate-confidence discovery followed by high-confidence verification, with the verification complexity determining the leading term. We specialize that interface to utility-qualified positive paths and checked blocking sets. Poiani et al. (2026) supplies non-asymptotic guarantees for Track-and-Stop and Sticky Track-and-Stop; the present finite label-specific bound is neither the first non-asymptotic analysis nor a claimed general improvement over those algorithms. The comparison below concerns assumptions, not a priority claim for the exploration/verification split.

Closest work	Its relevant setting	Relation and boundary here
Degenne & Koolen (2019); Kaufmann et al. (2016)	Fixed-confidence exploration with multiple valid answers; stopped information inequalities.	The global positive-path answer set is a specialization. Our procedure specializes explore–verify to useful routes or checked obstructions; it is not a new general information bound.
Chen et al. (2014)	Identifying the best member of a structured decision class through arm observations and an offline optimization oracle.	We ask for any positive utility-qualified path. Returning an approximate feasible path affects cost, not the validity of a separately tested deployment.
Ünlüyurt (2004); Deshpande et al. (2014)	Sequential costly-component tests; stochastic Boolean evaluation commonly assumes product-distributed bits and one-time exact revelation.	Here signs are fixed unknowns, repeated noisy queries certify them, and traversing an upgrade path can remove future observations. Evaluating connectivity itself is not a new problem.
Hassin et al. (2007); Carr et al. (2000)	Label-cost connectivity and covering combinatorics.	Supplies the non-additive route objective and relevant hardness. No new minimum-label-path algorithmic guarantee is claimed.
Keener (1984); Lalley & Lorden (1986)	Second-order sequential experimental design and related controlled first-passage problems. Keener's formulation has finite parameter and repeatable-experiment sets with small sampling costs.	The two-stage preview model imposes one irreversible commitment and a separate sign-safety obligation. We give a specialized finite bound and explicit coefficient; the $\log(1/e) + e\ell$ balancing principle is classical.
Nitinawarat & Veeravalli (2015)	Controlled Markov observations with nonuniform control costs and strictly positive observation transition probabilities (Equation (3.1)).	A permanent loss of a diagnostic creates zeros and may violate that theorem's assumptions. This is a boundary to direct application, not a claim that all controlled-sensing models exclude access constraints.
Necula (1997); Rothermel & Harrold (1997)	Checkable code evidence and safe regression-test selection.	Our finite integer normalizer independently establishes a sufficient transfer relation; direct validation and verification work remain charged. It is neither general program equivalence nor a new proof-carrying-code architecture.

What the present results add at the level of a decision. The construction does not replace existing statistical or combinatorial theories. It specializes explore–verify with optimistic path-restricted search and certified obstructions, then identifies contracts under which access and transfer can be composed. Its feasible-instance leading constant is inherited in form from the multiple-answer problem; the label-local screening accounting is an explicit implementation property. The finite-family coupling bound explains why a representative-only approach cannot repair the latter; the exact release rule and executable checker give two restricted alternatives. The preview result remains a separate two-branch calculation, not a general graph planner.

Further neighboring formulations. Linear best-arm and transductive design allow measurement directions distinct from answer directions (Soare et al., 2014; Fiez et al., 2019); exact sign-sharing labels are much more restrictive than general linear representations. A trusted-only confidence ellipsoid could certify arbitrary linear directions using the self-normalized theory of Abbasi-Yadkori et al. (2011); validity under adaptive direction selection follows on the same simultaneous event and is not a separate contribution here. Safe and stagewise optimization (Sui et al., 2015; 2018), conservative bandits (Kazerouni et al., 2016), safe Markov decision process (MDP) exploration (Turchetta et al., 2016), and high-confidence policy improvement (Thomas et al., 2015) motivate treating abstention and achieved utility jointly. Performative prediction concerns deployment-dependent distributions

(Perdomo et al., 2020); our conditional query law is instead fixed once a label is specified. Learned preference and reward models (Christiano et al., 2017; Lindner et al., 2022), misspecified or corrupted reward channels (Hadfield-Menell et al., 2017; Everitt et al., 2017), and broader safety concerns (Amodei et al., 2016) are not resolved by a Gaussian sign ledger.

J Experimental protocols and uncertainty

J.1 Experimental suites and independent units

The six suites evaluate random global graphs, distractor and ordering controls, checkpoint access, packet composition, solver accuracy, and source verification. An execution is not an independent task. The random-global study has 80 independent graph seeds, reused at two label counts; the solver study has 50 seeds per global width condition and 25 per local width/access condition, with seed reuse across widths and settings. Paired intervals resample these seed clusters and preserve shared settings. The checkpoint generator has 120 independent instances, each with eight paired observation streams. The code study has four fixed tasks and 500 orderings per task; its timings have no claimed population confidence interval over programming tasks.

The bootstrap uses 4,000 resamples and reports pointwise percentile 95% intervals, not simultaneous guarantees across all tables. Bootstrap seeds are 915 for mean summaries and 916 for paired relative reductions. Graph-paired noise uses independent per-label streams shared across policies, so changing one label's query count does not change the observations available from another. A policy obtains outcomes only through the charged query interface. Recorded outcomes include actual gain, terminal flag, safe-completion flag, harm, abstention and interruption separately. No budget or solver failure is converted into an infeasibility statement.

J.2 Global random graphs and verified abstention

The graph generator uses six layers and width three, 12 or 24 label IDs, and a shared-label mixture probability 0.35. Magnitudes follow the fixed generator; signs are independently negative with probability 0.35. Positive edge gains are sampled uniformly in $[0.15, 0.85]$, with amplitudes scaled by the known magnitudes. Even graph seeds use target zero; odd seeds use target 2.5. There are 59 feasible instances at 12 labels and 61 at 24 labels. Exact offline planning with true signs is used only by the reporting harness. Each policy uses randomized test order, eight paired Gaussian streams, $\sigma = 1$, and each of $\delta \in \{0.05, 10^{-4}, 10^{-8}, 10^{-16}\}$. The generous 200,000-query and 100,000-epoch guards do not bind.

The full and route-restricted methods use identical screening boundaries, fresh verification, epoch allowances, and negative-cut checks. The latter retains screening results only within an epoch and screens at most one time per label there. The cheap-rejection ablation has no preliminary route screening: positive certificates accumulate until a later cheap rejection rules out the candidate; it is not claimed optimal. Symmetric testing uses permanent two-sided certificates. Optional testing selects symmetric whenever $B \geq A_1/2$ before observations, with identical streams/order on that branch. That numerical factor is fixed, not chosen from the evaluation outcomes.

Table 5: Mean queries on the 40 infeasible graph instances, eight streams each. These are explicit certified negative outputs, not repeated restarts or budget failures.

δ	Symmetric	Cheap	Full	Selective	Optional
0.05	23.73	37.36	88.38	36.04	23.73
10^{-4}	43.48	57.36	111.13	50.25	47.28
$1e-08$	72.73	85.73	133.21	68.47	68.47
$1e-16$	131.18	140.29	171.16	102.19	102.19

At $\delta = 0.05$, each policy has 1,280 runs: 960 on feasible graphs and 320 on infeasible graphs. Selective and full screening have no wrong final answer. Symmetric/optional each have two harmful routes and one false abstention; cheap rejection has two harmful routes and no false abstention. One harmful route per such policy occurs on a feasible instance and one on an infeasible instance. At the

smaller displayed confidence levels no incorrect final output is observed. A terminal flag on one of those harmful routes is not safe completion. Every run halts below both guards.

The paired reduction of selective relative to full screening on feasible graphs is 67.29% [64.77,69.74] at 0.05, 62.58% [60.03,65.11] at 10^{-4}, 56.55% [53.95,59.14] at 10^{-8} and 47.67% [45.16,50.18] at 10^{-16}. This is not a claim of dominance over other methods: cheap rejection uses fewer mean queries in all four feasible conditions. On infeasible graphs at 10^{-16}, selective saves 22.10% [19.07,24.91] relative to symmetric testing; at 0.05 it instead spends more. The theorem is asymptotic; these measurements also expose unfavorable finite-confidence regimes.

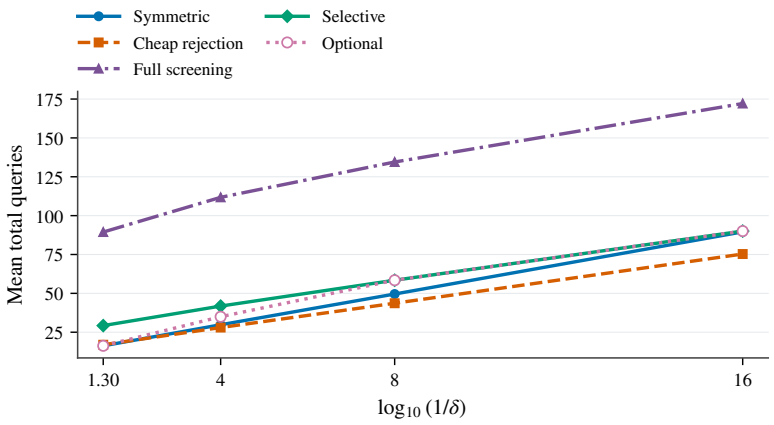

Figure 3: Random feasible instances: total work, including screens, unsuccessful attempts and gates. Confidence is on the horizontal scale; repeated observation streams are paired within graph seeds.

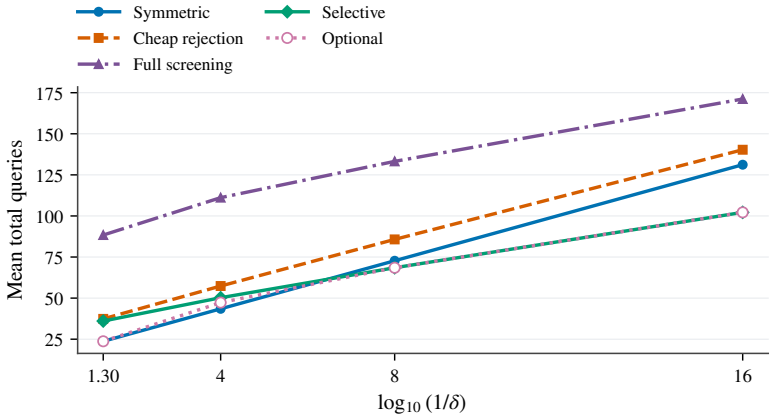

Figure 4: Random infeasible instances: full-screen and selective methods halt through a verified blocking set. No sample is removed for infeasibility.

J.3 Irrelevant routes and random ordering

Five rejected routes each have one positive and one negative label with magnitude one; the useful route is a single label with magnitude 0.5. Twenty added alternatives, when present, have positive magnitude 0.3 and cost greater than the useful route. There are either 11 or 31 labels. Candidate route tests use either their path order (positive first on the traps) or an independently shuffled order. With 300 streams, five confidence levels and five methods this yields 30,000 records. All-zero distractor costs or zero known labels are never supplied: even unvisited labels remain in the confidence allocation. The reporting-only noiseless screened set verifies that the expensive alternatives are irrelevant to optimistic search.

At $\delta = 0.05$ with 20 alternatives and positive-first order, normalized mean queries are 18.20 for symmetric, 18.70 for cheap rejection, 259.38 for full screening and 22.53 for selective. Optional equals symmetric. At 10^{-80} the respective values are 7.18, 4.80, 8.85 and 2.51. At that extreme confidence, removing the alternatives and randomizing order gives 2.38 for selective, 3.49 for cheap rejection and 5.87 for symmetric. The random-order control weakens the contrived trap advantage, as expected, but does not make early certification of doomed routes cost-free. Extremely small δ tests expected-work scaling only.

J.4 The checkpoint contract and a composed packet example

Four blocks have eight disjoint labels each, depth four and width two. At a block entry all eight diagnostics are available; no interior node permits a diagnostic query. Blocks share only their boundary mode. The generator uses a 30% negative-sign probability. Every route in each block has the same known positive-sign gain, respectively $0.5, 0.75, 1, 1.25$; the global target is their sum 3.5. Thus eligibility factorizes exactly as in Theorem 2, rather than by silently imposing stronger quotas on a generic global utility target. There are 26 feasible instances among 120. Queries record their block identity, and tests enforce that an earlier or future diagnostic cannot be called in the current block. A certified block is deployed before proceeding, so retained prefix gain is a separate outcome from terminal completion.

Each instance has eight noise streams, $\delta \in \{0.05, 10^{-8}\}$, budgets 20, 60 and 200,000, and four methods. No observed harm occurs in this suite. At $\delta = 0.05$, budget 60, symmetric/optional retain mean gain 1.15 and complete safely in 14.7% of runs; route screening retains 0.75 with 1.0% completion, and full screening 0.35 with zero completion. At 10^{-8} and the nonbinding budget, all preserve mean gain 1.26 and complete in $26/120$ of instances; the others halt at checked infeasibility with retained prefixes. Mean queries are 96.68 for symmetric, 110.75 for selective and 172.06 for full. These results check the joint contract, not a new finite-budget superiority claim.

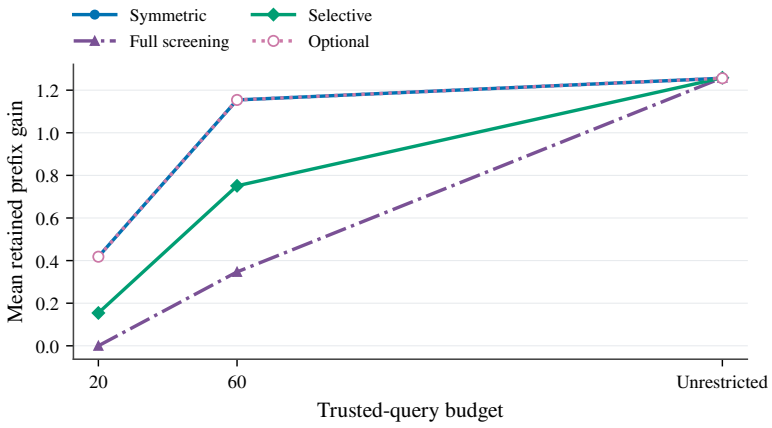

Figure 5: Retained gain, not a binary terminal flag, at $\delta = 0.05$. "Unrestricted" is the nonbinding 200,000-query cap. The prespecified optional policy coincides with symmetric testing.

For packets, $F = 64$, $m = 32$, $r = 3$ and allowance $\beta = 0.05$ are fixed before the checks. The final $v \in \{0, 1, 4, 12\}$ members by public ID violate the family relation, and the planner prefers high IDs inside the packet. This deliberately stresses biased *subset* selection without violating uniform packet issuance. There are 3,000 fresh permutations per violation-count/representative-sign condition. A representative has magnitude 0.5, noise one, and is certified at $\delta = 0.01$; the pipeline materializes only the actual released packet members (or checked-valid members after a dirty inspection). It never substitutes a favorite outsider.

With one violation and a positive representative, clean inspections occur in 49.8% and selected harmful release in 4.93%; the exact packet's joint allowance is $3/64$ and the overall uniform budget is $0.01 + 0.05$. With a negative representative, 0.67% of runs falsely authorize a route, below its sign-test allowance. A nominal package selector checks demands $q \in \{3, 8, 32, 48, 63\}$ before inspection. It

selects individual checking for three and eight slots, and packages with only three unchecked slots for the larger demands. This is intentionally not a claim of dramatic statistical reuse. Admission uses exact rational hypergeometric probabilities in the implementation. Validation effort (32 member checks in the fixed-package suite) is recorded separately from sign queries.

J.5 Certified solver optima, failure rates and LP gaps

There are 600 graph/access instances: label counts 32, 64, 96; widths three and six; depth ten; 50 seed clusters per width under global access and 25 under each local availability probability 0.5 and 0.8. The same seeds are paired across widths and label counts, not declared independent again. Nodes number 29 or 56. The solver receives a 60-second cap; every instance resolves within that cap. All 583 feasible cases return solver status "optimal" with zero solver MIP gap; 17 are certified infeasible. Objectives and returned paths are checked independently by the label-union and access feasibility functions. These are numerical MIP certificates within HiGHS tolerances, not rational-arithmetic proofs.

The solver formulation and objective are identical across exact and heuristic methods. Heuristics use beam widths 8/32/128, marginal-label greedy and additive shortlists of size 32/128. Under local access, the returned route is checked against the actual availability sequence. A shortlist that contains no admissible route is recorded as a failed search, not a proof of infeasibility. All comparative gaps are against the proven optimum on the same instance. Conditional gaps among successful searches are accompanied by found/feasible counts.

Table 6: Global-access optimality gaps and search-budget sensitivity. Percentages are relative to the MIP optimum; the 95% interval resamples 50 seed clusters while preserving both widths. All global heuristics find a feasible route.

Labels	Method	Mean gap (%)	95% interval	Max gap (%)	Time (ms)
32	Beam-8	1.68	[0.96, 2.49]	25.71	1.98
32	Beam-32	0.07	[0.00, 0.17]	3.55	6.58
32	Beam-128	0.00	[0.00, 0.00]	0.00	23.45
32	greedy	21.30	[17.79, 24.85]	72.29	2.13
32	Shortlist-32	23.35	[19.37, 27.62]	111.50	1.00
32	Shortlist-128	17.40	[13.93, 21.14]	111.50	2.63
64	Beam-8	2.46	[1.45, 3.60]	26.21	2.10
64	Beam-32	0.20	[0.00, 0.53]	13.95	6.14
64	Beam-128	0.00	[0.00, 0.00]	0.00	24.40
64	greedy	23.47	[19.95, 27.30]	90.66	3.70
64	Shortlist-32	26.04	[22.63, 29.47]	100.58	0.91
64	Shortlist-128	18.51	[15.71, 21.32]	82.71	2.36
96	Beam-8	3.65	[2.39, 5.01]	35.35	1.81
96	Beam-32	0.32	[0.09, 0.59]	7.96	7.24
96	Beam-128	0.00	[0.00, 0.00]	0.00	22.47
96	greedy	20.01	[16.57, 23.54]	82.37	5.70
96	Shortlist-32	21.81	[17.99, 25.67]	75.07	0.94
96	Shortlist-128	17.62	[14.08, 21.50]	75.07	2.43

Table 7: Actual LP gaps and solver outcomes. The LP gap is $(C^* - C_{\mathrm{LP}})/C^*$ on feasible instances. Local access cases were not repaired to force feasibility.

Labels	Access	Instances	Optimal	Infeasible	LP gap mean (%)	Max (%)	MIP (s)
32	Global	100	100	0	30.86	57.42	1.10
32	0.5	50	45	5	27.34	56.30	0.89
32	0.8	50	50	0	29.92	56.65	1.13
64	Global	100	100	0	25.53	49.88	1.27
64	0.5	50	43	7	22.59	47.43	1.07
64	0.8	50	50	0	22.63	45.67	1.12
96	Global	100	100	0	20.84	52.50	1.01
96	0.5	50	46	4	18.47	44.82	0.82
96	0.8	50	49	1	19.36	47.18	0.93

Table 8: Local-access solver gaps. Mean, interval and maximum are conditional on a returned feasible route; failures are explicitly shown. Each access/width setting has 25 seed clusters.

Labels	Access	Method	Found/feasible	Mean gap (%)	95% interval	Max (%)
32	0.5	Beam-8	45/45	1.56	[0.40, 3.45]	35.70
32	0.5	Beam-32	45/45	0.07	[0.00, 0.23]	3.29
32	0.5	Beam-128	45/45	0.00	[0.00, 0.00]	0.00
32	0.5	greedy	45/45	22.85	[16.02, 29.55]	85.39
32	0.5	Shortlist-32	14/45	18.09	[8.10, 29.44]	63.13
32	0.5	Shortlist-128	17/45	13.45	[5.69, 22.60]	58.43
32	0.8	Beam-8	50/50	0.86	[0.40, 1.43]	11.39
32	0.8	Beam-32	50/50	0.06	[0.00, 0.18]	3.04
32	0.8	Beam-128	50/50	0.00	[0.00, 0.00]	0.00
32	0.8	greedy	50/50	21.77	[15.79, 27.54]	72.29
32	0.8	Shortlist-32	44/50	20.61	[15.26, 26.61]	63.13
32	0.8	Shortlist-128	44/50	15.39	[10.37, 21.09]	58.43
64	0.5	Beam-8	43/43	1.38	[0.24, 2.85]	22.50
64	0.5	Beam-32	43/43	0.01	[0.00, 0.03]	0.45
64	0.5	Beam-128	43/43	0.00	[0.00, 0.00]	0.00
64	0.5	greedy	43/43	21.52	[15.82, 28.64]	115.12
64	0.5	Shortlist-32	16/43	28.39	[16.50, 41.91]	81.18
64	0.5	Shortlist-128	15/43	17.61	[9.66, 26.20]	43.46
64	0.8	Beam-8	50/50	1.91	[0.64, 3.40]	22.50
64	0.8	Beam-32	50/50	0.37	[0.00, 1.03]	13.95
64	0.8	Beam-128	50/50	0.00	[0.00, 0.00]	0.00
64	0.8	greedy	50/50	22.96	[17.11, 29.56]	115.12
64	0.8	Shortlist-32	41/50	27.53	[22.42, 33.10]	84.36
64	0.8	Shortlist-128	40/50	20.84	[16.07, 25.65]	82.71
96	0.5	Beam-8	46/46	1.52	[0.65, 2.53]	15.66
96	0.5	Beam-32	46/46	0.24	[0.00, 0.52]	4.59
96	0.5	Beam-128	46/46	0.00	[0.00, 0.00]	0.00
96	0.5	greedy	46/46	25.92	[21.21, 30.73]	58.20
96	0.5	Shortlist-32	16/46	16.14	[9.80, 23.53]	56.18
96	0.5	Shortlist-128	18/46	11.37	[6.66, 17.20]	46.72
96	0.8	Beam-8	49/49	1.83	[0.70, 3.36]	26.18
96	0.8	Beam-32	49/49	0.37	[0.00, 1.01]	13.49
96	0.8	Beam-128	49/49	0.00	[0.00, 0.00]	0.00
96	0.8	greedy	49/49	23.93	[19.56, 28.37]	76.99
96	0.8	Shortlist-32	36/49	24.60	[18.18, 30.75]	56.18
96	0.8	Shortlist-128	36/49	19.68	[14.18, 25.58]	56.07

Beam-128 attains the optimum on all 300 measured global instances; Beam-32 has mean gaps 0.07%, 0.20% and 0.32%. This is descriptive evidence for a bounded planner, not a general polynomial-time optimality theorem. The theoretical exact objective remains computationally hard. Runtime is descriptive for these measured instances and does not establish a universal timing ranking; the principal comparison is the gap to a certified optimum.

J.6 Order-matched source verification with total runtime

The four fixed tasks are absolute value, integer division by three, greatest common divisor with 12, and integer square root. Each has a 256-input finite domain, one representative, one baseline, 40 equivalent ring rewrites and 24 boundary mutations. A four-input fingerprint proposes the 64-member family but is not a certificate. The verification rule proves exact equivalence for its restricted total integer grammar; it does not infer sign transfer between non-equivalent edits. Candidates are considered in 500 independently permuted orders, shared by both policies. Both return the first eight beneficial candidates; an assertion checks identical IDs and gain for every pair.

Executable patch bank. The baseline and representative expressions are:

Task	Baseline	Representative
Integer square root	`round(math.sqrt(x))`	`math.isqrt(x)`
Floor division	`int(x/3)`	`x//3`
Absolute value	`max(0,x)`	`abs(x)`
Greatest common divisor (GCD) with 12	`12 if x%12==0 else 1`	`math.gcd(x,12)`

The square-root and GCD tasks use nonnegative integer domains; the other two use $-128, \ldots, 127$. Outputs are judged by a fixed reference implementation, and utility is domain-average exact-output accuracy relative to the baseline. Forty expressions per task are equivalent rewrites, while 24 have conditional boundary mutations. A four-input pilot fingerprint groups candidates by observed outputs. Full-domain execution later determines which collisions violate the proposed relation and which are actually harmful; the generator does not simply flip an abstract utility label. These are controlled arithmetic fixtures, not mined patches from an external repository.

A release batch replaces separate initially identical copies of a function. It does not compose eight alternative patches onto the same mutable function or count their gains as though it did. Mean gain is summed across released copies. This deliberately simple deployment contract makes verification cost and harmful releases measurable without hiding composition assumptions.

The direct policy caches the same 256 trusted reference answers as the hybrid and stops after the same eight candidates. Both pay the same initial 768 baseline, representative and fingerprint executions; each additional full-domain member check costs 256 executions. The hybrid additionally performs proofs on demand and skips full-domain candidate execution only after a successful proof. No precomputed candidate-utility vector is supplied. The runner compiles and executes actual requested source programs. The deployment protocol still releases separate function copies; it does not establish that this is better engineering than deploying a representative once.

Table 9: Per-task order-matched results, 500 paired orderings per fixed task. Total runtime includes reference acquisition, compilation, execution, fingerprints and on-demand proofs.

Task	Policy	Executions	AST nodes	Gain	Total ms
abs	direct	3423.74	0.00	3.47	2.54
abs	hybrid	2854.40	216.67	3.47	3.26
floor3	direct	3147.26	0.00	2.24	2.28
floor3	hybrid	2396.16	177.09	2.24	2.59
gcd12	direct	3147.26	0.00	3.94	2.55
gcd12	hybrid	2396.16	218.13	3.94	3.04
isqrt	direct	3317.25	0.00	3.33	2.46
isqrt	hybrid	2506.24	223.17	3.33	3.04

Timing measurements alternate which policy runs first across orderings. Mean executions fall from 3,258.88 to 2,538.24, but mean total time increases from 2.46 to 2.98 milliseconds. The 208.77 additional examined syntax nodes are not treated as equivalent to program executions. A replay re-executes the same scientific decisions; timing is intentionally excluded from exact reproduction. These fixtures demonstrate a charged, sound equivalence interface, not broad speedup or evidence on independent naturally occurring programming tasks.

J.7 Outcome validation

Repeated executions check scientific outcomes against the retained run, with numerical tolerance for solver objectives. Elapsed times and solver-internal search effort are excluded. Repeated executions are checks of the same scientific decisions, not additional independent evidence. Automated checks do not replace independent proof review.

K Additional access, utility, and testing studies

These studies examine arbitrary diagnostic access, finite-confidence planning, and the preview, sharing, and unknown-mean contracts separately. Their results do not establish a single algorithm that composes every extension. The arbitrary-access policy is the symmetric-ledger baseline in

Appendix D, not the checkpoint procedure. Code-sharing checks reuse the four fixed tasks in Appendix J and are not additional independent programming tasks. Results from distinct protocols are not pooled into the confidence intervals for the main comparisons.

K.1 Randomization and statistical units

A seed–condition–policy execution is not independent of other executions that reuse its seed. Paired bootstrap intervals use 10,000 resamples of independent seed indices and bootstrap seed 732029. Pointwise mean intervals use Student-t half-widths; neither interval construction adjusts for simultaneous interpretation of the full parameter grid. Mirrored preview worlds are averaged within seed before uncertainty estimation. Cross-task arithmetic summaries describe four fixed fixtures, not random draws from a population of programs.

The per-label oracle uses independently seeded streams shared across compared policies within a seed, so a policy's nth query of a label sees the same outcome. A policy receives only queried values; true signs and diagnostic feasibility are available only to the reporting harness. Fresh testing epochs advance the streams rather than reusing observations as independent data. The implementation epoch guard is 100,000 and was never reached in a feasible completed experiment.

K.2 Joint negative edits and local access

The study uses 400 paired graph seeds. The layered generator has depth six, width three, ten labels, and shared-label probability 0.4. Each label is negative with probability 0.3. Each node independently provides each diagnostic with probability $a_Q \in \{0.2, 0.4, 0.7\}$, including the root; no route is repaired or planted to ensure success. An omniscient sign-aware admissibility check is used only to report which instances are feasible. Positive edge gains are sampled independently from $[0.1, 1]$ and converted to amplitudes by dividing by the known magnitude. Both compared policies see the same magnitudes, amplitudes, and declared access sets.

Budgets are 12, 40, and 100,000 queries. The last cap is not reached and represents completed policy runs rather than an infinite-budget theorem. Feasibility rates are 18.25%, 45.0%, and 69.75% for access probabilities 0.2, 0.4, and 0.7. At $a_Q = 0.7$ and budget 40, the stop and replan policies complete in 38.0% and 68.75% of all 400 runs, with mean gain 1.32 and 2.31. The reported completion and gain averages include infeasible instances; conditioning on feasibility would define a different estimand. At budget 12, the gains are 1.14 and 1.44, illustrating genuine retained partial progress rather than an all-or-nothing terminal reward. Replanning cannot restore an unqueried diagnostic after its mode has been left.

At $a_Q = 0.7$ with effectively uncapped queries, completion is 38.0% versus 69.0%, still below the 69.75% omniscient feasible fraction for replanning. Wrong negative certificates and irreversible route choice both remain potential failure sources. Blocked fractions and mean queries conditional on becoming blocked are recorded separately from ordinary budget exhaustion. No harmful edge was observed in this access suite; the exact sign-ledger event, not this count, supports the uniform bound.

K.3 Finite-confidence cost and utility frontier

Finite-cost planning uses 200 paired seeds, 14 labels, depth seven, width three, and magnitudes drawn log-uniformly from 0.3 to 2.5. The confidence levels are 0.05 and 10^{-8}. Both asymptotic and finite-cost objectives use the same MIP planning interface and the same eventual sign tests. Planned expected query lengths at $\delta = 0.05$ are 11.52 for inverse information and 11.51 for finite-cost weights. Their paired difference is -8.23×10^{-3} $[-1.65 \times 10^{-2}, -1.71 \times 10^{-3}]$ queries, a very small absolute effect despite being consistently nonpositive on changed optimum routes. At 10^{-8} it is -3.07×10^{-3} $[-7.81 \times 10^{-3}, 0]$. Preprocessing takes about 27 and 23 ms, respectively, and is additional to solver time. Sampled execution counts are also reported; their noise should not be confused with exact mean-cost ordering.

The utility-target suite uses 100 fixed heterogeneous-gain graphs and targets $u = 1, 2, 3, 4, 5$. The feasible fractions are $1, 1, 1, 0.99, 0.24$. Mean minimum union costs conditional on feasibility are 1.24, 1.25, 1.32, 1.66, and 2.40. These are deterministic planning outcomes on random instances, not sign-learning runs. Within any fixed instance, $C_\theta(u)$ is nondecreasing; across targets, conditional

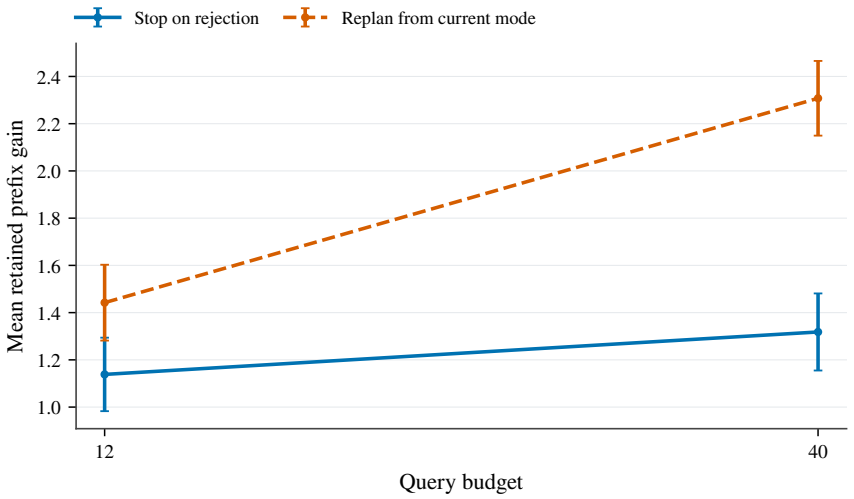

Figure 6: Actual retained prefix gain under mixed signs and local access ($a_Q = 0.7$), including infeasible and blocked runs. Larger query budgets permit additional replanning; they do not change the per-run confidence threshold $\delta = 0.05$.

means also change because the feasible population changes. We provide raw per-seed values rather than interpreting the conditional mean curve as a universal scaling law.

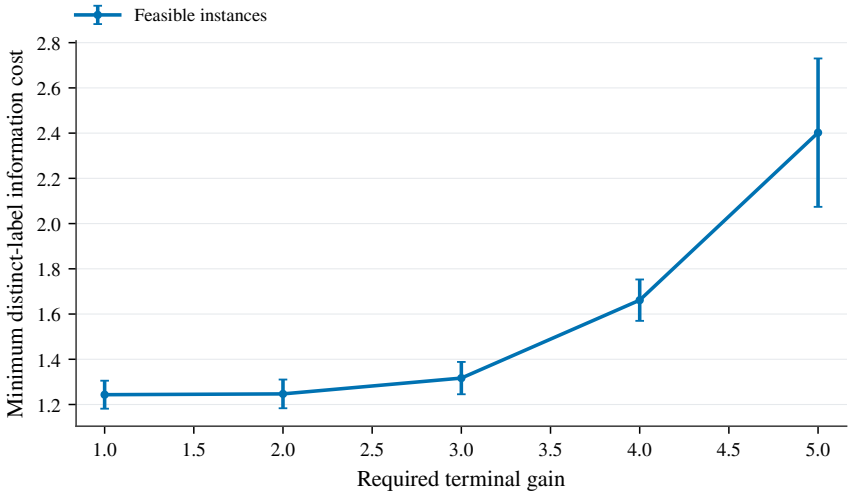

Figure 7: Minimum query-information cost versus required terminal gain, conditional on feasibility. Only 24 of the 100 graphs meet target five, so its interval summarizes a different feasible subset.

K.4 Optional preview experiments and numerical sensitivity

The grid uses $\epsilon \in \{0.25, 0.5\}$, $\lambda_{\mathrm{pr}} \in \{0.25, 4\}$, $\delta \in \{10^{-2}, 10^{-8}, 10^{-32}\}$, and $\mu_{\mathrm{pr}} = 0.5$, $\tau = \sigma = 1$, giving $I = 0.5$. The two magnitude worlds are mirrored and each has 1,000 runs; paired inference uses their mean within seed. Policies are blind commitment, the forced threshold $\log \max\{2, (c_{\mathrm{hard}} - c_{\mathrm{easy}})I\ell/\lambda_{\mathrm{pr}}\}$, and the numerical Bellman policy. The downstream sign test is identical. For Monte Carlo cost reporting, its conditional expected length replaces a noisy realization, while every actually taken preview remains charged. This Rao–Blackwellization is not an uncharged observation available to the policy. The maximum change between horizon 128/spacing 0.025 and horizon 256/spacing

Table 10: Selected preview conditions. Blind and Bellman values are numerical expectations; the forced-heuristic column is a Monte Carlo mean with downstream costs Rao–Blackwellized. The Bellman policy explicitly skips at the first condition.

ϵ	δ	Price	Blind	Forced heuristic	Bellman value
0.5	0.01	4	6.84	11.82	6.84
0.25	0.01	4	21.00	20.87	17.70
0.25	10^{-32}	0.25	314.88	41.84	41.78

0.0125 is 4.89×10^{-4}. The maximum last-step Bellman residual is 8.7×10^{-8}, and independently increasing the sign-test quadrature order changes its value by at most 7.0×10^{-8}. These are empirical numerical checks over the parameter grid, not uniform error guarantees. In particular, a small horizon residual does not prove a general infinite-horizon bound.

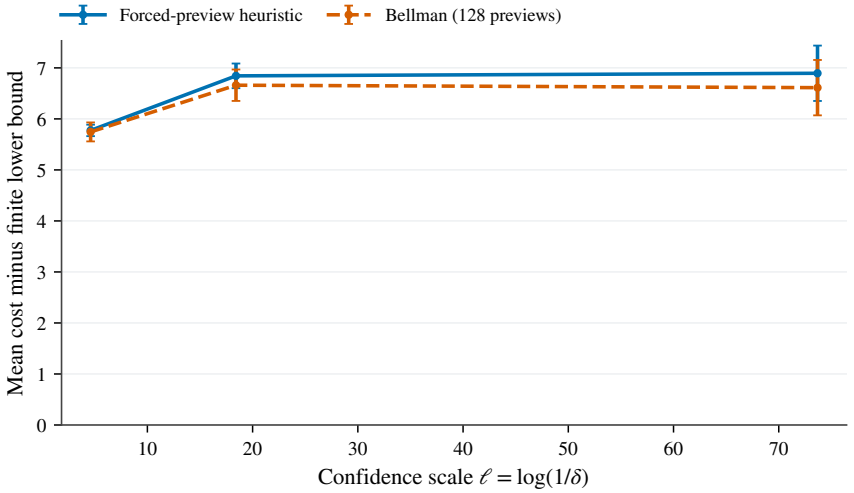

Figure 8: Excess total preview-policy cost above the universal finite-confidence lower bound, for $\epsilon = 0.25$ and $\lambda_{\mathrm{pr}} = 0.25$. Both axes are linear. The bound applies to a broader policy class than the fixed-downstream-test Bellman problem, so its nonzero gap is not solely numerical error.

The constant-gap and finite-horizon policies have different stopping contracts and are not conflated.

K.5 Population checks and code-derived sharing

The population experiment fixes $F = 64$, $v \in \{0, 1, 4, 12\}$, $m \in \{0, 8, 16, 32, 48, 56, 60, 63\}$, and requested unchecked release eight. It compares the clean-inspection minimax rule with the hypergeometric rule at $\alpha = 0.01$ and $\beta \in \{0.05, 0.2\}$. Consequently the respective total guarantees are β and $\alpha + \beta$, not numerically identical conditional guarantees. Counts and risk values are computed exactly; 1,000 sampled permutations per setting validate the implementation. A separate deterministic grid includes $F = 32$, 64, and 128 and several release sizes, and compares the exact frontier with the single-violation lower bound.

Fingerprint-only and partial-check controls. These controls use the same four fixed arithmetic tasks described in Appendix J, with 500 random orderings per task. They test the consequence of treating a fingerprint as permission for reuse, rather than the runtime of the order-matched hybrid. Outputs and gain are evaluated on the full 256-input task domain; gain is summed over the separate released function copies. Each checked member costs 256 candidate executions in addition to the common 768 initial executions and 256 cached reference answers.

Unchecked uniform release of eight fingerprint-matched members is harmful in approximately 81.8% of runs averaged over the four fixed tasks. That probability is not imposed by a Bernoulli sign-flip

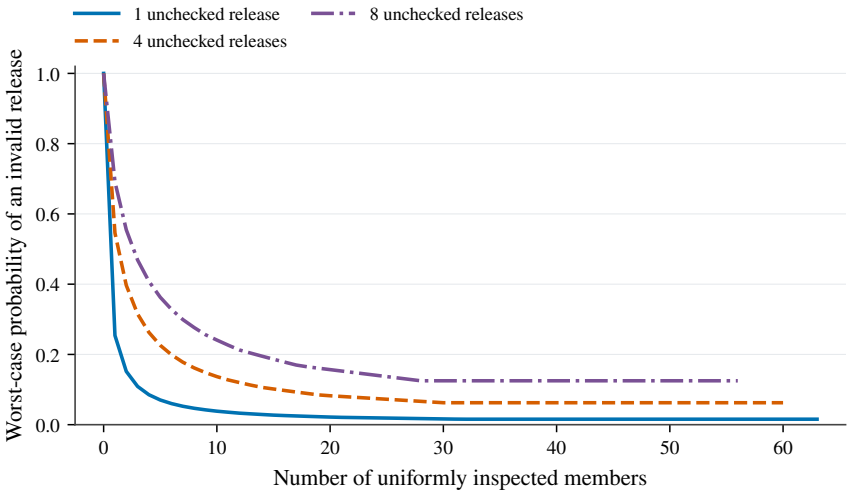

Figure 9: Exact worst-case joint risk for a clean-inspection gate with $F = 64$. Additional inspections reduce the worst case but cannot beat r/F when r unchecked members are released on the clean family. Uniform sampling is part of the contract.

parameter: the true harmful-member counts arise from executed boundary mutants. Eight partial member checks release a mean 5.08 edits with zero observed harmful releases, while 32 checks release eight at substantially greater execution cost. These results demonstrate a verification–release frontier, not a general predictor of real coding-agent performance.

K.6 Unknown-magnitude validation

The unknown-magnitude suite uses the parallel five-negative/one-positive instance, with true negative means -1, true positive mean 0.5, and unit noise. It compares the known-pair screen-and-certify method with the pilot-plus-composite-test policy on the *larger* unknown-mean hypothesis class, over 500 paired seeds at $\delta \in \{10^{-4}, 10^{-16}, 10^{-80}\}$. Their information constants are two and eight. At the smallest confidence, measured queries divided by ℓ are 2.25 and 9.82. The finite-confidence excess includes the $\ell^{2/3}$ pilot. No assertion is made that the implemented composite boundary has an optimal lower-order term.

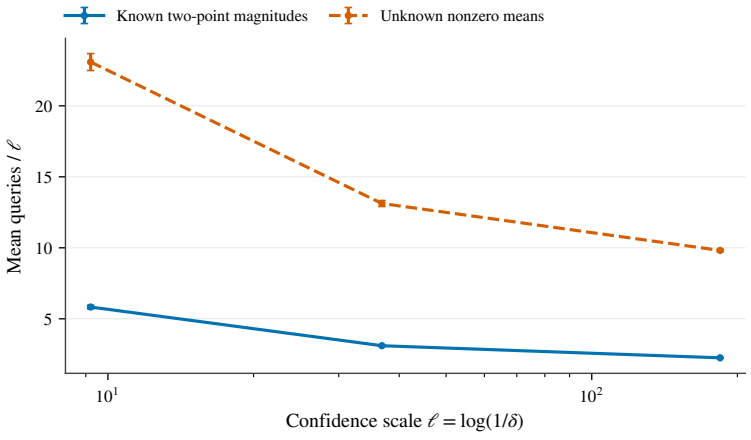

Figure 10: Known two-point magnitudes versus unknown nonzero means. The factor-four change is a hypothesis-class distinction proved in Proposition 8, not merely a penalty for estimating a parameter in the two-point test.