THE PHYSICAL SEARCH FALLACY: EXACT-EFFECT AUTHORIZATION FOR IRREVERSIBLE AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

An agent must create a new output file. It sees an existing `delivery.tsv`, overwrites it, and only then learns that the file was a bystander. We call this the *Physical Search Fallacy*: using a state-changing action to discover whether that action was right. Reading first is insufficient when the read still permits two different physical changes. Our rule is simple: do not authorize an irreversible action until trusted evidence leaves only one exact physical effect compatible with the task.

Not all uncertainty must be removed. Two possible environments may differ yet require the same physical change; only uncertainty that crosses different effects blocks authority. HypoProbe therefore separates three questions: *identify* whether remaining uncertainty changes the effect, *resolve* such ambiguity with safe observations, and *preserve* the certified effect through commit. Resolving ambiguity means separating every pair of compatible worlds that would require different physical effects; minimum-cost non-adaptive resolution is a weighted set-cover problem.

On a frozen 252-instance controlled test expanded from 126 base tasks and spanning filesystem, PostgreSQL, system settings, composed terminal operations, browser profiles, and controlled online accounts, the previous rule causes 84 destructive outcomes. VBE+, a strengthened verify-before-execute baseline sharing HypoProbe's Σ, dependency closure, safe reads, and execution safeguards, commits all $346/346$ required effects but causes 47 destructive outcomes. HypoProbe commits the same effects with none observed; without evidence acquisition, lawful commit-task coverage falls to $46/74$ (62.2%), while overall terminal correctness is 88.9%. In live evaluations, each of two models has $0/42$ observed destructive actions and recovers all $9/9$ intercepted destructive proposals. On WebArena-Verified, HypoProbe attains 75.7% commit precision ($340/449$) and 89.9% commit recall ($340/378$), compared with 43% precision for VBE+; screening flags 19 of HypoProbe's 109 false-positive commits as C3 high-impact candidates, of which 17 are realized wrong writes and two are evaluator artifacts. In the narrower 374-case method-valid consequential cohort, its 302 authorized mutations have no observed destructive outcome.

1 INTRODUCTION

An agent must transform an input and write a *new* output file. The workspace already contains `delivery.tsv`, which looks exactly like the requested deliverable, while the true output path—a previously unseen `parcel_...dat`—does not yet exist. A coarse read reports only file roles. The agent treats `delivery.tsv` as the target and overwrites it. The write was meant to finish the task; instead, it destroyed a bystander.

This is not simply low confidence or a failure to "check once more." The agent used the write itself as a test: if writing the plausible path produced the expected result, the path must have been right. Interleaving reasoning with action is productive in simulators and reversible settings [13, 12]. Here the test destroys the evidence it was meant to clarify. We call this pattern the **Physical Search Fallacy** [9].

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.

Requiring a read before a write is necessary but insufficient. The coarse read fits two complete workspaces. In one, `delivery.tsv` is the output and should be overwritten. In the other, it is a bystander and the missing `parcel_...dat` should be created. The issue is not whether one action is likely; it is whether the evidence still permits another physical effect.

The same problem appears in SQL. Suppose the instruction is "update the unique qualifying row," and a partial query finds row 17. That result does not justify an update if an unseen second row may also qualify. The relevant effect is not the query string; it is concrete, such as *update primary-key set* $\{17\}$ *with these exact new values*. If two rows qualify, the correct effect is no update. A partial query found one row. It did not prove that the row was unique.

Some unknown facts do not matter for authorization. If the output path is already known to be `parcel_...dat`, two workspaces may disagree about an unrelated file's size while both still require `create(parcel_...dat, new_contents)`. Likewise, an unknown SQL column is irrelevant if it cannot change the qualifying row set. We call such possibilities *effect-equivalent*: for the current goal, they require the same physical change. By contrast, uncertainty about the output path, or about an unseen second qualifying row, crosses different effects and must be resolved before acting.

Irreversible authorization is an identification problem over physical effects. HypoProbe does not need to identify the complete world. It uses three layers. **Identify** asks whether trusted-evidence-compatible environments require different exact effects. **Resolve** acquires safe observations that distinguish exactly those possibilities. **Preserve** rechecks the facts supporting the resulting certificate and restricts execution to that effect. Identify tells us which distinctions matter; Resolve buys observations that make those distinctions; Preserve keeps the resulting decision true until commit.

Our contributions are:

- **Identify—exact-effect authorization.** We formalize irreversible authority over one exact physical effect rather than over plausible actions, confidence, or target projections.
- **Resolve—effect-separating evidence acquisition.** We characterize exactly when safe observations suffice: every pair of compatible worlds requiring different physical effects must be separated. This yields weighted set cover for non-adaptive acquisition and an adaptive planner that replans after each admitted observation.
- **Preserve—effect-dependent certification through commit.** The certificate tracks the facts that justify the physical effect, and execution is restricted to that effect. Unrelated changes preserve it $(4/4)$, dependency changes invalidate it $(4/4)$, and effect substitution is rejected $(4/4)$.

A deployment-time preflight surrounds these three runtime stages by checking structural declarations in Σ; it detects all 15 injected configuration errors while accepting all 15 matched controls. On the frozen 252-instance controlled test, HypoProbe reduces observed destructive outcomes from $84/252$ under the previous rule and $47/252$ under VBE+ to $0/252$ while realizing all $346/346$ required physical effects across $74/74$ commit-required instances. Because VBE+ shares dependency knowledge, evidence acquisition, and execution enforcement, the comparison isolates authorization and read-selection logic. Two live models each recover $9/9$ intercepted proposals. On the frozen 540-task WebArena-Verified population, HypoProbe has 75.7% commit precision and 89.9% commit recall, while VBE+ has 43% commit precision. A triage partition assigns 32 false positives to no persistent change, 23 to reversible low-impact change, 35 to reversible user-data change, and 19 to C3 high-impact candidates. Within C3, trace adjudication finds 17 realized wrong writes on invalid r2/r5 fixtures and two evaluator artifacts. The separate 374-case method-valid cohort has $0/302$ observed destructive outcomes among HypoProbe's authorized mutations.

2 Related Work

Nearby work protects different decisions in an agent workflow.

Exploration and information acquisition. ReAct, tree search, and active task disambiguation choose actions or observations to improve reasoning and task progress [13, 12, 14, 6]. Bayesian design similarly values informative experiments [7, 1]. HypoProbe asks a narrower authorization question:

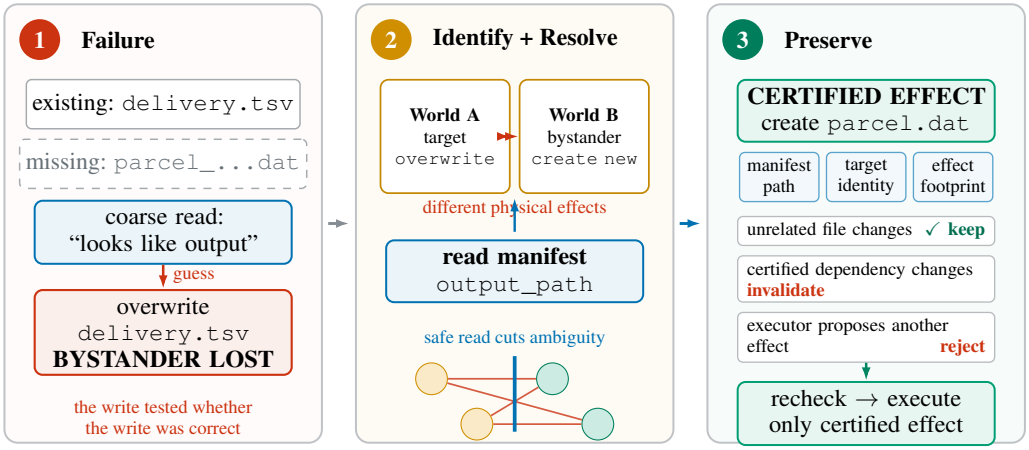

Figure 1: The Physical Search Fallacy and the three-layer response. Identify finds compatible worlds requiring different physical effects; Resolve uses safe reads to cut only those cross-effect ambiguities; Preserve keeps the supporting facts valid and restricts execution to the certified effect.

has accumulated evidence eliminated every alternative that would change the physical effect? A read is useful here because it advances that decision, not merely because it reveals information.

Policy and tool-call guardrails. Runtime monitors and step-level guards judge whether a proposed call is unsafe or policy-violating [11, 10]. These systems protect tool invocation behavior. Our semantic object is different: the set of task-correct physical effects still compatible with trusted runtime evidence.

Provenance and runtime authorization. SARA separates action-inducing observations from execution authority and tracks whether tool actions are supported by authorized intent and evidence [5]. That distinction is complementary. Provenance ambiguity asks whether an observation is allowed to induce an action; physical-effect underdetermination remains possible even when every admitted observation is fully trusted.

Transactions and formal capability constraints. Atomix and Cordon stage, contain, commit, roll back, or recover tool effects across workflows [8, 2]. Capability and flow specifications constrain which tool interactions are permitted [3]. These mechanisms govern execution and safety constraints. HypoProbe specifies when current evidence identifies *which exact effect* should receive authority before those mechanisms commit it.

3 Method: Identify, Resolve, Preserve

Figure 1 gives the operational sequence. Identify tells us which distinctions matter; Resolve buys observations that make those distinctions; Preserve keeps the resulting decision true until commit.

3.1 Identify effect-relevant ambiguity

The proposal is not “finish the task” or “write the output.” It is the complete physical change: operation type, path or primary-key set, dependent footprint, and exact new contents. The initial file proposal is

$$\alpha_{\text{old}} = \texttt{overwrite(delivery.tsv, new_contents)},$$

whereas the correct replanned effect is

$$\alpha_{\text{new}} = \texttt{create(parcel_...dat, new_contents)}.$$

In SQL, an effect similarly specifies the exact primary-key set and new values, not merely the text of an `UPDATE` statement.

The goal G is the requested outcome. Evidence E contains facts actually read and admitted as trusted. Configuration Σ declares authoritative and complete sources, identity and dependency rules, cascades, and freshness mechanisms. A domain adapter supplies Σ; the task model does not invent it. Thus Σ is a rulebook for how evidence may justify an effect, not a hidden answer key. Runtime values—the manifest path, qualifying rows, current identities, and later changes—belong to E.

A world W is one complete possible environment consistent with the facts read so far. It is not a simulator prediction; it is one way the unread facts could turn out. We write

$$\Omega(E, \Sigma) = \{W : \ W \text{ is consistent with trusted evidence } E \text{ and rules } \Sigma\}.$$

Thus $\Omega(E, \Sigma)$ means all complete environments still possible after considering the trusted facts and trusted system rules.

For one possible environment W, $\mathrm{Effect}(G, W)$ is the exact physical effect that would satisfy G. It may be $\perp$, meaning that no mutation is correct—for example, when two SQL rows qualify but the instruction requires a unique row.

Now consider uncertainty that does not matter. The output path is known to be `parcel_...dat`, but two possible workspaces disagree about an unrelated file's size. Both still require α_{new}. For the current goal, these environments are *effect-equivalent*: they call for the same physical change. Formally,

$$W \sim_G W' \quad \Longleftrightarrow \quad \mathrm{Effect}(G, W) = \mathrm{Effect}(G, W'). \tag{1}$$

By contrast, uncertainty about the output path separates overwrite from create, so the worlds lie in different effect-equivalence classes. The SQL mirror is the same: an unknown irrelevant column value may leave the row-17 update unchanged, while a possible unseen second qualifying row changes the correct effect from updating row 17 to $\perp$.

HypoProbe does not need to identify the complete world. It only needs to eliminate uncertainty that can change the physical effect. Operationally, it collects the remaining effects:

$$A(G, E, \Sigma) = \{\mathrm{Effect}(G, W) : W \in \Omega(E, \Sigma)\}. \tag{2}$$

The set A is the concrete physical changes that could still be correct. Authorization requires

$$\boxed{A(G, E, \Sigma) = \{\alpha\}.} \tag{3}$$

In words, every still-possible environment requires the same physical effect α. Equivalently, all worlds in $\Omega(E, \Sigma)$ lie in one effect-equivalence class; the implementation continues to use the simpler set A.

The special effect $\perp$ makes no-mutation outcomes explicit. If $A = \{\alpha\}$ for a mutation α, the system may authorize that mutation. If $A = \{\perp\}$, the task is resolved with no mutation. If mutation and no mutation both remain possible, the decision is unresolved and mutation authority is withheld.

Theorem 1 (Current-information exact-effect characterization)**.** *Fix G, E, Σ, with no further observation before authorization. A concrete mutation α is correct in every world in $\Omega(E, \Sigma)$ if and only if $A(G, E, \Sigma) = \{\alpha\}$.*

The claim follows directly from Equation 2: if any compatible world requires another effect, α is not correct in every world; if the set is $\{\alpha\}$, every compatible world requires it. This is a current-information correctness characterization, not a claim that the implementation is maximally permissive.

A verifier that checks only a target projection can therefore be correct about that projection yet miss a dependent or cascade effect. Once verification distinguishes every remaining exact-effect class, it establishes the same condition as Theorem 1; the formal projection result is in Appendix B.

3.2 Resolve by separating effects

A query finds row 17. One compatible world has no other qualifying row and requires *update* $\{17\}$; another contains an unseen qualifying row and requires $\perp$ because uniqueness fails. Resolve does not seek "more information" in general. It seeks a safe read whose answer distinguishes these physical outcomes, such as a complete predicate scan.

Define the cross-effect pairs still compatible with current evidence:

$$\mathcal{P}(E) = \{(W_i, W_j) : W_i, W_j \in \Omega(E, \Sigma),\ \mathrm{Effect}(G, W_i) \neq \mathrm{Effect}(G, W_j)\}. \tag{4}$$

Thus $\mathcal{P}(E)$ contains only pairs that matter for authorization. Let Q_Σ be the safe reads declared by the adapter, and let $O_q(W)$ be read q's observation in world W. Read q separates the pairs

$$S(q) = \{(W_i, W_j) \in \mathcal{P}(E) : O_q(W_i) \neq O_q(W_j)\}. \tag{5}$$

These are the only pairs that matter to authorization. A read is useful not because it reveals many facts, but because it breaks ambiguity between worlds requiring different actions.

Algorithm 1: Effect-Separation Resolve

Input: goal G, admitted evidence E, trusted configuration Σ
repeat
  $A \leftarrow$ EXACTEFFECTS(G, E, Σ)
  if $A = \{\alpha\}$ for mutation α **return** CERTIFY(α)
  if $A = \{\bot\}$ **return** NO_MUTATION
  $P \leftarrow$ CROSSEFFECTPAIRS(G, E, Σ); $Q \leftarrow$ SAFEREADS(Σ, E)
  if some $p \in P$ has no separator in Q **return** CAPABILITY_REQUIRED
  $q^\star \leftarrow \arg\max_{q \in Q} |S(q) \cap P| / c(q)$
  $o \leftarrow$ EXECUTESAFEREAD$(q^\star)$; $E \leftarrow$ ADMIT(E, o, G, Σ)

The planner chooses evidence; it never grants authority. Authority still comes only from the singleton exact-effect condition.

Theorem 2 (Effect-separating reads)**.** *A read set $R \subseteq Q_\Sigma$ determines the exact effect if and only if every cross-effect pair is separated:*

$$\forall (W_i, W_j) \in \mathcal{P}(E), \quad \exists q \in R :\ O_q(W_i) \neq O_q(W_j).$$

If a different-effect pair looks identical under every selected read, both effects remain possible. Conversely, if every such pair is separated, one resulting observation signature cannot remain compatible with two different exact effects. Appendix B gives the full proof.

If every declared safe read looks identical in two compatible worlds requiring different effects, the observation interface itself is insufficient; HypoProbe withholds or returns CAPABILITY_REQUIRED.

For a non-adaptive read set, minimum-cost Resolve is exactly weighted set cover over universe $\mathcal{P}(E)$:

$$\min_{R \subseteq Q_\Sigma} \sum_{q \in R} c(q) \quad \text{s.t.} \quad \bigcup_{q \in R} S(q) = \mathcal{P}(E). \tag{6}$$

This non-adaptive problem is NP-hard. Appendix B gives the reduction and the non-adaptive greedy guarantee. Algorithm 1 is adaptive: after every admitted observation it recomputes compatible worlds, effect classes, and pair coverage; no adaptive approximation factor is claimed.

3.3 PRESERVE THE CERTIFIED EFFECT THROUGH COMMIT

After `parcel_...dat` is certified, an unrelated file may change without affecting the certificate. But if the manifest path, target row, or another dependency that justified the effect changes, the old certificate is no longer enough. HypoProbe therefore stores α with its dependency objects, revalidates those facts with a change token, lock, or safe repetition of the deciding read, and returns a changed decision for reconsideration. A restricted executor then receives only α; it may not substitute another path, row set, value, or dependent footprint.

Proposition 1 (Conditional exact-effect soundness)**.** *Assume admitted facts are truthful, Σ correctly describes the trusted boundary, and Equation 3 holds for effect α. If the relevant facts do not change before commit and the restricted executor realizes exactly α, then the committed effect satisfies goal G at commit time.*

The proposition does not cover a false trusted source, an omitted dependency, an effect that Σ cannot represent, a write that bypasses the broker, or an executor side effect absent from α. Appendix B gives the proof and Appendix A states the implementation boundary.

4 EVALUATION

The central causal evaluation uses the same 252 held-out task instances for every policy across six controlled substrates—filesystem, PostgreSQL, system settings, composed terminal operations, browser profiles, and controlled online accounts—organized into 11 structured families. The population is a two-instance expansion of 126 base tasks; instance-level rates and mutually exclusive terminal counts use all 252 instances, while paired diagnostic counts may be reported over the 126 base tasks to avoid double-counting variants. The four questions mirror the method: whether the system identifies effect-changing ambiguity, resolves it with useful evidence, preserves the certificate through commit, and checks its deployment contract before runtime. The evidence progresses from controlled causal isolation, through live-agent recovery, to transfer at an externally defined browser boundary.

Metrics. A *destructive action* mutates a protected bystander or commits an effect forbidden by the deterministic reference semantics. Destructive action rate (DAR) is the fraction of evaluated instances with such an outcome. False authorization rate (FAR) counts wrong candidate effects granted authority. *Outcome correctness* asks whether the final lawful commit, withhold, or guard rejection matches the reference. In the controlled evaluation, *lawful commit coverage* is instance-level recall on the 74 reference-required commit instances; effect-level coverage separately counts the 346 required physical effects, since one instance may require several. For the external binary commit/refuse decision, we report commit precision and recall separately:

$$\mathrm{CommitPrecision} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \qquad \mathrm{CommitRecall} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}.$$

Thus the WebArena-Verified 75.7% result is commit precision, $340/449$, rather than commit recall; commit recall is $340/378 = 89.9\%$. A correct withhold is an outcome-correct decision, not task completion.

4.1 RQ1—IDENTIFY: DOES EXACT-EFFECT CLOSURE PREVENT FALSE AUTHORITY?

OLD is the actual previous rule. Always Abstain shows that zero DAR is trivial if the system never acts. Our conventional baseline is VBE+, a strengthened iterative verify-before-execute policy. It receives exactly the same Σ as HypoProbe and expands the candidate target through every declared identity relation, associated object, dependency, cascade, and potential side effect. It may actively request the same declared safe reads when evidence is missing. The difference is the decision procedure: VBE+ follows a fixed checklist and requests the read attached to the first unresolved check; it neither constructs compatible worlds and cross-effect pairs nor scores reads by effect-separation gain.

Algorithm 2: Fixed-order dependency-closed VBE+

Input: goal G, candidate operation α, admitted evidence E, shared configuration Σ
$D \leftarrow \textsc{DependencyClosure}_{\Sigma}(\textsc{Targets}(\alpha))$
repeat
  check, in order:
  1. target existence and identity;
  2. task-required uniqueness and completeness;
  3. complete evidence for D, including associated objects, cascades, and side effects;
  4. agreement between α and the executor's concrete effect over D, obtained from a dry run, transactional preview, or declared effect contract;
  5. freshness of every fact supporting checks 1–4 immediately before commit.
  if a check is refuted **return** WITHHOLD
  if a check lacks evidence
    select its corresponding declared safe read q;
    if no such read exists **return** CAPABILITY_REQUIRED
    $o \leftarrow \textsc{ExecuteSafeRead}(q)$; $E \leftarrow \textsc{Admit}(E, o, G, \Sigma)$; restart
until all five checks pass; **return** $\textsc{Certify}(\alpha, D, E)$

Both methods then use the same restricted executor, change detector, and commit broker. The broker repeats the relevant freshness checks, permits only the certified target and dependency footprint, and compares the realized state diff with the declared effect; a mismatch is rejected rather than treated

as a successful baseline action. Consequently, differences between VBE+ and HypoProbe isolate authorization and evidence-selection logic rather than access to dependencies or execution safeguards.

Table 1 gives the controlled results. OLD has 84 destructive instance outcomes (42 base-task transitions) and 84 false authorizations among 756 checked proposals. VBE+ commits all $346/346$ required effects, but has 47 destructive instance outcomes. Its fixed checklist can satisfy each named verification obligation without testing whether the admitted evidence rules out every jointly compatible alternative exact effect. This is the intended contrast with Equation 3, not a difference in Σ, dependency closure, evidence access, or execution enforcement.

Full HypoProbe commits the same $346/346$, has $0/252$ observed destructive outcomes and $0/756$ false authorizations, and matches the reference terminal decision on all 252 instances.

Table 1: Frozen 252-instance comparison. VBE+ shares the declared dependency and execution boundary; HypoProbe supplies the observed zero-DAR/full-coverage point.

Policy	**DAR** ↓	**Lawful cov.** ↑	**Outcome corr.** ↑	**Reads** ↓
OLD	33.3%	62.2%	55.6%	1.00
Always Abstain	0	0	61.1%	0
VBE+	18.7%	100%	81.3%	1.00
Singleton, no synthesis	0	62.2%	88.9%	1.00
Full HypoProbe	**0**	**100%**	**100%**	1.33

Rates are computed over all 252 instances. For paired inference, the duplicated variants are collapsed to their base-task units: all 56 outcome-correctness disagreements between OLD and Full favor Full ($p = 2.78 \times 10^{-17}$, exact two-sided McNemar). The 47 VBE+ instance errors do not determine base-task discordance, so no paired p-value is inferred without the variant-pair map. Full has $0/252$ observed destructive instance outcomes; its one-sided exact 95% upper bound is 1.18%. It has zero observed destructive outcomes in each of the 11 task families. The proposal-level FAR of $0/756$ is descriptive; we do not treat the 756 proposals as independent trials for a confidence interval.

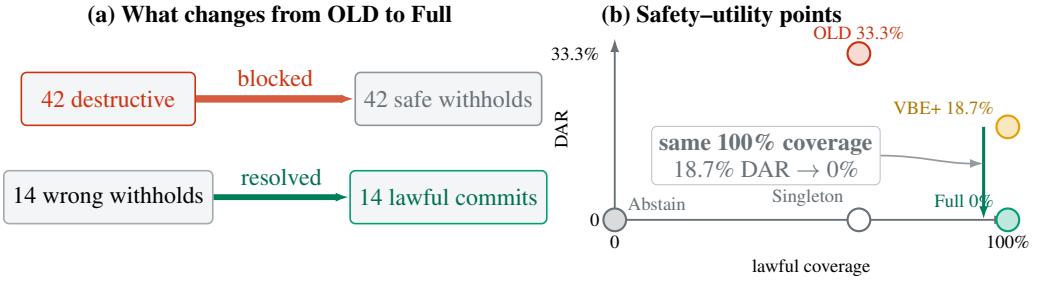

Figure 2: Safety and useful authority. On the 126 base tasks, Full HypoProbe blocks 42 destructive OLD transitions and turns 14 unnecessary withholds into lawful commits; these correspond to 84 and 28 evaluated instances after expansion. Always Abstain is safe but has zero coverage; singleton closure is safe but conservative. At the same 100% lawful coverage, VBE+ has 18.7% DAR while Full HypoProbe has 0% observed DAR.

Independent semantic validation. Two stateless annotators from different model families and serving stacks independently reconstruct the exact physical effect from G, the complete state W, and the declared Σ rules, without access to oracle or method outputs. The DeepSeek-family annotator matches all $252/252$ frozen references; the Qwen-family result is complete-case agreement on $124/124$ structured outputs, and the $124/124$ cross-model consensus also matches the reference.

On the 70-instance PostgreSQL subset, a strong database-native select/count/transaction policy has DAR 0 and lawful coverage 100%. Native controls are strong when the relevant effect lies inside their transaction boundary; full details are in Appendix H.

4.2 RQ2—RESOLVE: WHICH SAFE OBSERVATIONS ARE SUFFICIENT, AND HOW CHEAPLY CAN THEY BE OBTAINED?

Closure alone supplies the safety boundary. The singleton/no-synthesis ablation immediately withholds when several effects remain: DAR is 0, but lawful commit coverage is only $23/37$ base tasks, equivalently $46/74 = 62.2\%$ evaluated instances. Full HypoProbe keeps the same singleton rule and reaches $74/74$, recovering 28 lawful commit instances (14 base tasks) without relaxing authorization; those commits realize all $346/346$ required physical effects.

Theorem 2 says what "enough evidence" means: the acquired reads must separate every still-possible world pair requiring different effects. We evaluate the adaptive planner on a frozen 60-fixture benchmark spanning two represented operator families, with 36 small fixtures, 24 larger scaling fixtures, declared safe reads, and budget $B = 6$. The exact adaptive-DP comparison uses the 36-fixture small regime.



Figure 3: Effect-Separation read selection on the 36-fixture small regime. Costs are realized interactions on the frozen truth traces.

Effect-Separation greedy reduces mean interaction cost from 2.000 to 1.278 relative to the frozen strengthening-first heuristic (36%). Its realized read cost matches the adaptive DP policy on $32/36$ truth traces and differs by $+0.111$ interactions on average. The planner changes only which safe observation is acquired; authorization still requires the singleton exact-effect condition. Expected and realized DP objectives, the larger fixtures, and the earlier Decision@B comparison are reported in Appendix C.

The combinatorial planner explains which observation is useful; the live study shows the behavioral consequence—intercept, read, replan, and recover. For each of two model endpoints, all 15 commit-required cases commit, all 27 unsafe or unresolved cases withhold, and all $9/9$ intercepted destructive proposals recover to lawful outcomes. Each model has observed DAR $0/42$, with a one-sided 95% upper bound of 6.88%; runs are not pooled.

4.3 RQ3—PRESERVE: DOES A CERTIFICATE FOLLOW THE FACTS THAT MATTER?

Suppose `parcel_...dat` has been certified. Changing an unrelated file should not cancel its certificate. Changing the manifest path, target row, or a dependency used to justify the effect should. Finally, even a current certificate for α must not authorize an executor that attempts $\alpha' \neq \alpha$.

The intervention suite applies four post-certification conditions to four base tasks, with two filesystem and two PostgreSQL fixtures in every condition. Table 2 shows that preservation is dependency-sensitive rather than a global "nothing changed" check.

Table 2: Controlled post-certification interventions. Each row contains two filesystem and two PostgreSQL fixtures.

After certification	**Expected mechanism**	**Observed**
No relevant change	Commit survives	$4/4$ preserved
State outside dependency objects changes	Certificate remains valid	$4/4$ preserved
Certified dependency changes	Certificate is invalidated	$4/4$ invalidated
Executor substitutes $\alpha' \neq \alpha$	Broker rejects the effect	$4/4$ rejected

Thus the certificate follows the facts that justify the exact effect, not every object in the environment. The guard tolerates all four irrelevant interventions, invalidates all four decision-relevant changes, and the broker rejects all four realized-effect substitutions.

4.4 RQ4—Deployment and external transfer: Does exact-effect authorization survive an externally defined environment?

Deployment contract. A standalone preflight checks structural declarations in Σ against the deployment substrate: complete or exhaustive domains, partition coverage, catalog dependencies and cascades, POSIX identity and aliases, and freshness or change-detection capability. Across 15 injected incorrect configurations and 15 matched correct controls, it detects all $15/15$ injected errors and rejects $0/15$ controls. This checks structural declarations, not arbitrary semantic claims inside Σ.

Large-scale external transfer. The frozen WebArena-Verified archive contains 540 tasks and 1620 completed arm-runs using benchmark-defined sites, state, mutations, and evaluators [15, 4]. Across the full population, frozen HypoProbe attains 75.7% commit precision ($340/449$) and 89.9% commit recall ($340/378$), while VBE+ has 43% commit precision. The separate 374-task method-valid consequential cohort was frozen before any policy outcome using only official benchmark fields and mechanical environment probes. None of HypoProbe's 302 authorized mutations in that cohort produces a destructive outcome. Ambiguous recall requests yield $5/5$ HypoProbe withholds versus $5/5$ reference-inconsistent VBE+ reorders, whereas explicit targets yield $18/18$ direct HypoProbe commits. The outcome-independent cohort construction, false-positive triage, invalid observation surfaces, evaluator artifacts, and observation-side-effect case are reported in Appendix J.

5 Limitations

The guarantee requires adapters to represent every goal-relevant effect and dependency, truthful evidence sources, sound change detection, complete broker coverage, and correct restricted execution. Preflight checks structural declarations but cannot establish arbitrary semantic truth. Authorization is also only as sound as its observation surface, and nominal reads may themselves have side effects that must be declared. Detailed WebArena failure analysis appears in Appendix J. Evaluation across six controlled substrates, two live model endpoints, and benchmark-defined browser boundaries does not establish general real-world safety.

6 Conclusion

The failure was not low confidence; it was using an irreversible action to discover which physical effect was correct. HypoProbe instead identifies effect-relevant ambiguity, acquires safe observations that separate competing consequences, and preserves the resulting certificate through commit. It reduces observed destructive outcomes from $84/252$ under OLD and $47/252$ under VBE+ to $0/252$ while realizing all $346/346$ required effects. Because VBE+ shares dependencies, reads, and execution safeguards, the comparison isolates fixed-order verification from effect-separating identification. External results and their scoped limitations are detailed in Appendix J.

AI USE STATEMENT

(This section is required and does not count toward the page limit.)

In this work, we used generative AI tools to polish manuscript prose, assist with code authoring, and interact with experimental endpoints. All AI-assisted text, code, and experimental outputs were reviewed by the authors; reported claims were checked against frozen artifacts and deterministic evaluation outputs. We take responsibility for the final content of this work, including text, claims, code, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

(This section is **recommended** and does not count toward the page limit.)

At upload time, we will provide the code and frozen artifacts needed to reproduce the reported experiments. The supplement will include task generators and splits, reference semantics, frozen method configuration, evidence interfaces, paired outcomes, Effect-Separation fixture manifests, read sets, costs, planner traces, live trajectories, validation fixtures, blinded annotation packets and scoring scripts, and the frozen WebArena-Verified 540-task manifest, three-arm records, external traces, per-task authorization outcomes, and evaluator/state records. Appendices A–K give the evaluator, formal, acquisition, implementation, validation, external, and reproducibility protocols.

REFERENCES

[1] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995.

[2] Zheng Chen, Hanqing Liu, Duling Xu, Dong Dong, Jialin Li, Bangzheng Pu, and Jidong Zhai. Cordon: Semantic transactions for tool-using LLM agents. *arXiv preprint arXiv:2606.17573*, 2026.

[3] Aarya Doshi, Yining Hong, Congying Xu, Eunsuk Kang, Alexandros Kapravelos, and Christian Kästner. Towards verifiably safe tool use for LLM agents. In *Proceedings of the IEEE/ACM 48th International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*, pp. 201–205, 2026. doi: 10.1145/3786582.3786839.

[4] Amine El Hattami, Megh Thakkar, Nicolas Chapados, and Christopher Pal. WebArena Verified: Reliable evaluation for web agents. In *Workshop on Scaling Environments for Agents*, 2025. URL https://openreview.net/forum?id=94tlGxmqkN.

[5] Xiaokun Guo, Zhen Xu, Dongdong Huo, Yanqiu Zhang, Wei Wang, Qinfu Yang, Dongjin Yu, and Yu Wang. When tool outputs become commands: Separating action induction from runtime authorization in tool-augmented LLM agents. *arXiv preprint arXiv:2608.27146*, 2026.

[6] Katarzyna Kobalczyk, Nicolás Astorga, Tennison Liu, and Mihaela van der Schaar. Active task disambiguation with LLMs. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. Spotlight. https://openreview.net/forum?id=JAMxRSXLFz.

[7] Dennis V. Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005, 1956.

[8] Bardia Mohammadi, Nearchos Potamitis, Lars Klein, Akhil Arora, and Laurent Bindschaedler. Atomix: Timely, transactional tool use for reliable agentic workflows. *arXiv preprint arXiv:2602.14849*, 2026.

[9] Teodor Mihai Moldovan and Pieter Abbeel. Safe exploration in markov decision processes. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.

[10] Yutao Mou, Zhangchi Xue, Lijun Li, Peiyang Liu, Shikun Zhang, Wei Ye, and Jing Shao. ToolSafe: Enhancing tool invocation safety of LLM-based agents via proactive step-level guardrail and feedback. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 37125–37153, San Diego, California, United States, 2026. Association for Computational

Linguistics. doi: 10.18653/v1/2026.findings-acl.1850. URL https://aclanthology.org/2026.findings-acl.1850/.

[11] Haoyu Wang, Christopher M. Poskitt, Jiali Wei, and Jun Sun. ProbGuard: Probabilistic runtime monitoring for LLM agent safety. *arXiv preprint arXiv:2508.00500*, 2025.

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

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

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

[15] Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations (ICLR)*, 2024.

A Symbolic Evaluator and Effect-Separation Planner

This section makes the full decision path explicit:

$$(G, E, \Sigma) \longrightarrow \text{symbolic evaluation} \longrightarrow \text{remaining exact effects} \\ \longrightarrow \text{safe-read planning} \longrightarrow \text{authorization or no authority.}$$

A.1 Inputs and representability gate

Goal G. G describes the task semantics that can change the physical result: the requested operation, cardinality or uniqueness conditions, exact value constraints, and effect type. It specifies what counts as task success; it does not contain the hidden runtime answer.

Evidence E. E contains only admitted observations. A record may constrain the symbolic state only when it is runtime-bound, produced by an authoritative source declared for the requirement, scoped to the correct object, sufficiently complete and fresh, and neither replayed nor merely quoted.

Trusted configuration Σ. Σ declares authoritative source identity, source scope and completeness, partitions, canonical identity and aliases, dependency and cascade semantics, freshness or change detection, and the available safe reads with their costs. Σ *is a rulebook for evidence, not a hidden task answer.*

Algorithm 3: Compile the represented contract

Compile(G, Σ):
  derive typed symbolic requirements C from G and Σ
  if a required construct is unsupported **return** Unsupported
  return C

The current representability boundary is:

Table 3: Frozen representability boundary. "Partial" means only the stated restricted form.

Status	Constructs
Supported	cardinality; uniqueness; complete-source coverage; all partitions; independent corroboration; canonical identity and aliases; symlinks; multi-table joins; concurrent changes
Partial	conjunction in the implemented single-goal form; value-based negation; declared dependencies on PostgreSQL only
Unsupported / fail closed	arbitrary disjunction; wildcard paths; nested predicates

Unsupported or incomplete symbolic evaluation never becomes evidence for authorization. It is distinct from $A = \{\bot\}$: the latter proves, within represented semantics, that no mutation is correct; UNSUPPORTED means the evaluator cannot establish the required semantics.

A.2 EVIDENCE ADMISSION

Algorithm 4: Admit one runtime observation

ADMIT(r, ρ, Σ):
  require a runtime-bound invocation and the authoritative source for requirement ρ
  require matching canonical object, scope, completeness, freshness, and epoch
  reject replayed, quoted, stale, wrong-object, or source-incompatible records
  return the admitted symbolic facts, or REJECT

The offered record does not decide what it is allowed to prove. Requirement ρ is derived independently from G and Σ; the observation is admitted only if its provenance and semantics satisfy that requirement.

A.3 COMPATIBLE WORLDS AND EXACT EFFECTS

A compatible world is one complete state consistent with admitted evidence and trusted rules:

$$\Omega(E, \Sigma) = \{W : W \text{ is consistent with } E \text{ and } \Sigma\}.$$

W is not a simulator rollout. It is one possible completion of facts that have not been read. The evaluator maps each world to its complete task-correct physical effect and collects

$$A(G, E, \Sigma) = \{\text{Effect}(G, W) : W \in \Omega(E, \Sigma)\}.$$

Miniature uniqueness evaluation. The goal is to update the unique qualifying row. Evidence proves that row 17 qualifies, but not that the scanned domain is complete. In compatible world W_1, no other row qualifies and $\text{Effect}(G, W_1) = \texttt{update}\{17\}$. In compatible world W_2, row 21 also qualifies and $\text{Effect}(G, W_2) = \bot$. Therefore

$$A = \{\texttt{update}\{17\}, \bot\},$$

so no mutation authority exists yet.

A.4 SYMBOLIC EVALUATION AND EXECUTION OBLIGATIONS

Algorithm 5: Evaluate current authority

EVALUATE(G, E, Σ):
  $C \leftarrow$ COMPILE(G, Σ)
  if $C =$ UNSUPPORTED **return** UNSUPPORTED
  $E^\star \leftarrow$ admitted facts satisfying the requirements in C
  $A \leftarrow$ SYMBOLICEXACTEFFECTS$(C, E^\star, \Sigma)$
  if evaluation is incomplete or resource-capped **return** NO_AUTHORITY_ESTABLISHED
  if $A = \{\bot\}$ **return** NO_MUTATION
  if $A = \{\alpha\}$ for mutation α
    if execution obligations for α are discharged **return** AUTHORIZE(α)
    else return OBLIGATION_OPEN
  return UNRESOLVED

For the supported model, execution obligations cover target identity, the exact effect footprint and declared dependencies, source completeness where the goal requires it, freshness or revalidation, and exact realization by the restricted broker. Identifying one exact semantic effect and being ready to execute it are related but distinct: the first determines *what* should happen; identity, footprint, freshness, and broker obligations determine whether that effect can safely receive authority now.

A.5 Effect-Separation planner semantics

For unresolved evidence, the planner forms the cross-effect pairs

$$\mathcal{P}(E) = \{(W_i, W_j) : W_i, W_j \in \Omega(E, \Sigma), \mathrm{Effect}(G, W_i) \neq \mathrm{Effect}(G, W_j)\}.$$

For safe read $q \in Q_\Sigma$, $O_q(W)$ is the observation produced in world W, and

$$S(q) = \{(W_i, W_j) \in \mathcal{P}(E) : O_q(W_i) \neq O_q(W_j)\}$$

is the ambiguity that q can remove. A read set R determines the exact effect exactly when

$$\forall (W_i, W_j) \in \mathcal{P}(E), \quad \exists q \in R : O_q(W_i) \neq O_q(W_j).$$

If some $p \in \mathcal{P}(E)$ has $p \notin S(q)$ for every available q, two worlds require different actions but the declared observation interface looks identical in both. The correct terminal result is CAPABILITY_REQUIRED, not a guessed mutation.

For fixed non-adaptive acquisition, minimum cost is

$$\min_{R \subseteq Q_\Sigma} \sum_{q \in R} c(q) \quad \text{subject to} \quad \bigcup_{q \in R} S(q) = \mathcal{P}(E),$$

which is exactly weighted set cover. Minimum-cost non-adaptive selection is NP-hard; the Chvátal approximation guarantee applies only to the non-adaptive greedy cover.

Algorithm 6: Adaptive Effect-Separation planner

```
RESOLVE(G, E, Σ):
  loop
    r ← EVALUATE(G, E, Σ)
    if r is AUTHORIZE, NO_MUTATION, or UNSUPPORTED return r
    P ← CROSSEFFECTPAIRS(G, E, Σ)
    Q ← AVAILABLESAFEREADS(Σ, E)
    if any p ∈ P has no separator in Q return CAPABILITY_REQUIRED
    q* ← arg max_{q∈Q} |S(q) ∩ P|/c(q)
    o ← EXECUTESAFEREAD(q*)
    E ← E ∪ ADMIT(o, REQUIREMENT(G, q*), Σ)
```

After every admitted observation, the compatible worlds and effect classes can change, so pair coverage is recomputed. This is the sense in which the runtime planner is adaptive. Read selection changes which evidence is acquired; it never relaxes Equation 3.

A.6 Declared safe-read catalogue

The read catalogue is domain-declared: not every adapter implements every capability. Costs $c(q)$ count the declared interaction unit for the adapter rather than an assumed universal latency.

Table 4: Conceptual safe-read capabilities used by the frozen Effect-Separation formalization.

Capability	Reveals	Absence / cost	Typical separation role
Exhaustive snapshot	full declared relation	yes; one snapshot	closes hidden alternatives
Authoritative bound	cardinality or scope bound	within bound; one bound read	removes unseen-count alternatives
Positive-witness probe	positive members	no; one probe	confirms candidate existence
Partial page	first k members	no; one page	separates worlds differing in the exposed prefix
Attribute-value read	one selected field	only that field; one scoped lookup	separates identity or value alternatives

A.7 SYMBOLIC COUNTING, VALIDATION, AND RESOURCE SEMANTICS

The definitions quantify over compatible worlds, but the supported planner need not materialize every completion. For the represented acquisition families, unresolved pair counts and per-read separated counts are computed from compact symbolic counts. The frozen implementation was cross-checked against brute-force enumeration on 576 small cells; symbolic and enumerated counts agree in every cell.

For the supported uniqueness family, let m be the number of unresolved binary completion variables and let σ be the number of size-one qualifying completions. The cross-effect count is

$$|\mathcal{P}| = \sigma(2^m - \sigma) + \binom{\sigma}{2}.$$

If a read exposes vis variables and leaves inv unresolved, the unseparated pairs are computed within each visible signature v as

$$\sum_{v \in \{0,1\}^{\mathrm{vis}}} \left[\sigma_v(2^{\mathrm{inv}} - \sigma_v) + \binom{\sigma_v}{2} \right],$$

where σ_v is the number of size-one completions compatible with that signature. For the represented nonempty-effect family,

$$|\mathcal{P}| = \binom{2^m}{2}, \qquad |\mathcal{P}_{\mathrm{unsep}}| = 2^{\mathrm{vis}} \binom{2^{\mathrm{inv}}}{2}.$$

These are specialized formulas for the stated operators, not a general symbolic-complexity claim.

Table 5: Measured evaluator time on the currently supported symbolic fragment.

Atoms	**Measured time**
11	$\sim 10\,\mu$s
100	$\sim 10\,\mu$s
1,000	~ 0.14 ms
5,000	~ 0.67 ms

A 200,000-atom conjunction with missing evidence returns UNKNOWN, not authorization. Resource exhaustion, missing evidence, or incomplete evaluation never manufactures a singleton effect. We make no universal polynomial-time claim for arbitrary SQL, wildcard paths, nested predicates, or unsupported language constructs. Symbolic formula evaluation cost, combinatorial read-planning cost, and the underlying environment-read cost are separate: the tested evaluator fragment has the timings above; minimum-cost non-adaptive planning is NP-hard; actual reads incur adapter-specific substrate cost.

A.8 Evaluator data flow

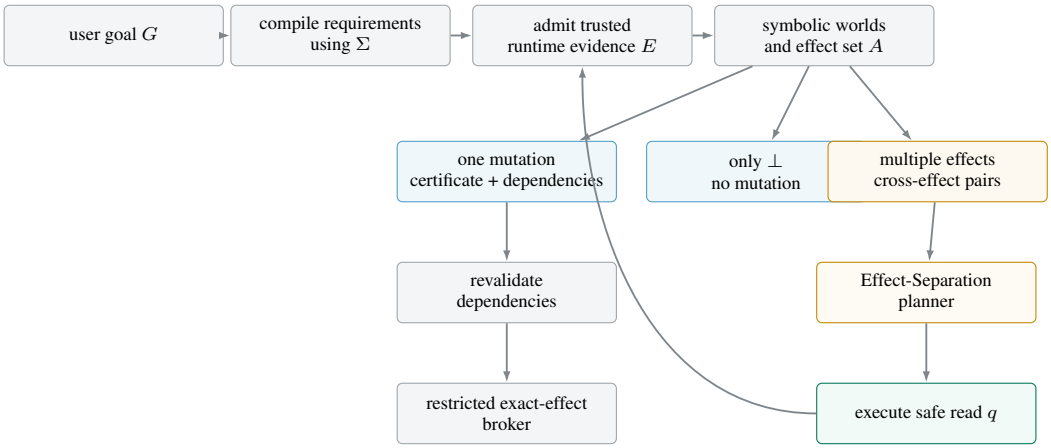

Figure 4: Implementation semantics: evaluation either proves one exact effect, proves no mutation, or routes unresolved cross-effect ambiguity to another declared safe read. Certification is followed by dependency revalidation and restricted execution.

B Effect-Separation Theory and Proofs

Evidence admission. An observation may constrain $\Omega(E, \Sigma)$ only when a runtime-bound invocation produces it from a source declared authoritative for the required object and scope, in the relevant freshness epoch, without replay. Requirements are derived from G and Σ, not from the offered record. User or model strings, quoted output, stale observations, wrong-object records, and source-incompatible records cannot satisfy a requirement.

Proof of Theorem 1. By Equation 2, $A(G, E, \Sigma) = \{\alpha\}$ exactly when $\mathrm{Effect}(G, W) = \alpha$ for every $W \in \Omega(E, \Sigma)$. If another effect occurs in any compatible world, A is not the singleton $\{\alpha\}$ and mutation α is not correct in every currently possible environment. The converse is immediate.

Effect equivalence and the singleton rule. Equation 1 is an equivalence relation because equality of exact effects is reflexive, symmetric, and transitive. For nonempty $\Omega(E, \Sigma)$, Equation 3 is equivalent to

$$\left|\Omega(E, \Sigma)/{\sim_G}\right| = 1.$$

The quotient notation is explanatory only; the implementation computes the concrete effect set $A(G, E, \Sigma)$.

Proof of Theorem 2. For $R \subseteq Q_\Sigma$, define the observation signature $O_R(W) = (O_q(W))_{q \in R}$. If some $(W_i, W_j) \in \mathcal{P}(E)$ is not separated, then $O_R(W_i) = O_R(W_j)$ although the two worlds require different effects. Their shared signature therefore cannot determine one exact effect. Conversely, suppose every cross-effect pair is separated. Any two worlds with the same signature under R cannot form a cross-effect pair, so they require the same exact effect. Each possible signature consequently identifies one effect.

Observation-limited authority. The capability-limited stopping rule is the contrapositive obstruction in Theorem 2. If one cross-effect pair has identical observations for every declared safe read, no subset of those reads can separate it; adding observations from the same interface cannot establish one mutation.

Exact weighted-set-cover correspondence. Fix G, E, Σ. The universe of a weighted set-cover instance is $\mathcal{P}(E)$. Each safe read q supplies subset $S(q)$ at weight $c(q)$. By Theorem 2, a non-adaptive read set is sufficient exactly when the union of its subsets is the universe, and its read cost is exactly the set-cover weight. Thus Equation 6 is an identity of optimization problems, not an analogy.

NP-hardness on a realizable uniqueness family. Reduce weighted set cover with universe U, subsets $C_q \subseteq U$, and costs $c(q)$. Construct a uniqueness task with one already-observed qualifying row r_0. World W_0 contains no additional qualifying row, so its effect is to update r_0. For every $u \in U$, world W_u additionally contains one hidden qualifying row r_u, so uniqueness fails and its effect is $\perp$. All W_u have the same effect, hence the cross-effect universe is the star $\{(W_0, W_u) : u \in U\}$. Safe read q checks the declared subset of hidden witnesses indexed by C_q and reports whether one qualifies. It differs between W_0 and W_u exactly when $u \in C_q$, so $S(q) = \{(W_0, W_u) : u \in C_q\}$. Costs are preserved. A sufficient non-adaptive read set therefore exists at cost K exactly when the original set-cover instance has a cover of cost K; the construction is polynomial.

Non-adaptive greedy guarantee. Weighted-set-cover greedy repeatedly chooses the read with minimum cost per newly covered pair, equivalently maximum newly covered pairs per unit cost. The standard Chvátal analysis gives cost at most $H(d)OPT$, where $d = \max_q |S(q)|$ and $H(d) \leq 1+\ln d$. This guarantee applies only to the fixed non-adaptive cover. The runtime planner recomputes compatible worlds after each observation, and we claim no approximation factor for that adaptive policy.

Projected verification. If compatible worlds W_1, W_2 satisfy $\pi(\text{Effect}(G, W_1)) = \pi(\text{Effect}(G, W_2))$ but $\text{Effect}(G, W_1) \neq \text{Effect}(G, W_2)$, projection agreement cannot establish Equation 3: the two worlds contribute distinct elements to $A(G, E, \Sigma)$. Strengthening the verifier until it separates every remaining cross-effect pair recovers the exact-effect condition.

Proof of Proposition 1. Let W_c be the complete environment at certification. Truthful admitted evidence and sound Σ imply $W_c \in \Omega(E, \Sigma)$. Equation 3 therefore gives $\text{Effect}(G, W_c) = \alpha$. Between certification and commit, a sound change detector rejects any decision-relevant change, serialization prevents it, or rereading reconstructs the singleton claim against current state. Hence an accepted commit environment W_m also satisfies $\text{Effect}(G, W_m) = \alpha$. Exact manifest matching prevents the broker from substituting any $\alpha' \neq \alpha$. The realized broker effect is therefore the effect required by G in W_m. The argument is conditional on the stated trusted boundary and excludes unmodeled executor effects.

Probability and closure. Suppose a posterior assigns probability .99 to the environment in which `delivery.tsv` is the target and .01 to the bystander environment. Equation 3 still fails because two effects remain. This is a specification distinction, not an empirical claim that calibrated thresholding must fail. In the frozen calibrated model, confidence and voting wait for decisive evidence and then agree with closure.

C Effect-Separation Acquisition Benchmark

The frozen benchmark contains 60 fixtures across two represented operator families: 36 small fixtures supporting exact adaptive dynamic programming and 24 larger scaling fixtures. Each fixture supplies adapter-declared safe reads and read budget $B = 6$; read selection never changes the singleton authorization condition.

Table 6: Realized interaction cost on the 36-fixture small regime.

Read policy	**Mean realized truth-trace cost**
Frozen strengthening-first heuristic	2.000
Adaptive Effect-Separation greedy	1.278
Adaptive dynamic programming	1.167

Effect-Separation greedy is 36% lower cost than strengthening-first. It matches the adaptive DP policy's realized trace cost on $32/36$ fixtures; its mean paired realized-cost difference is $+0.111$ interactions. The DP policy's expected objective is 1.000, distinct from its 1.167 mean cost on the realized truth traces. Per-trace DP cost is not treated as a universal lower bound. The 24 larger fixtures exercise the same two represented families without an exact adaptive-DP comparison.

Secondary acquisition comparison. The earlier budgeted read-selection benchmark measures Decision@B rather than realized interaction cost. It remains useful for comparing generic information objectives with proof-directed objectives, but is not the primary Resolve result.

Table 7: Secondary frozen Decision@B comparison.

Read objective	**Decision@B ↑**
Mutual information	0.227
One-step expected proof progress	0.523
+ proof-relevant information	0.683
Proof-aware heuristic / ExactDP	0.717

ExactDP's expected cost in this separate Decision@B benchmark is approximately 8.79. Neither benchmark relaxes Equation 3.

D Deployment Preflight

The deployment checker runs before any task agent. It compares structural declarations in Σ with substrate capabilities; it neither supplies task-instance facts nor changes Equation 3. The checked families are complete or exhaustive domains, partition coverage, catalog dependencies and cascades, POSIX identity and alias semantics, and freshness or change-detection capability.

Table 8: Frozen deployment-preflight evaluation across the five structural declaration families.

Configuration set	**Number**	**Preflight outcome**
Injected incorrect configurations	15	15 detected
Matched correct controls	15	0 rejected

This establishes detection for all 15 injected structural errors and acceptance of all 15 matched controls. It does not show that preflight can prove arbitrary semantic assertions inside a trusted adapter.

E Preserve Interventions

The intervention suite contains 16 fixtures: four base tasks crossed with four controlled post-certification conditions. Every condition contains two filesystem and two PostgreSQL fixtures. The dependency set stored with the certificate determines whether a state change is relevant; the broker separately compares the realized effect with the certified effect.

Table 9: Detailed Preserve outcomes. FS and PG denote filesystem and PostgreSQL fixtures.

Condition	**Intervention**	**FS + PG**	**Observed response**
A	No state change	$2/2 + 2/2$	Commit preserved
B	Change outside dependency objects	$2/2 + 2/2$	Certificate preserved
C	Certified dependency changes	$2/2 + 2/2$	Certificate invalidated
D	Offer $\alpha' \neq \alpha$ to the broker	$2/2 + 2/2$	Effect mismatch rejected

The exact terminal classes are:

A: `COMMIT_PRESERVED` B: `PRESERVED`
C: `CERTIFICATE_INVALIDATED` D: `BROKER_EFFECT_MISMATCH_REJECTED`

Conditions A and B distinguish dependency-sensitive preservation from a global revision check: irrelevant activity need not cancel useful authority. Conditions C and D test the two ways a previously justified decision can cease to authorize commit—the supporting facts changed, or execution no longer matches the certified effect.

F STATISTICAL DETAIL

McNemar tests use paired base-task outcome correctness, collapsing the two frozen instances of each base task. The OLD comparison is not a test only of the 42 destructive base-task transitions: it includes all terminal correctness disagreements. The VBE+ summary supplies an instance-level error count but not the variant-pair map needed to reconstruct base-task discordance.

Table 10: Available discordant paired outcomes for an exact two-sided McNemar test.

Comparison	**Full correct only**	**Comparator correct only**	**Exact p**
OLD vs. Full	56	0	2.78×10^{-17}

The 47 VBE+ errors are counted over 252 instances and cannot be substituted as 47 discordant base tasks: depending on how errors overlap across the two variants, the paired count can differ. We therefore report no VBE+ McNemar result until the per-instance pairing is available.

For zero observed destructive outcomes, the one-sided exact 95% upper bound is 1.18% for Full's $0/252$ internal result and 6.88% for each model's $0/42$ live result. Full has zero observed destructive outcomes in every one of the 11 task families. Proposal-level FAR remains $0/756$ as a descriptive count; no independent-trials interval is constructed from those proposals.

G RUNTIME COST

The deterministic timings are LOCAL_SUBSTRATE_COST, measured across the controlled filesystem, PostgreSQL, system-setting, composed-terminal, browser-profile, and online-account adapters. They characterize the instrumented test substrates and should not be read as production terminal, browser, account-service, or distributed-database latency.

Table 11: Mean local-substrate time per evaluated instance in the frozen 252-instance evaluation.

Component	**Mean seconds/instance**
Evidence read and acquisition	0.3177
Contract compilation	0.00023
Semantic evaluation	0.00172
Guard and evidence revalidation	0.3085
Total wall time	0.6282

The total median is 0.4984 s and p95 is 1.9320 s. Contract compilation and semantic evaluation together take about 2 ms on average; reads and revalidation dominate.

H INTERNAL EVALUATION PROTOCOL AND COMPLETE RESULTS

Frozen paired test. The 252 held-out task instances span six controlled substrates—filesystem, PostgreSQL, system settings, composed terminal operations, browser profiles, and controlled online accounts—in 11 structured families. They are generated from 126 base tasks, with two frozen instances per base task. Every policy receives the same instances. Terminal state is scored against deterministic reference semantics as lawful commit, correct withhold, guard rejection, or destructive/reference-forbidden effect. Rates and terminal totals use the 252 instances; paired significance tests collapse the variants to base-task units.

Policies. OLD is the actual pre-final rule. Always Abstain never commits. VBE+ implements Algorithm 2: it uses the same Σ as HypoProbe, expands the complete declared dependency and cascade closure, and iteratively performs safe reads according to the fixed five-check order. Missing evidence triggers the corresponding read and a fresh pass through the checklist rather than immediate refusal. VBE+ does not enumerate compatible worlds, test cross-effect pair separation, or use HypoProbe's

read-gain objective. Both policies share the restricted executor, realized-change detector, and pre-commit revalidation. Singleton/no-synthesis applies Equation 3 to available evidence and withholds instead of requesting a strengthening observation. Full HypoProbe requests a safe distinguishing read when one is available and otherwise withholds.

After two-instance expansion, OLD has $84/252$ destructive outcomes and $84/756$ false authorizations. Full HypoProbe has $0/252$ destructive outcomes and $0/756$ false authorizations, with 74 lawful commits, 154 correct withholds, and 24 guard rejections. These mutually exclusive terminal classes sum to 252, and all decisions match the reference; the 74 commit instances realize 346 required physical effects ($177 + 169$ across the two frozen instance variants). VBE+ has $47/252$ destructive outcomes, 100% lawful coverage, 81.3% outcome correctness, and 1.0 mean reads. These are the results of Algorithm 2 with dependency closure, iterative evidence acquisition, and the shared execution pipeline enabled.

Acquisition, confidence, and cost. Singleton/no-synthesis has DAR 0 and lawful commit coverage $23/37$ over base tasks, equivalently $46/74 = 62.2\%$ over evaluated instances; Full HypoProbe has coverage $37/37 = 74/74 = 100\%$. Singleton's correct terminals are 23 lawful commits, 77 correct withholds, and 12 guard rejections over the 126 base tasks, yielding $112/126 = 88.9\%$ outcome correctness; expansion doubles these counts. On the calibrated evidence model, thresholds 0.50–0.99 and majority, two-thirds, and unanimous voting have no destructive actions and match Full HypoProbe with the same decisive reads. Under limited evidence, the confidence policy abstains until sufficient evidence is available. Full HypoProbe is reported at 1.33 reads per evaluated instance, compared with 1.0 for VBE+. This accounting treats one strengthening read on 42 base tasks as 84 evaluated instances and 84 extra reads, giving $(252 + 84)/252 = 1.33$; the released read-event logs are required to verify both policies' event-level counts. Appendix G gives the timing decomposition.

Database-native control. On the 70-instance PostgreSQL subset, the select/count/transaction policy has DAR 0 and lawful coverage 100%. It uses row locking where appropriate and rechecks, rolls back, or refuses effects beyond its native guarantees.

Independent reference checks. A second deterministic implementation imports neither the frozen oracle nor method decisions. It independently reconstructs the valid exact-effect set, proposal label, and expected terminal class, agreeing on all $252/252$ tasks. Separately, two blinded model annotators reconstruct exact effects directly from G, complete state W, and the declared Σ rules without access to oracle or method outputs. The DeepSeek-family annotator matches $252/252$ frozen references. The Qwen-family annotator yields 124 completed structured outputs, all matching the reference; the other 128 packets are unscored because they do not yield completed structured annotations. The frozen manuscript summary does not disaggregate timeout, provider, and malformed-output causes, so $124/124$ is a complete-case result and no agreement claim is made for the missing 128.

Table 12: Blinded independent semantic validation on the 252 frozen TEST instances.

Independent semantic check	Scored	Exact-effect agreement
Qwen-family, completed outputs	124	$124/124$
Qwen-family, no completed output	128	Unscored
DeepSeek-family	252	$252/252$
Cross-model consensus	124	$124/124$

For every completed annotation, agreement is exact for mutation versus no mutation, operation, target set, new values, and represented dependency footprint. Task identity, family, expected labels, method outputs, and original object naming were hidden; object identifiers were independently permuted per task. The packet protocol, normalizer, and all 252 packets were frozen before annotation, and both annotation passes were frozen before the reference was revealed to the scorer. Because Qwen completion is only $124/252$, its complete-case agreement may be subject to selection bias.

Live protocol. The same representative 42-task mixture is evaluated separately with each model. Fifteen cases require a lawful commit and 27 require withhold. Nine cases per model begin with a destructive proposal and test intercept–evidence–replan–commit. Per model, DAR is $0/42$ with a one-

sided 95% upper bound of 6.88%, and recovery is $9/9$ with a two-sided 95% interval approximately $[66.4\%, 100\%]$. There are no timeouts and the runs are not pooled.

I Component Validation

Evidence laundering. The paired validation changes only the offered observation record. Requirement-bound admission rejects all 40 attacks, accepts all matched controls, and causes no trusted-cache pollution or unauthorized capability request.

Semantic differential check. The check covers 27 formula forms, 463 admissible partial-evidence states, and 6,316 completions. Direct enumeration and the symbolic evaluator agree on every case, with zero premature true, premature false, unsound authorization, or premature authorization.

Mutation and mechanism checks. Fifteen of 16 preregistered semantic mutations are killed. The remaining mutation removes a closure optimization but is observationally equivalent because filtering in $\Omega(E, \Sigma)$ enforces the same constraint. The mechanism suite passes all ten preregistered criteria, including negation, contradictory threshold–uniqueness requests, complete partitions, canonical identity, exact scope, silent dependency drift, and exact realized effect.

Read-selection validation. Appendix C reports both the primary Effect-Separation interaction-cost benchmark and the secondary Decision@B comparison. Both vary only evidence selection and leave Equation 3 unchanged.

J External Authorization-Boundary Transfer

Outcome-independent cohort construction. The consequential exact-effect cohort is determined before any policy result is produced. The script `extscale_manifest.py` freezes the screen with `frozen_before_any_outcome: true` and seed `20260925`. It reads only official benchmark fields and mechanical environment probes; it does not read decisions, traces, scores, or any other HypoProbe output. This frozen manifest is the sole entry point to the cohort analysis.

The screen is the conjunction of six gates, denoted C1–C6 in the manifest and unrelated to the later C0–C3 impact labels. Applied to the frozen 540-task population, their conjunction retains 374 tasks and excludes 166. C1 requires the task's site set to be a subset of sites that the environment probe can reach with HTTP status below 500. C2 requires the official `agent_types` field to equal `mutate`, so the task defines a mutating effect. C3 requires `representability == RE1` in the frozen EVID-29 screen manifest: the exact effect is expressible and its evidence source lies on the admitted observation surface. C4 requires the official `config_files/{tid}.json`. C5 requires a mechanically evaluable official evaluator of type `program_html`, `url_match`, or `string_match`. C6 requires the official `intent_template_id` to be covered by one of 37 frozen reusable executor templates. The final conjunction is invariant to gate order; any gate-specific attrition table must be regenerated from this same frozen manifest under the listed fixed short-circuit order, because marginal rejection counts can change when a task fails multiple gates.

WebArena-Verified protocol. The frozen external-scale population contains 540 WebArena-Verified tasks evaluated under three policy arms, for 1620 completed arm-runs. The tasks, websites, initial state, mutating actions, and evaluators are benchmark-defined. The protocol uses the benchmark reset, navigation to the consequential boundary without revealing the gold effect, a browser observation, a semantic-effect proposal, frozen authorization, a real site mutation when authorized, and the benchmark evaluator. Across the full 540-task population, frozen HypoProbe has 75.7% commit precision $(340/449)$ and 89.9% commit recall $(340/378)$, while the frozen VBE+ summary reports 43% commit precision. The available manuscript snapshot does not contain VBE+'s TP, FP, FN, and TN counts, so we do not infer its confusion matrix or recall; these counts are required in the release. Within the finalized method-valid consequential cohort, 374 tasks support exact-effect authorization scoring; among HypoProbe's 302 authorized mutations, none produces a destructive outcome $(0/302)$.

Table 13: Full-population confusion matrix for the frozen HypoProbe WebArena-Verified arm. Commit is the positive class.

Reference \ decision	Commit	Refuse	Total
Should commit	340 (TP)	38 (FN)	378
Should refuse	109 (FP)	53 (TN)	162
Total	449	91	540

The matrix yields commit precision $340/449 = 75.7\%$, commit recall $340/378 = 89.9\%$, refusal precision $53/91 = 58.2\%$, refusal recall (specificity) $53/162 = 32.7\%$, and overall decision accuracy $393/540 = 72.8\%$.

Table 14: Two-stage triage of all 109 full-population false-positive commits. C0–C2 classify realized outcomes; C3 is an apparent high-impact screening bucket adjudicated separately, not a fourth point on one realized-harm scale.

Class	Triage label	Operational definition	Count
C0	No persistent change	No persistent state changed, or the change was session-local and disappeared on refresh.	32
C1	Reversible / low impact	UI preference, temporary field, cache, or similar non-core state that can be cleanly undone.	23
C2	User data, reversible	The wrong user-data item changed, but the original information was not lost and a compensating operation can restore it.	35
C3	Apparent high-impact candidate	The recorded operation appears potentially irreversible and is routed to second-stage trace adjudication, regardless of realized state.	19
		Total	**109**

This is a triage partition, not an ordinal realized-harm scale. C0–C2 classify realized outcomes, including 35 C2 reversible wrong user-data modifications. C3 preserves the pre-adjudication screening label: among its 19 candidates, a stricter read-only trace audit finds 17 realized wrong writes and two evaluator artifacts with no state change. The latter would be C0 under a realized-outcome-only taxonomy, but remain C3 here so that screening and adjudication are not conflated. Consequently, 17 is an audit result for the C3 subset, not the number of all realized wrong writes in the full 109-record population.

Of the 17 C3-subset writes, eight occur on the r2 fabricated GitLab/navigation surface and nine on the repaired r5 admin surface where the requested target entity is absent. In both families, the gate authorizes from a convincing but misleading surface; the shared failure condition is a distorted observation surface combined with target absence. Frozen wrong authorization is zero on the repaired r6/r7 surfaces. The two score-1.0 GitLab artifacts arise from residual merge-request state; all three arms share them. Separately, the ungated and VBE+ arms each realize $22/44$ wrong writes on a reported hazardous slice. The available summary does not establish a task-ID join between that 44-case slice and the 19 C3 candidates, so $22/44$ and $17/19$ are not used as a paired comparison.

The $449/91$ counts above are mutually exclusive final decisions, one per task. Separately, trace instrumentation records intermediate events before final decision aggregation: 229 planner-rejection events, 127 NOT_IDENTIFIABLE events, 36 unresolved-navigation events, 24 empty-surface events, and three execution-failure events, totaling 419 occurrences. The recorded event operations are themselves non-writing, but the occurrences are neither mutually exclusive nor one-to-one with tasks: one trace may contribute multiple events before its single final decision. Thus 419 is not a count of final refusals, does not classify final trace outcomes, and is not directly comparable with the 91 REFUSE decisions. The separate $0/302$ destructive-outcome endpoint is restricted to the 374 finalized method-valid consequential cases and does not classify all 109 full-population false-positive commits.

One r7 boundary case (task 437) exposes a different executor obligation. The gate correctly returns WITHHOLD(NOT_IDENTIFIABLE), and no order is placed, but the `sh_reorder` executor clicks "Reorder" to observe the cart and thereby adds an item. This reversible cart change is not an authorized mutation and receives score zero, yet it shows that an observation can have a native site side effect.

Any reuse of such an executor must include click-to-observe side effects in the declared effect footprint.

Table 15: Frozen WebArena-Verified external transfer. RIA denotes a reference-inconsistent effect authorization.

Slice	**Scale / comparator**	**Results**
External population	540 tasks; 3 arms; 1620 runs	Main endpoint scoped below
Decision outcomes	Full 540-task population	HypoProbe: 75.7% precision; 89.9% recall; VBE+: 43% aggregate precision
False-positive triage	All 109 HypoProbe FP commits	Realized C0/C1/C2: 32/23/35; C3 screen: 19 (17 writes, 2 artifacts)
Consequential exact-effect cohort	374 finalized method-valid tasks	HypoProbe: $0/302$ destructive outcomes among authorized mutations
Recall-reorder illustration	VBE+: 5/5 RIA	HypoProbe: 5/5 withholds; 0 reorders
Explicit-target control	Target already identified	18/18 direct commits; 0 extra reads; 0 withholds

The recall tasks illustrate fail-closed identification: the page shows prior orders but does not identify which one the user means, so executability does not determine intent. The explicit-target controls show the complementary behavior: once the requested order identifies one exact effect, the gate neither probes nor withholds. Across the full 540-task population, the evaluation reports 75.7% commit precision and 89.9% commit recall for HypoProbe, compared with 43% aggregate precision for VBE+. Within the C3 candidate subset, trace adjudication localizes 17 wrong writes to invalid r2/r5 observation fixtures and identifies two evaluator artifacts; it does not reclassify the C0–C2 realized outcomes. On the narrower 374 finalized method-valid consequential cases, the separate HypoProbe safety endpoint is $0/302$ observed destructive outcomes among authorized mutations. These scoped results are not a claim of general browser-agent safety.

K Reproducibility and Provider Details

The release will record software versions, fixed seeds, task manifests, expected exact effects, authoritative-source declarations, evidence records, guard outcomes, final state diffs for every controlled substrate, browser-boundary traces, and per-task terminal labels. It will include scripts that verify the 126-base-task/252-instance transitions, variant-pair mappings and paired statistics for every comparator, $84/756 \rightarrow 0/756$ false-authority result, Effect-Separation fixtures, safe-read declarations, costs, planner traces and scorers, preflight fixtures, Preserve interventions, blinded annotation packets and outputs, Qwen completion-status and failure-reason codes, the 84 strengthening-read events, timing records, and per-model live counts. For VBE+, the release will additionally include the expanded dependency closure, ordered check outcomes, missing-evidence read requests, checklist restarts, freshness results, restricted-executor inputs, and realized-effect comparisons for every task. External artifacts will include the frozen WebArena-Verified 540-task manifest, three-arm records with per-arm TP/FP/FN/TN counts, external-scale traces, per-task authorization outcomes, hazardous-slice membership and task-ID joins, all 109 false-positive impact labels, trace adjudications and evaluator-artifact flags, the task-437 observation-side-effect record, and evaluator/state records needed to reproduce the final endpoint.