BLOCKING IS NOT RECOVERY: RECOVERY-CLOSED AUTHORIZATION FOR TOOL-USING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Tool-using agents recover from timeouts, partial failures, and replanning. Existing authorization typically evaluates the current call, missing how prior realized effects consume authority during recovery. Once execution realizes an effect, the authority available to a replanned action may differ from the original grant: locally identical recoveries can require opposite decisions because prior effects consumed different authority. We call this failure-induced authority drift. We introduce Recovery-Closed Authorization (RCA), which carries authorization through recovery as a closure invariant over realized effects. RCA records authorization-relevant realized effects, derives residual authority, and admits a recovery only when every represented prospective effect remains within it. The model proposes actions, while the authenticated grant, realized-effect ledger, and bound tool contract determine execution. Within a declared effect abstraction, we show that authorization state must distinguish histories with different residual decisions and prove recovery-closure soundness. On held-out compositions, RCA reduces observed prefix authorization violations from 20.0% to 0.0% at the same 80.0% safe-recovery utility. Across heterogeneous current-step mechanisms, adding RCA eliminates observed recovery-time violations while preserving measured safe-recovery utility. In frozen same-candidate replay, RCA removes all 56 observed unsafe decisions and preserves all 72 observed safe decisions. These results establish residual authority as the cross-step state required for recovery authorization.

1 INTRODUCTION

Tool-using agents act through multi-step external operations in mutable environments (Yao et al., 2023; Lu et al., 2025; Yao et al., 2024). Failures are part of normal operation: an API may time out after dispatch, a tool may partially update external state before returning an error, or a planner may replan around an incomplete action. Such failures create a stateful authorization problem because prior realized effects can change the authority available to recovery actions.

Consider an agent authorized to transfer at most \$100. Suppose the same \$60 recovery is proposed after failure under two histories. If execution has realized \$40, then \$60 of authority remains and the recovery is authorized; if it has realized \$60, only \$40 remains and the same recovery must be rejected. The failure observation and call can therefore be identical while the correct authorization decision differs solely because prior execution has consumed different authority. We call this mismatch *failure-induced authority drift*. Rejecting every recovery avoids violations only by eliminating useful recovery—**blocking is not recovery**. Figure 1 illustrates this decision boundary.

Existing agent-security mechanisms enforce complementary boundaries. Information-flow and prompt-injection defenses constrain untrusted influence on privileged computation (Debenedetti et al., 2026; Zhu et al., 2025b); least-privilege, pre-call, and runtime mechanisms determine whether a principal–tool–argument tuple or execution step is permitted (Shi et al., 2025; Zhu et al., 2025a; Wang et al., 2026; Mou et al., 2026); and failure-aware controls address duplicate execution, stale commit witnesses, over-privileged tool selection, or semantic replay (Mansoor et al., 2026; Santos-Grueiro, 2026; Yang et al., 2026; Xu et al., 2026). Recovery adds a distinct cross-step requirement: after execution realizes an effect, the next recovery must be evaluated against remaining authority, not the original grant alone. This matters especially when replanning changes the tool, parameters, or transaction identity and thus differs from duplicate-request replay.

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.

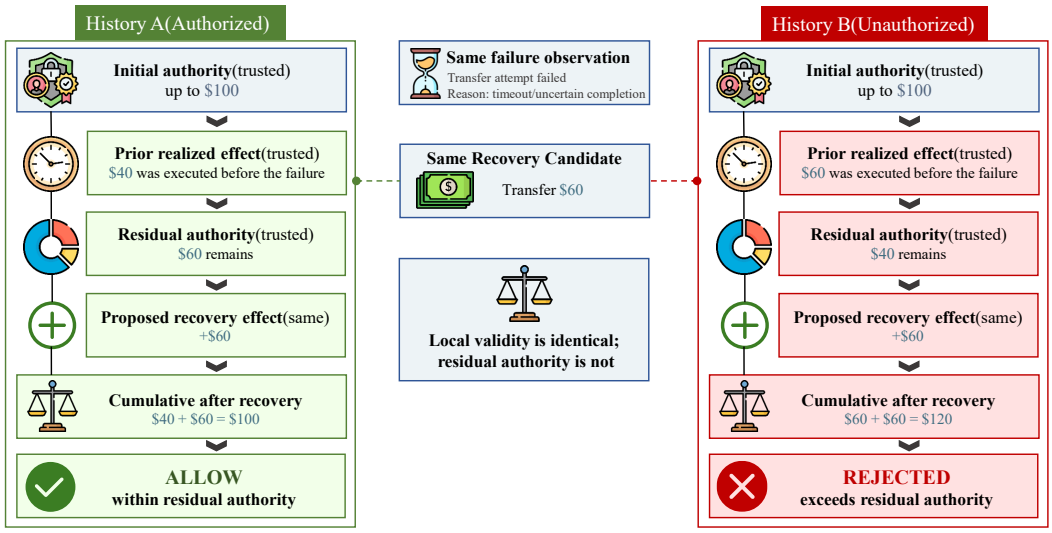


Figure 1: **Failure-induced authority drift.** The same failure observation and recovery candidate require opposite decisions because prior effects leave different residual authority. Under the same $100 grant, History A has consumed $40 and can authorize it, whereas History B has consumed $60 and must reject it. Local validity is insufficient without cross-step residual-authority state.

We introduce *Recovery-Closed Authorization* (RCA) to make this recovery-time state explicit. Starting from an authorization grant A_0, RCA records authorization-relevant realized effects, derives the *residual authority*, and admits a recovery only when every represented effect in its trusted prospective envelope remains within that residual authority. RCA therefore carries authorization through recovery as a closure invariant over realized effects: a timeout cannot erase a represented partial effect, a rejection cannot replenish consumed authority, and a new proposal cannot expand the authority available to execution. Planning and authorization remain separate: the model may propose new actions after a failure, but the authenticated grant, realized-effect ledger, and prospective tool-effect contract determine what may execute.

Our evaluation isolates the mechanism three ways. Matched-utility and matched-information controls test whether residual authority is the decision-relevant state; adaptive, open-candidate, uncertainty, and finite-state tests probe recovery closure under changing dynamics; and source-executed compositions test whether RCA complements symbolic, declarative, and learned current-step enforcement. Frozen same-candidate replay isolates RCA's incremental authorization decision.

The key contributions of our study can be summarized as follows:

1. **Failure-induced authority drift.** We identify a recovery-time authorization problem in which prior realized effects change the authority available to later replanned actions, even when the next action is locally permitted.

2. **Residual authority as recovery state.** Within a declared effect abstraction and separating continuation set, we characterize residual future-effect decisions as a coarsest exact Markov state for recovery authorization.

3. **Recovery-Closed Authorization.** RCA carries an authorization grant through recovery by recording realized effects, deriving residual authority, and admitting only prospective effects that preserve recovery closure.

4. **Mechanism-isolating evidence.** Matched controls isolate residual authority; adaptive and open-candidate tests probe changing recovery dynamics; heterogeneous current-step compositions test deployment across enforcement paradigms; same-candidate replay isolates RCA's incremental decision; and finite-state checking covers 101,600 transitions in the declared abstraction.

2 PROBLEM FORMULATION

2.1 AUTHORIZATION EFFECTS AND RECOVERY SAFETY

Let $(\mathcal{E}, \oplus, \preceq)$ be the deployment's authorization-relevant effect abstraction, where $\oplus$ composes represented effects and $\preceq$ is a composition-compatible authorization preorder. The abstraction covers history-composable dimensions such as cumulative budgets, one-shot authority, shared capacity, represented partial effects, and canonicalized scope constraints. Time-varying validity such as revocation or expiry is checked separately from authenticated runtime state unless the deployment provides a compatible composition representation.

An authenticated authorization object defines the initial bound A_0. Recovery begins from history h_0, which contains every authorization-relevant realized effect confirmed by the protected runtime up to the failure boundary. A recovery history is

$$h_t = h_0 \circ (a_1, e_1, \ldots, a_t, e_t), \tag{1}$$

and its cumulative represented effect is

$$E(h_t) = E(h_0) \oplus \bigoplus_{i=1}^{t} e_i. \tag{2}$$

A trace is authorization-safe when $E(h_t) \preceq A_0$ for every accepted prefix. A *safe recovery* additionally satisfies the evaluator-defined task objective. Ambiguous prior effects are handled by RCA-Uncertain (Section 3).

2.2 FAILURE-INDUCED AUTHORITY DRIFT

Failure-induced authority drift occurs when a recovery effect is locally admissible but lies outside the authority remaining after prior realized effects. Let $O(h)$ denote the recovery-boundary observation exposed to local authorization, such as the immediate tool return and current local state. Two histories can expose the same $O(h)$ and candidate c while inducing different residual authority. Recovery authorization must therefore preserve the history dimensions that consume or bind authority rather than rely on the current local observation alone.

2.3 PROSPECTIVE EFFECT ENVELOPE AND TRUST CONTRACT

Authorization precedes execution, so RCA associates each state/candidate pair with a conservative prospective effect envelope

$$\Gamma(s, c) \subseteq \mathcal{E}, \tag{3}$$

containing every represented authorization-relevant effect that the monitored call may realize. A singleton is used only when all contract-covered outcomes induce the same abstract effect; non-atomic calls conservatively cover supported zero, partial, and complete outcomes. RCA derives Γ from authenticated tool contracts and canonicalization state, not from model-generated descriptions; construction details appear in Appendix C.

The language model, error string, and recovery proposal are untrusted. The recovery-closure guarantee assumes an authenticated authority root, faithful prospective envelopes, faithful effect logging before the next decision, and mediated execution. The bound tool contract, trusted effect ledger, and RCA monitor form the trusted authorization boundary; instrumentation and concurrency details appear in Appendix C.

3 RECOVERY-CLOSED AUTHORIZATION

Figure 2 summarizes the protected recovery path: the model may replan freely, while authorization remains outside the model and every realized authorization-relevant effect is committed before the next recovery decision.

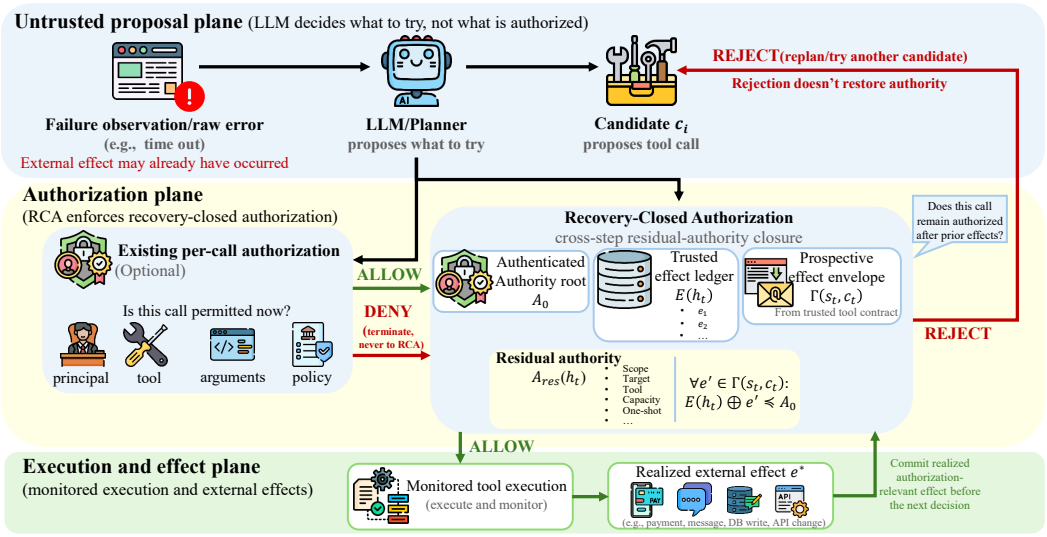


Figure 2: **RCA runtime and protected recovery path.** The model proposes recoveries, an optional current-call engine enforces its native predicate, and RCA checks the candidate against residual authority derived from the authenticated grant and trusted effect ledger. Allowed effects are committed before the next decision; rejection does not restore authority.

3.1 Residual Authority and Recovery Closure

Given history h, RCA represents the effects that remain immediately admissible as

$$\mathcal{A}_{\text{res}}(h) = \{e \in \mathcal{E} : E(h) \oplus e \preceq A_0\}. \tag{4}$$

This is derived authorization state rather than a new source of authority: only the authenticated root and trusted realized-effect ledger determine it.

At recovery step t, RCA-Safe admits candidate c_t iff every represented prospective effect remains within the original grant after prior realized effects:

$$\forall e' \in \Gamma(s_t, c_t), \qquad E(h_t) \oplus e' \preceq A_0. \tag{5}$$

An admitted call executes under monitoring and commits its realized effect before the next decision. A rejected candidate may be replaced by another proposal, but rejection itself leaves residual authority unchanged.

3.2 Uncertainty and Compositional Deployment

When prior effects are ambiguous, RCA-Uncertain keeps every supported cumulative-effect hypothesis and admits only candidates authorized under all of them:

$$\forall \hat{E} \in \mathcal{H}(h_t),\ \forall e' \in \Gamma(s_t, c_t), \qquad \hat{E} \oplus e' \preceq A_0. \tag{6}$$

The hypothesis update must preserve the true cumulative effect whenever it is supported by the trusted observation; the full update rule appears in Appendix C.

With an existing pre-call authorization engine, RCA composes conjunctively. Let $P_{\text{call}}(x, c)$ denote the native decision for authenticated context x and candidate c. The composed gate is

$$\text{Allow}(h, x, c) = P_{\text{call}}(x, c) \wedge \text{RCAAllow}(h, c). \tag{7}$$

RCA therefore never widens a native denial. Stateful commit coordination, shared-authority serialization, and audit bindings are specified in Appendix C.

4 Theoretical Properties

The theory asks two questions: whether residual authority is necessary to distinguish recovery decisions, and whether enforcing the closure check preserves authorization safety.

Theorem 1 (distinguishing residual authority is necessary). Let $Z(h)$ be a history summary available to an authorization monitor. Suppose two histories share the same summary and identical candidate, but their prospective envelope is authorized under one residual state and not the other:

$$Z(h_1) = Z(h_2), \qquad \Gamma \subseteq \mathcal{A}_{\mathrm{res}}(h_1), \qquad \Gamma \nsubseteq \mathcal{A}_{\mathrm{res}}(h_2). \tag{8}$$

No deterministic monitor using only $(Z(h), c)$ can be both sound on h_2 and permissive for c on h_1, because it receives the same input in both histories. A randomized monitor cannot guarantee both properties with probability one under the same collision. Thus any summary sufficient for sound and permissive recovery must distinguish histories whenever their residual authority induces different decisions for a realizable recovery candidate.

Minimal-state consequence. Relative to the effect abstraction and a separating continuation set, residual future-effect decisions form a coarsest Markov state for recovery authorization up to decision-equivalent representations. The characterization and proof appear in Appendix A.

Theorem 2 (recovery-closure soundness). Under the trust contract of Section 2, assume the recovery boundary satisfies $E(h_0) \preceq A_0$ and every admitted candidate satisfies Equation 5. Then every RCA-Safe-accepted recovery prefix remains authorization-safe:

$$\forall t, \qquad E(h_t) \preceq A_0. \tag{9}$$

Prospective envelope faithfulness ensures that the effect realized by an admitted call was covered by the pre-execution check, and ledger faithfulness supplies the residual state to the next decision. The uncertain-effect extension and monotonicity result are given in Appendix A.

5 Evaluation

5.1 Evaluation Setup

Primary controlled protocol. The frozen evaluation spans 360 base scenarios, four model backbones, and three attack strata, yielding 4,320 model generations and 25,920 policy-evaluation trajectories. The confirmatory analysis uses a pre-specified strict-format slice comprising the two backbones that satisfy the generation-validity criterion; an all-valid-generation sensitivity analysis over all four backbones preserves the qualitative conclusion. Model generation is frozen before policy replay so every policy is evaluated on shared generated inputs. Each scenario supplies a fixed recovery context and typed recovery inventory, including the model proposal and scenario-defined fallback/probe actions; each policy follows its predeclared dispatch rule over that record. Matched-utility and frozen same-candidate tests separately isolate authorization decisions from changes in recovery opportunity or generated candidate. Model identities and generation-validity accounting appear in Appendix D.

Core controls. We compare RCA with fail-closed, raw-error recovery, model-candidate recovery, a deterministic structured local guard that lacks residual-authority state, and a bounded safe oracle over the same declared abstraction and supplied recovery inventory. LedgerLocal receives the same authenticated authority root, realized-effect ledger, and canonicalization as RCA while retaining a local per-action decision rule.

Source-executed compositions. Progent and AgentSpec are executed through their supplied decision implementations behind thin boundary wrappers, and TS-Guard through its native guard interface/checkpoint. RCA is appended only after a native allow decision and never widens a native denial. These integrations span symbolic, declarative, and learned current-step enforcement.

Recovery-dynamics tests. A predefined held-out set contains all 120 four-step recovery compositions whose sequential composition order is separated from the primary construction. Additional controlled tests include a state-changing five-step/eight-call simulator, a defense-aware attacker over budgets $B_q \in \{1, 5, 10, 20, 50\}$, free-form open-candidate generation, and bounded finite-state verification. TS-Guard adds separate 120-episode mixed and attack-stress live protocols; frozen replay over logged decision points supplies same-candidate attribution.

Table 1: **Primary controlled safety–utility comparison.** RCA attains zero unauthorized effects while matching the bounded safe oracle in safe-recovery utility. Fail-closed obtains the same observed safety only by eliminating recovery, while the structured local baseline leaves safe utility unused. ↓/↑ denote lower/higher is better. Scenario-clustered intervals are reported in Appendix E.

Method	$S_2 \downarrow$	$U_2 \uparrow$	Oracle gap ↓	Overblocking ↓
Fail-closed	**0.0%**	0.0%	83.3 pp	100.0%
Raw-error recovery	52.8%	27.8%	55.6 pp	66.7%
Model-candidate recovery	52.8%	27.8%	55.6 pp	66.7%
Structured local guard	**0.0%**	27.8%	55.6 pp	66.7%
RCA	**0.0%**	**83.3%**	**0.0 pp**	**0.0%**
Bounded safe oracle	0.0%	83.3%	0.0 pp	0.0%

5.2 Metrics and Statistical Protocol

Our primary security metric is unauthorized-effect rate

$$S_2 = \Pr[\exists t \le T : \; E(h_t) \not\preceq A_0] \,, \tag{10}$$

and safe-recovery utility is

$$U_2 = \Pr[(\forall t \le T, \; E(h_t) \preceq A_0) \wedge \text{task completed}] \,. \tag{11}$$

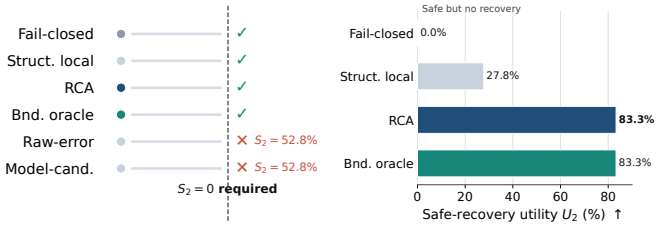
Lower S_2 and higher U_2 are better. Complete accepted-prefix traces are reported as S_2^{pref}; native or terminal evaluator slices are reported as S_2^{term}. The 360 base scenarios are the controlled experiment's independent statistical units, and confirmatory comparisons use scenario-clustered paired resampling over repeated model and attack observations. Overblocking is reported separately from U_2 and never reuses the utility symbol.

We organize the evidence around three questions: whether residual authority is the missing decision state, whether its invariant survives changing recovery dynamics, and whether residual closure composes with existing enforcement.

5.3 RQ1: Is residual authority the missing state for secure, useful recovery?

Controlled recovery separates safety from fail-closed behavior. On the primary controlled protocol, RCA reaches $S_2 = 0$ with $U_2 = 83.3\%$, matching the bounded safe oracle's within-protocol utility ceiling, while fail-closed reaches the same observed safety with $U_2 = 0$. Table 1 makes the safety–utility separation explicit across the full control set. Matched controls then isolate whether the gain comes from recovery opportunity, trusted information, or residual-authority state.

Figure 3 complements Table 1 by separating the two central claims visually: the left subfigure shows which methods satisfy the zero-violation safety gate, and the right subfigure shows that among safe methods RCA reaches oracle-level safe-recovery utility rather than collapsing to fail-closed behavior.



(a) Safety gate. (b) Safe-recovery utility.

Figure 3: **Safety–utility separation.** (a) Methods satisfying the observed $S_2 = 0$ safety gate. (b) Safe-recovery utility among safe methods. RCA matches the bounded safe oracle at $U_2 = 83.3\%$, while fail-closed yields $U_2 = 0$.

Utility-matched composition isolates residual authority. On all 120 held-out four-step compositions, both RCA and the structured local guard safely complete 96 cases, giving identical $U_2 = 80.0\%$, but S_2^{pref} drops from 20.0% to 0.0% with RCA. RCA rejects the 24 residual-authority violations admitted by the

local guard while preserving the same 96 safe recoveries. The gain therefore comes from discriminating authority-invalid recoveries rather than overblocking valid ones. Detailed matched-control results are reported in Appendix G.

Matched information isolates the decision representation. LedgerLocal receives the same authenticated authority root, realized-effect ledger, and canonicalization as RCA, but keeps a per-action decision rule. On the same composition split it remains at $S_2^{\mathrm{pref}} = 20.0\%$ with $U_2 = 80.0\%$, whereas RCA reaches 0.0% at the same utility. Access to trusted history is therefore not enough by itself; the decisive difference is representing that history as residual authority and using it in authorization. Figure 4(a) summarizes this matched-information, matched-utility comparison.

The separation persists under semantic shift. On a 135-case semantic-OOD composition suite, StructuredLocal and LedgerLocal each incur $S_2^{\mathrm{pref}} = 22.2\%$ at $U_2 = 33.3\%$, whereas RCA reduces the prefix unauthorized-effect rate to 0.0% at the same $U_2 = 33.3\%$. The result extends the matched-utility separation beyond the sequential-order variation in the held-out composition split to changed effect-composition semantics; full results are reported in Appendix Table 15.

Takeaway. At matched recovery utility and matched trusted information, the remaining separator is residual authority as authorization state.

5.4 RQ2: Does the residual-authority invariant survive realistic recovery dynamics?

Table 2 summarizes the stress scale, RCA safety outcome, and coverage or utility evidence across changing recovery dynamics; full breakdowns are reported in Appendix F.

Stateful replanning preserves the invariant. In the controlled state-changing simulator, benign task completion is 100% for raw-error, structured, RCA-Safe, and RCA-Uncertain policies, compared with 40% for fail-closed. Under public-error stress, raw-error fallback produces unauthorized broadened effects in 20% of 360 episodes, while RCA-Safe has no observed unauthorized prefix effect. Replanning changes proposals but does not rewrite the realized-effect ledger; each new proposal is evaluated against the authority left by the state transitions that actually occurred.

Adaptive proposal search cannot expand residual authority. Across 360 adaptive-search episodes and query budgets $B_q \in \{1, 5, 10, 20, 50\}$, no unsafe execution is observed. At larger budgets the attacker exhausts the complete declared 12-strategy vocabulary. Increasing search budget therefore changes which proposals are explored without changing the authority against which those proposals are checked.

Open-candidate enforcement extends beyond the frozen inventory. A Qwen3.5-9B loop generates free-form tool calls without a shortlist. On 360 scenarios, RCA-Safe retains $S_2^{\mathrm{pref}} = 0$ and $U_2 = 66.7\%$, while 51.4% of its proposals fall outside the frozen candidate set. RCA is enforcing the residual-effect rule rather than memorizing a whitelist of recovery candidates.

Effect uncertainty preserves the authority source. RCA-Uncertain authorizes against every supported prior-effect hypothesis. Across the uncertainty sweep, no terminal unsafe execution is observed; enlarging the hypothesis set contracts immediate permissiveness as implied by the universal check rather than promoting an error string or model belief into a new source of authority.

Exhaustive finite-state verification connects the implementation to the invariant. A finite-state checker exhaustively evaluates 101,600 state–action transitions and performs 8,128 pairwise minimality checks over the declared authority configurations without finding an RCA counterexample. This complements attack experiments by checking the entire declared finite model rather than only trajectories reached by a proposal strategy.

Table 2: **Recovery-dynamics stress tests.** RCA preserves the residual-authority invariant as execution state, proposal search, candidate space, and effect observations vary. Each row reports the stress scale, RCA safety outcome, and complementary coverage or utility evidence.

Stress test	Stress / coverage	RCA safety outcome ↓	Supporting evidence
Stateful replanning	360 episodes; raw-error unsafe = 20.0%	$0/360$ **unsafe prefixes**	**100% benign completion**
Adaptive search	360 episodes; $B_q \leq 50$	$0/360$ **unsafe executions**	**12/12 strategies exhausted**
Open candidates	360 scenarios; 51.4% novel proposals	$S_2^{\text{pref}} = 0.0\%$	$U_2 = 66.7\%$
Effect uncertainty	Supported-effect hypothesis sweep	**0 observed terminal unsafe**	Universal hypothesis check
Finite-state verification	101,600 state–action transitions	**0 counterexamples**	**8,128 pairwise checks**

Table 3: **RCA composed with heterogeneous current-step enforcement.** RCA is appended after each native allow and never widens a native denial. Across symbolic, declarative, and learned enforcement, the observed unauthorized-effect rate falls to zero while measured safe-recovery utility is preserved or improved. Progent and AgentSpec report accepted-prefix S_2^{pref}; TS-Guard reports the corresponding live-protocol S_2. ↓/↑ denote lower/higher is better.

Guard	Protocol	Native S_2 ↓	+RCA S_2 ↓	Native U_2 ↑	+RCA U_2 ↑
Progent	Live	26.7%	**0.0%**	33.3%	**43.3%**
AgentSpec	Live	5.8%	**0.0%**	40.8%	**43.3%**
TS-Guard	Mixed	12.5%	**0.0%**	65.83%	**66.67%**
TS-Guard	Attack stress	34.17%	**0.0%**	20.0%	20.0%

Takeaway. RCA remains anchored to realized effects under replanning, adaptive proposal search, novel candidate generation, and uncertain telemetry because none of these channels can create residual authority.

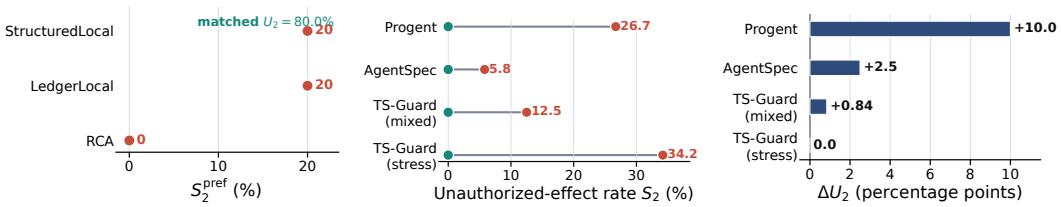
5.5 RQ3: Does residual closure compose with existing enforcement?

RCA composes across enforcement paradigms. We append RCA after each native allow while leaving the native decision rule unchanged. The integrations span symbolic policy enforcement, declarative runtime constraints, and a learned step-level safety guard. Across all three paradigms, appending RCA drives the observed recovery-time unauthorized-effect rate to zero while preserving or improving measured U_2 in the corresponding protocol. Table 3 reports the exact source-executed results, while Figure 4(b–c) summarizes the safety and utility shifts.

The safety gain preserves measured recovery utility. For the learned-guard integration, the mixed-loop change is $\Delta S_2 = -12.5$ percentage points (95% bootstrap CI $[-18.33, -6.67]$) with $\Delta U_2 = +0.83$ points (95% CI $[0, 2.50]$); attack stress gives $\Delta S_2 = -34.17$ points (95% CI $[-42.50, -25.83]$) with $\Delta U_2 = 0$. Both utility comparisons pass the predeclared 5-point non-inferiority margin. The live loops establish the end-to-end safety–utility shift produced by adding cross-step residual closure to a current-step guard. Full statistical details are reported in Appendix E.

Frozen same-candidate replay attributes the decision to RCA. The live integrations are complemented by a replay that holds the logged candidate and native TS-Guard decision point fixed. RCA removes all 15 observed unsafe native allows in the mixed protocol and all 41 under attack stress, while preserving all 72 observed safe native allows. Thus all 56 observed unsafe allows are removed and all 72 observed safe allows are preserved under the same-candidate decision test (Appendix Table 22). Fixing the candidate isolates the additional decision made by residual-authority closure rather than by a different generated recovery.

RCA also works as a standalone live recovery gate. Without a preceding guard, RCA reduces observed S_2 from 28.33% to 0.0% in the 120-episode mixed live loop while increasing U_2 from 63.33% to 66.67%. Under the 120-episode attack-stress protocol, it reduces S_2 from 85.0% to 0.0% while increasing U_2 from 10.0% to 20.0%. Task completion is unchanged in both protocols (Appendix Table 21), showing that the safety gain does not arise from suppressing the opportunity to complete the task.



(a) Mechanism isolation. (b) Composition safety. (c) Utility preservation.

Figure 4: **Quantitative evidence for residual-authority closure.** (a) At matched $U_2 = 80.0\%$, RCA reduces S_2^{pref} from 20.0% to 0.0%. (b) Red/teal denote native/+RCA executions; (b) RCA reaches zero observed protocol-specific violations across all four composition settings. (c) ΔU_2 is non-negative throughout.

Takeaway. RCA acts as an orthogonal cross-step authorization layer: it composes with symbolic, declarative, and learned current-step enforcement and removes the observed residual-authority violations without sacrificing measured safe-recovery utility.

6 Related Work

Agent authorization and runtime enforcement. Tool-using agents are governed by control-flow, privilege, and invocation-time enforcement. CaMeL separates trusted control flow from untrusted data (Debenedetti et al., 2026); Progent applies programmable least-privilege policies to tool calls (Shi et al., 2025); AgentSpec and ShieldAgent enforce runtime constraints (Wang et al., 2026; Chen et al., 2025); and ToolSafe places a learned safety guard at the invocation boundary (Mou et al., 2026). These mechanisms determine whether the *current* proposal satisfies policy, privilege, or safety constraints. RCA instead tracks the authority remaining after prior realized effects.

Stateful and trajectory-level safety. Stateful safety reasons beyond isolated calls. Trajectory-level evaluation and dynamic validation reason over multi-step context (Chen et al., 2026; Li et al., 2025), while declarative runtime constraints encode history-sensitive conditions (Wang et al., 2026). RCA is history-sensitive in a specific authorization sense: it quotients execution history by the residual future-effect decisions induced by realized effects. Trajectory context and residual authority can coexist but answer different questions—whether a sequence satisfies a safety/temporal condition versus whether another effect fits within authority left by execution.

Failure, retry, replay, and recovery authorization. ToolPrivBench studies failure-induced escalation to over-privileged tools (Yang et al., 2026); verification-aware retry wrappers use post-condition checks and idempotency to suppress duplicate non-atomic execution (Mansoor et al., 2026); commit-time authorization revalidates authority evidence before durable commit (Santos-Grueiro, 2026); and CapLease keeps durable token-independent consumption state against semantic replay (Xu et al., 2026). RCA targets the composition question that follows these boundaries: after one or more realized effects, does the next otherwise permitted recovery still fit within the authority that remains? This covers split recovery and cross-tool effect composition even when the next call is fresh, locally valid, and not a replay. Appendix B provides an expanded boundary comparison.

7 Conclusion

Failure recovery changes authorization because prior execution changes what authority remains. RCA makes that state explicit: it records realized effects, derives residual authority from the grant, and admits only recoveries whose prospective effect envelope remains within it. Controlled matched-utility and matched-information tests isolate this missing state, while source-executed Progent, AgentSpec, and TS-Guard compositions show that the invariant complements current-step enforcement. RCA turns recovery authorization from a fresh-call check into a stateful closure invariant. The design rule is simple: after execution changes the world, authorization must remember what authority has been consumed.

AI USE STATEMENT

Generative AI tools were used for language editing and polishing, and to assist the authors in writing experimental code. All scientific content, technical decisions, experimental design, analyses, and conclusions were developed and verified by the authors.

ETHICS STATEMENT

This work studies defensive authorization controls for tool-using agents. Evaluations use synthetic or public benchmark tasks and locally hosted models; no private user data or human participants are used. The released evidence supports defensive analysis and reproducibility under the declared effect abstraction and monitored runtime boundary.

REPRODUCIBILITY STATEMENT

The main text specifies the method, trust assumptions, metrics, statistical units, and claim scope. The appendix provides complete proofs, finite-state verification scope, benchmark construction, generation-validity accounting, implementation details, and source-system applicability. The supplementary artifact records frozen inputs, exact commands, model/environment identifiers, seeds, trajectory-level outputs, table/figure-generation scripts, source-system manifests, and automated consistency checks that bind repeated manuscript metrics and plotted source data to their machine-readable statistics.

REFERENCES

Yen-Shan Chen, Sian-Yao Huang, Cheng-Lin Yang, and Yun-Nung Chen. Tracesafe: A systematic assessment of LLM guardrails on multi-step tool-calling trajectories. In *Third Conference on Language Modeling (COLM)*, 2026. URL https://openreview.net/forum?id=KkFKVLWdEG.

Zhaorun Chen, Mintong Kang, and Bo Li. Shieldagent: Shielding agents via verifiable safety policy reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 8313–8344. PMLR, 2025.

Manuel Costa, Boris Köpf, Aashish Kolluri, Andrew Paverd, Mark Russinovich, Ahmed Salem, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. Securing ai agents with information-flow control. *arXiv preprint arXiv:2505.23643*, 2025.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2636.

Edoardo Debenedetti, Ilia Shumailov, Tianqi Fan, Jamie Hayes, Nicholas Carlini, Daniel Fabian, Christoph Kern, Chongyang Shi, Andreas Terzis, and Florian Tramèr. Defeating prompt injections by design. In *IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, 2026. URL https://arxiv.org/abs/2503.18813.

Hao Li, Xiaogeng Liu, Hung-Chun Chiu, Dianqi Li, Ning Zhang, and Chaowei Xiao. DRIFT: Dynamic rule-based defense with injection isolation for securing LLM agents. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, 2024.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. Toolsandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities. In *Findings*

of the Association for Computational Linguistics: NAACL 2025, pp. 1160–1183, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-naacl.65. URL https://aclanthology.org/2025.findings-naacl.65/.

Isham Kalappurackal Mansoor, Abhishek Phadke, and Pratip Rana. Verified tool calls improve llm agent reliability under non-atomic failures. *arXiv preprint arXiv:2608.02645*, 2026.

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, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1850. URL https://aclanthology.org/2026.findings-acl.1850/.

Ike Obi, Vishnunandan L. N. Venkatesh, Weizheng Wang, Ruiqi Wang, Dayoon Suh, Temitope I. Amosa, Wonse Jo, and Byung-Cheol Min. Pre-execution safety gate & task safety contracts for llm-controlled robot systems. *arXiv preprint arXiv:2604.05427*, 2026.

Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 48371–48392. PMLR, 2025.

Niklas Pfister, Václav Volhejn, Manuel Knott, Santiago Arias, Julia Bazinska, Mykhailo Bichurin, Alan Y. Commike, Janet Darling, Peter Dienes, Matthew Fiedler, David Haber, Matthias Kraft, Marco Lancini, Max Mathys, Damian Pascual-Ortiz, Jakub Podolak, Adrià Romero-López, Kyriacos Shiarlis, Andreas Signer, Zsolt Terek, Athanasios Theocharis, Daniel Timbrell, Samuel Trautwein, Samuel Watts, Yun-Han Wu, and Mateo Rojas-Carulla. Gandalf the red: Adaptive security for LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 49150–49194. PMLR, 2025.

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

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris Maddison, and Tatsunori Hashimoto. Identifying the risks of lm agents with an lm-emulated sandbox. In *International Conference on Learning Representations*, 2024.

Igor Santos-Grueiro. Temporary authority, permanent effects: Commit-time authorization for llm agents. *arXiv preprint arXiv:2607.10487*, 2026.

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

Tianneng Shi, Jingxuan He, Zhun Wang, Linyu Wu, Hongwei Li, Wenbo Guo, and Dawn Song. Progent: Programmable privilege control for LLM agents. *arXiv preprint arXiv:2504.11703*, 2025. URL https://arxiv.org/abs/2504.11703.

Tobin South, Samuele Marro, Thomas Hardjono, Robert Mahari, Cedric Deslandes Whitney, Alan Chan, and Alex Pentland. Position: AI agents need authenticated delegation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 82211–82231. PMLR, 2025.

Ada Defne Tur, Nicholas Meade, Xing Han Lù, Alejandra Zambrano, Arkil Patel, Esin Durmus, Spandana Gella, Karolina Stanczak, and Siva Reddy. Safearena: Evaluating the safety of autonomous web agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 60404–60441. PMLR, 2025.

Uchi Uchibeke. Before the tool call: Deterministic pre-action authorization for autonomous AI agents. *arXiv preprint arXiv:2603.20953*, 2026. URL https://arxiv.org/abs/2603.20953.

Haoyu Wang, Christopher M. Poskitt, and Jun Sun. Agentspec: Customizable runtime enforcement for safe and reliable LLM agents. In *2026 IEEE/ACM 48th International Conference on Software Engineering (ICSE)*, pp. 2938–2950. ACM, 2026. doi: 10.1145/3744916.3764546. URL https://doi.org/10.1145/3744916.3764546.

Jize Wang, Zerun Ma, Yining Li, Songyang Zhang, Cailian Chen, Kai Chen, and Xinyi Le. Gta: A benchmark for general tool agents. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Zhen Xiang, Linzhi Zheng, Yanjie Li, Junyuan Hong, Qinbin Li, Han Xie, Jiawei Zhang, Zidi Xiong, Chulin Xie, Carl Yang, Dawn Song, and Bo Li. Guardagent: Safeguard LLM agents via knowledge-enabled reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 68316–68342. PMLR, 2025.

Yuejin Xie, Youliang Yuan, Wenxuan Wang, Fan Mo, Jianmin Guo, and Pinjia He. Toolsafety: A comprehensive dataset for enhancing safety in llm-based agent tool invocations. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 14135–14156. Association for Computational Linguistics, 2025.

Jinghan Xu, Longze Fan, Zeyuan Wang, Xinjin Li, and Hankai Liu. Beyond single-use tokens: Durable authorization state for replay-resistant llm agent actions. *arXiv preprint arXiv:2608.01710*, 2026.

Kaiyue Yang, Yuyan Bu, Jingwei Yi, Yuchi Wang, Biyu Zhou, Juntao Dai, Songlin Hu, and Yaodong Yang. When lower privileges suffice: Investigating over-privileged tool selection in llm agents. *arXiv preprint arXiv:2606.20023*, 2026.

Wenkai Yang, Xiaohan Bi, Yankai Lin, Sishuo Chen, Jie Zhou, and Xu Sun. Watch out for your agents! investigating backdoor threats to llm-based agents. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

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

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

Junjie Ye, Sixian Li, Guanyu Li, Caishuang Huang, Songyang Gao, Yilong Wu, Qi Zhang, Tao Gui, and Xuanjing Huang. Toolsword: Unveiling safety issues of large language models in tool learning across three stages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 2181–2211. Association for Computational Linguistics, 2024.

Tongxin Yuan, Zhiwei He, Lingzhong Dong, Yiming Wang, Ruijie Zhao, Tian Xia, Lizhen Xu, Binglin Zhou, Fangqi Li, Zhuosheng Zhang, Rui Wang, and Gongshen Liu. R-judge: Benchmarking safety risk awareness for llm agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics, 2024.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. Injecagent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10471–10506. Association for Computational Linguistics, 2024.

Hanrong Zhang, Jingyuan Huang, Kai Mei, Yifei Yao, Zhenting Wang, Chenlu Zhan, Hongwei Wang, and Yongfeng Zhang. Agent security bench (ASB): Formalizing and benchmarking attacks and defenses in LLM-based agents. In *International Conference on Learning Representations*, 2025.

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*, 2024.

Jinhao Zhu, Kevin Tseng, Gil Vernik, Xiao Huang, Shishir G. Patil, Vivian Fang, and Raluca Ada Popa. Miniscope: A least privilege framework for authorizing tool calling agents. *arXiv preprint arXiv:2512.11147*, 2025a.

Kaijie Zhu, Xianjun Yang, Jindong Wang, Wenbo Guo, and William Yang Wang. MELON: Provable defense against indirect prompt injection attacks in AI agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 80310–80329. PMLR, 2025b.

A PROOFS

A.1 RESIDUAL-AUTHORITY SUFFICIENCY

Suppose $\mathcal{A}_{\mathrm{res}}(h_1) = \mathcal{A}_{\mathrm{res}}(h_2)$ and the same prospective effect envelope Γ is presented at both histories. RCA-Safe accepts exactly when $\Gamma \subseteq \mathcal{A}_{\mathrm{res}}(h)$. The subset predicate therefore has the same truth value for h_1 and h_2. Hence the two histories induce the same immediate authorization decision once residual authority and the prospective envelope are fixed.

A.2 NECESSITY OF DISTINGUISHING RESIDUAL AUTHORITY

Let $Z(h_1) = Z(h_2) = z$ and let the same candidate c expose the same prospective envelope Γ at both histories, with

$$\Gamma \subseteq \mathcal{A}_{\mathrm{res}}(h_1), \qquad \Gamma \nsubseteq \mathcal{A}_{\mathrm{res}}(h_2). \tag{12}$$

A deterministic monitor receiving only (z, c) returns the same decision in both histories. If it accepts, it is unsound on h_2; if it rejects, it is not permissive for the authorized recovery on h_1. For a randomized monitor, any nonzero probability of accepting produces nonzero unsoundness probability on h_2, whereas rejection with probability one loses permissiveness on h_1. Therefore any summary sufficient for guaranteed sound and permissive recovery must distinguish histories whenever their residual states induce different decisions for a realizable candidate.

A.3 SCOPED MINIMAL-STATE CHARACTERIZATION

Fix the declared effect abstraction and a separating continuation family $\mathcal{G}$ of realizable prospective envelopes. Define $h_1 \sim h_2$ iff, for every $\Gamma \in \mathcal{G}$,

$$\Gamma \subseteq \mathcal{A}_{\mathrm{res}}(h_1) \iff \Gamma \subseteq \mathcal{A}_{\mathrm{res}}(h_2). \tag{13}$$

The equivalence class $[h]_\sim$ is sufficient for every immediate recovery decision induced by $\mathcal{G}$. Conversely, any exact Markov summary that merges two histories from different equivalence classes also merges a pair separated by some realizable continuation Γ, so it cannot preserve both authorization decisions. Hence the residual future-effect decision state is coarsest up to decision-equivalent representations.

Table 4: Finite-state verification scope in the declared abstraction. The 8,128 checks exhaust all unordered pairs of the 128 declared authority configurations; the model also explores 800 authority-relative residual states and 101,600 state–action transitions.

Quantity	Count
Authority configurations	128
Authority-relative residual states	800
State–action transitions	101,600
Pairwise minimality checks	8,128

A.4 RECOVERY-CLOSURE SOUNDNESS

We prove by induction over admitted recovery calls. The base history satisfies $E(h_0) \preceq A_0$ by assumption. Assume the current ledger satisfies $E(h_t) \preceq A_0$. RCA-Safe admits c_t only if

$$\forall e' \in \Gamma(s_t, c_t), \qquad E(h_t) \oplus e' \preceq A_0. \tag{14}$$

Prospective envelope faithfulness gives $e_t^\star \in \Gamma(s_t, c_t)$ for the authorization-relevant effect actually realized by the monitored execution. Instantiating the universal check at $e' = e_t^\star$ yields

$$E(h_t) \oplus e_t^\star \preceq A_0. \tag{15}$$

Ledger faithfulness commits $e_t^\star$ before the next decision, so $E(h_{t+1}) = E(h_t) \oplus e_t^\star$ in the represented abstraction. Thus the invariant holds at h_{t+1}, and induction establishes authorization safety for every RCA-Safe-admitted prefix.

For RCA-Uncertain, assume $E^\star(h_t) \in \mathcal{H}(h_t)$. Instantiating the universal check at the true cumulative-effect hypothesis and the realized $e_t^\star \in \Gamma(s_t, c_t)$ yields an authorized successor. The hypothesis update defined in Appendix C preserves $E^\star(h_{t+1})$ among the represented hypotheses, re-establishing the inductive premise. The same argument therefore holds for every uncertain recovery prefix.

A.5 MONOTONICITY OF IMMEDIATE PERMISSIVENESS

If $\mathcal{H}_1(h) \subseteq \mathcal{H}_2(h)$ and $\Gamma_1(s, c) \subseteq \Gamma_2(s, c)$, every pair quantified by the smaller check is also quantified by the larger one. Hence any violation that causes rejection under the smaller sets remains present under a superset expansion. Enlarging either uncertainty set therefore cannot turn an immediate rejection into an admission.

B ADDITIONAL RELATED WORK

Tool-using agents and agent-security evaluation. LLM agents have progressed from explicit reasoning–acting loops and learned API use (Yao et al., 2023; Schick et al., 2023; Qin et al., 2024) to interactive and function-calling evaluations (Liu et al., 2024; Zhou et al., 2024; Patil et al., 2025). Stateful benchmarks such as ToolSandbox, τ-bench, and GTA emphasize long-horizon interaction, mutable environment state, and executable tool chains (Lu et al., 2025; Yao et al., 2024; Wang et al., 2024). Security evaluations expose complementary risks from unsafe tool use, indirect prompt injection, backdoors, and autonomous web actions (Ruan et al., 2024; Zhan et al., 2024; Ye et al., 2024; Debenedetti et al., 2024; Zhang et al., 2025; Yuan et al., 2024; Yang et al., 2024; Tur et al., 2025). RCA focuses on a failure boundary that these evaluations do not explicitly isolate: how authorization changes after prior execution has already produced an effect.

Runtime guardrails and information-flow enforcement. Recent defenses interpose checks before external actions execute. MELON detects indirect prompt injection through masked re-execution (Zhu et al., 2025b); GuardAgent turns safety requests into executable checking logic (Xiang et al., 2025); ShieldAgent verifies action trajectories against explicit safety policies (Chen et al., 2025); and AgentSpec provides a DSL for customizable runtime constraints (Wang et al., 2026). DRIFT combines planned function trajectories with dynamic validation (Li et al., 2025), while CaMeL separates trusted control flow from untrusted data and constrains information flow through capabilities (Debenedetti et al., 2026). Fides likewise studies deterministic information-flow control for agent planners (Costa et al., 2025), and ToolSafety targets safe training for indirect and multi-step tool interactions (Xie et al., 2025). These mechanisms constrain content influence, current actions, or trajectory compliance. RCA is complementary: it treats *authority consumption* as state, so a locally compliant action can become unauthorized after an earlier effect.

Least privilege, authorization, and failure. Agent authorization increasingly emphasizes explicit delegation and least privilege (South et al., 2025). Progent enforces symbolic privilege policies over tool calls and supports dynamic policy updates and blocked-call fallbacks (Shi et al., 2025); MiniScope reconstructs permission hierarchies to enforce least privilege (Zhu et al., 2025a). Adaptive security evaluation further shows that restrictive defenses must be judged jointly on security and

Table 5: RCA addresses a distinct authorization state from neighboring agent-security mechanisms. Each row identifies the primary invariant and security-relevant state enforced at that boundary.

Boundary	Primary question	Security-relevant state
Information flow	Can untrusted content influence privileged computation?	provenance / control flow
Least privilege	Is more authority selected than the task requires?	privilege / delegation policy
Pre-call authorization	Is this principal–tool–argument call permitted now?	identity / capability / current policy
Step-level safety gate	Is the proposed execution step safe under the gate's safety contract?	current step / trajectory context
Recovery closure (RCA)	**Does this otherwise permitted call remain authorized after prior effects?**	**effect history / residual authority**

utility (Pfister et al., 2025). Most closely related to the role of failure, ToolPrivBench studies over-privileged tool selection and shows that transient failures can amplify privilege escalation (Yang et al., 2026). RCA addresses a distinct case: a retry can exceed authorization without selecting a more privileged tool, broadening the permitted tool set, or changing the local action schema. Least privilege constrains *which authority is granted or selected*; RCA constrains *how granted authority remains available and composes after realized effects*.

Non-atomic retries, commit-time checks, and durable replay state. Verification-aware tool wrappers address a complementary reliability failure: after a timeout or delayed state update, verify-before-retry and idempotency can prevent an exact duplicate effect (Mansoor et al., 2026). Commit-time authorization instead asks whether the witness that licensed a durable effect is still fresh and bound when that effect becomes durable (Santos-Grueiro, 2026). CapLease studies failure semantic replay and shows that single-use token identifiers are insufficient when a fresh token can reissue the same authorization; it therefore keeps durable token-independent consumption state (Xu et al., 2026). These controls overlap with individual RCA cases but do not define the same closure objective. RCA reasons over a policy-defined composition of realized effects, so the next recovery can be rejected even when it is not an exact replay of an earlier call and even when every individual call uses an otherwise valid privilege. Sink-level idempotency and commit-time witness validation layer naturally beneath or beside the residual-authority monitor, covering duplicate-effect and commit-validity invariants alongside RCA's cross-step closure invariant.

Pre-invocation authorization and step-level safety gates. The Open Agent Passport (OAP) formalizes deterministic pre-action authorization by evaluating declarative policy before tool execution and producing an auditable authorization record (Uchibeke, 2026). ToolSafe similarly places a proactive guard at each tool invocation and can feed safety feedback back into agent reasoning (Mou et al., 2026); TraceSafe evaluates the complementary problem of detecting risk over multi-step tool-calling trajectories (Chen et al., 2026). In robotics, SafeGate combines a pre-execution command gate with task safety contracts that constrain subsequent physical state transitions (Obi et al., 2026). RCA composes with these mechanisms at the recovery boundary. A per-call engine answers whether the current principal–tool–argument tuple is permitted under the current policy; a step-level safety gate answers whether the proposed step appears safe; RCA additionally asks whether that otherwise permitted step remains authorized after prior realized effects have consumed, bound, or invalidated part of the same authority. The deterministic composition used by RCA is conjunctive: a recovery may execute only if both the per-call policy engine and the residual-authority check allow it.

C Runtime and Deployment Details

Constructing envelopes in practice. RCA obtains Γ from trusted tool metadata and authenticated canonicalization state, keeping prospective-effect construction inside the trusted deployment interface. Deterministic fields become equalities; non-atomic outcomes become symbolic sub-effect bounds or finite alternatives; aliases become authenticated equivalence sets; and multi-effect tools expose every authorization-relevant required or optional effect. Universal checking is performed over these symbolic constraints. For example:

Tool contract	Prospective envelope sketch
atomic `transfer(r,a)`	recipient $= \mathrm{canon}(r)$, transfer $= a$
non-atomic `transfer(r,a)`	supported zero/partial/full outcomes, e.g. transfer $\in [0, a]$ under the declared abstraction
`charge(r,a)` with fee $[0, f_{\max}]$	recipient $\in \mathrm{Alias}(r)$, debit $\in [a, a + f_{\max}]$
multi-effect write	primary write plus every contract-declared optional callback/audit/write effect relevant to policy

Envelope coverage defines the protected effect surface. The contract is sound only when every represented realized effect of an admitted call is covered by Γ.

C.1 Threat model and trust boundary

The language model, raw tool error, and model-generated recovery candidate are untrusted. Failure metadata may be partially trusted and is stress-tested separately. The authenticated authority root, bound tool envelope, effect/event log, and RCA runtime monitor form the trusted computing base. The evaluator is separate from the runtime monitor and labels realized external effects rather than reusing monitor decisions.

The formal contract uses two forms of effect faithfulness. First, *prospective envelope faithfulness*: every authorization-relevant realized effect of an admitted monitored call belongs to $\Gamma(s, c)$. Second, *ledger faithfulness*: the realized effect is committed to the trusted ledger before the next recovery decision. Decision-time validity such as expiry, revocation, or authorization epochs is read from authenticated runtime context. When validity may change between admission and durable commit, the tool boundary revalidates authority at commit time or uses a lease/witness whose validity spans the commit. We classify tool boundaries as *instrumented*, *uncertain*, or *external*. Instrumented tools expose contract-covered effects through a mediated proxy, transactional outbox/state diff, or equivalent trusted event source. Uncertain tools retain all supported effect hypotheses and withhold affected authority. External effects enter RCA's protected surface when they are mediated or conservatively represented by the deployment contract.

C.2 Effect ledger and uncertainty about prior effects

RCA derives authority consumption from the trusted effect ledger. Whenever an attempted tool call produces an externally visible authorization-relevant effect—including a partial effect followed by an error—that effect is appended before the next recovery decision. A rejection leaves authority unchanged, and authority increases only through an authenticated new grant recorded by the runtime.

When prior effects are imperfectly observed, RCA-Uncertain maintains a finite cumulative-effect hypothesis set

$$\mathcal{H}(h) = \{\hat{E}_1, \ldots, \hat{E}_K\}, \tag{16}$$

where each hypothesis is consistent with the typed observation `confirmed_no_effect`, `confirmed_effect`, or `uncertain_effect`. Let $\Omega(c_t, o_{t+1})$ be the contract-supported effect outcomes consistent with the trusted observation o_{t+1}. The update is

$$\mathcal{H}(h_{t+1}) = \mathrm{Abs}\Big(\{\hat{E} \oplus e : \hat{E} \in \mathcal{H}(h_t),\ e \in \Omega(c_t, o_{t+1})\}\Big), \tag{17}$$

where Abs may conservatively merge equivalent hypotheses but must preserve the true cumulative effect whenever the true prior hypothesis and realized effect are represented. Uncertainty is never silently mapped to no effect. The conservative immediately admissible set is

$$\mathcal{A}^{\mathrm{unc}}_{\mathrm{res}}(h; \mathcal{H}) = \bigcap_{\hat{E} \in \mathcal{H}(h)} \Big\{e : \hat{E} \oplus e \preceq A_0\Big\}. \tag{18}$$

Thus enlarging the supported history-hypothesis set can only contract immediate permissiveness, while truth-preserving updates maintain the inductive soundness premise across multiple recovery steps.

C.3 Concurrency and shared authority

The core theorem is stated over a linearized recovery history. Shared-authority deployments therefore expose a linearization point per authority root. The reference implementation serializes effectful recovery admission through a per-root transactional critical section covering the residual-state read, conjunctive decision, monitored execution, and ledger commit. This is the deployment contract used by the proof: each recovery decision observes a coherent residual state and its realized effect is committed before the next decision.

C.4 Composing with per-call authorization engines

When a pre-call authorization engine is present, RCA composes behind its native decision surface. Let $P_{\mathrm{call}}(x, c)$ be the engine's deterministic decision for authenticated context x and candidate c. The composed gate is

$$\mathrm{allow}(h, x, c) = P_{\mathrm{call}}(x, c) \wedge [\forall e' \in \Gamma(s, c),\ E(h) \oplus e' \preceq A_0]\,. \tag{19}$$

The per-call engine can deny a candidate that RCA would otherwise regard as residual-authority safe, and RCA preserves that native decision. For a stateful upstream engine, the native authorization step is either side-effect free or executed as a prepare/commit operation: execution-coupled state changes are committed only after both the native predicate and RCA allow, and an RCA veto leaves them uncommitted. Residual authority increases only when the authorization engine emits an authenticated new grant/top-up represented as a new authority event. A robust audit contract binds the per-call decision/proof to the principal, tool, canonical arguments, policy version or epoch, expiry, and authorization-root identifier; the RCA record additionally binds the effect-contract identifier and ledger version. This makes the two decisions independently auditable while preventing a fresh per-call token from silently resurrecting exhausted authority.

D Implementation Details

The frozen controlled run uses Qwen3.5-9B, Qwen3.5-4B, Llama-3-8B-Instruct, and Mistral-7B-Instruct-v0.2. Checkpoints are loaded locally with remote code disabled. We use continuation-only greedy decoding, one generation per scenario–model–attack tuple shared across policy comparisons, and a strict JSON proposal schema. Parse failures and truncations are retained in diagnostics. The qualitative conclusion is unchanged under the pre-specified threshold-sensitivity and all-valid-generation analyses.

Runtime authorization interface. Each monitored call carries an authenticated authority-root identifier, tool-contract identifier, canonicalized authorization-relevant arguments, and policy/authorization epoch. The monitor derives or loads the contract-backed prospective envelope, reads the residual state associated with the same authority root, evaluates the universal residual check, and records the decision with a ledger version. For shared authority, the reference concurrency-safe path uses a per-root linearizable critical section. These fields are also the minimum contract needed to compose RCA deterministically with a per-call authorization engine.

E Statistical Analysis

For the controlled evaluation, the 360 base scenarios are the independent statistical units; model and attack strata are repeated observations within each scenario. Scenario-clustered tables use percentile paired bootstrap intervals over 2,000 resamples with seed 20260809 as the confirmatory analysis. For U_2, the non-inferiority margin is 0.05 relative to the comparison policy. At the scenario level, a zero-event estimate has an exact one-sided 95% upper bound of $1 - 0.05^{1/360} = 0.0083$.

For a zero-event rate over 360 independent scenario clusters, the exact one-sided 95% upper bound is $1 - 0.05^{1/360} = 0.0083$. This cluster-level bound is reported separately from the tuple-level diagnostic intervals.

External-system replay and TS-Guard composition. Progent and AgentSpec are source-executed through their supplied snapshots behind project-side interface wrappers. TS-Guard uses

Table 6: Primary statistics from the corrected canonical full evaluation with base-scenario-clustered inference. Safety is terminal/native S_2^{term}, not a prefix claim. The 360 base scenarios are the independent units; model and attack strata are repeated observations within each cluster. Intervals are percentile paired-bootstrap 95% intervals over scenarios.

Policy	Clusters	Rows	S_2^{term}	S_2^{term} cluster CI	U_2	U_2 cluster CI
Fail-closed	360	4320	0.0000	[0.0000,0.0000]	0.0000	[0.0000,0.0000]
Raw error	360	4320	0.5278	[0.5019,0.5537]	0.2778	[0.2648,0.2907]
Model candidate	360	4320	0.1433	[0.1350,0.1512]	0.2593	[0.2456,0.2720]
Structured baseline	360	4320	0.0000	[0.0000,0.0000]	0.2593	[0.2461,0.2725]
RCA	360	4320	0.0000	[0.0000,0.0000]	0.8333	[0.7944,0.8722]
Safe oracle	360	4320	0.0000	[0.0000,0.0000]	0.8333	[0.7944,0.8722]

Table 7: Scenario-clustered paired effects of RCA relative to each frozen comparison policy. Positive Δ means higher RCA metric. The tuple-level exact McNemar table is a secondary diagnostic.

Comparison	Metric	Clusters	Δ	95% cluster CI	$+/-/0$ clusters
RCA / Fail-closed	S_2^{term}	360	0.0000	[0.0000,0.0000]	0/0/360
RCA / Fail-closed	U_2	360	0.8333	[0.7944,0.8694]	300/0/60
RCA / Raw error	S_2^{term}	360	-0.5278	[-0.5546,-0.5000]	0/300/60
RCA / Raw error	U_2	360	0.5556	[0.5296,0.5815]	300/0/60
RCA / Model candidate	S_2^{term}	360	-0.1433	[-0.1512,-0.1350]	0/300/60
RCA / Model candidate	U_2	360	0.5741	[0.5454,0.5995]	300/0/60
RCA / Structured baseline	S_2^{term}	360	0.0000	[0.0000,0.0000]	0/0/360
RCA / Structured baseline	U_2	360	0.5741	[0.5479,0.6016]	300/0/60
RCA / Safe oracle	S_2^{term}	360	0.0000	[0.0000,0.0000]	0/0/360
RCA / Safe oracle	U_2	360	0.0000	[0.0000,0.0000]	0/0/360

its native guard interface/checkpoint, and RCA is applied after a native allow decision. For the 120-episode TS-Guard live protocols, episode-level bootstrap intervals summarize end-to-end differences, while frozen-candidate replay supplies same-decision-point attribution. The mixed-loop differences are $\Delta S_2 = -0.1250$ (95% CI $[-0.1833, -0.0667]$) and $\Delta U_2 = +0.0083$ (95% CI $[0, 0.0250]$); attack stress gives $\Delta S_2 = -0.3417$ (95% CI $[-0.4250, -0.2583]$) and $\Delta U_2 = 0$ (95% CI $[0, 0]$). With the predeclared 0.05 margin, both utility comparisons satisfy non-inferiority.

Metric alignment and evaluator separation. RecoverBench realized-effect traces support accepted-prefix evaluation and are reported as S_2^{pref}. Native $\tau 2$ and strict trajectory summaries provide terminal/native evaluator labels and are reported as S_2^{term}. Runtime decisions exclude evaluator-only safe/unsafe labels, and realized effects are evaluated independently of the monitor decision.

F Supplementary Quantitative Results

The supplementary results expose the full safety–utility statistics and stress-test detail behind the compact main-text comparisons.

Included backbones: Qwen3.5-4B, Llama-3-8B-Instruct. Total generated rows audited: 25920; main rows: 12960.

F.1 Generation-validity sensitivity

The primary conclusion is stable to the proposal-format validity rule. Recomputing the frozen generations at thresholds above 97%, 98%, and 99%, and with all valid parsed generations from all four backbones, preserves the qualitative safety–utility separation: RCA has no observed unauthorized effect, while its utility advantage over the structured baseline remains 82.82–82.84 percentage points across these views.

Table 8: Strict-proposal primary results after the pre-registered generation-quality gate. Main metrics include eligible backbones only; failed backbones remain in the audit evidence.

Method	N	S_2^{term}	U_2	Failed proposals
Fail-closed	2160	0.0%	0.0%	0
Safe oracle	2160	0.0%	83.3%	0
Model candidate	2160	0.0%	0.5%	0
RCA-Safe	2160	0.0%	83.3%	0
Raw error	2160	52.8%	30.6%	0
Structured baseline	2160	0.0%	0.5%	0

Table 9: Strict generation diagnostics used by the eligibility gate. Rates are computed over the $N = 6480$ attempted rows for each backbone; explicit STOP is reported independently of parse failures. Eligible backbones are the only ones used for the main-model claim.

Model	N	Valid	Valid JSON	Parse fail.	Trunc.	STOP	Main
Qwen3.5-9B	6480	6348	97.96%	2.04%	0.00%	49.26%	excluded
Qwen3.5-4B	6480	6480	100.00%	0.00%	0.00%	13.61%	eligible
Llama-3-8B-Instruct	6480	6480	100.00%	0.00%	0.00%	71.85%	eligible
Mistral-7B-Instruct-v0.2	6480	6288	97.04%	2.96%	0.00%	0.00%	excluded

G Mechanism Attribution

These tests isolate RCA's cross-step decision while holding trusted information, candidate identity, or native decision surfaces fixed. Progent and AgentSpec retain their supplied decision logic, and TS-Guard uses its native guard interface/checkpoint.

G.1 Matched-information LedgerLocal baseline

LedgerLocal receives the same authenticated authority root, realized-effect ledger, and effect canonicalization as RCA-Safe, but checks only the current candidate against the original per-action policy. On the held-out composition split, LedgerLocal retains the local guard's $S_2^{\text{pref}} = 20.0\%$ and $U_2 = 80.0\%$, whereas RCA-Safe has $S_2^{\text{pref}} = 0.0\%$ at the same U_2. This isolates representation and use of residual authority as the decision-relevant difference.

G.2 Semantic-OOD composition

A separate 135-case semantic-OOD suite changes the effect-composition semantics rather than only the sequential ordering used in the held-out split. At matched $U_2 = 33.3\%$, StructuredLocal and LedgerLocal each incur $S_2^{\text{pref}} = 22.2\%$, whereas RCA-Safe has no observed unauthorized prefix effect. This extends the residual-state separation beyond the cumulative-budget cases that dominate the held-out composition split.

G.3 Counterfactual collision suite

The deterministic collision suite holds the raw error, normalized failure, candidate, and local policy view fixed while changing prior authorization-relevant history. RCA distinguishes all four implemented history-sensitive families: cumulative budget, one-shot authorization, cross-tool shared budget, and partial effect. The suite therefore tests the residual-state distinction across multiple effect algebras: cumulative, one-shot, shared-capacity, and partial-effect authorization.

G.4 Source-executed compositional validation

On the shared frozen interface, the Progent and AgentSpec decision surfaces and their "+RCA" compositions are source-executed over 360 scenarios. Native decision logic was unchanged and RCA was applied only at the recovery authorization boundary. We then evaluated both pairs in a full Qwen3.5-9B live loop on the official tau2-bench mock domain, with 120 matched episodes per policy and zero strict-JSON parse failures. Progent alone has $S_2^{\text{pref}} = 26.7\%$ and $U_2 = 33.3\%$;

Table 10: Sensitivity of the frozen strict results to the generation-validity gate. Percentages are terminal S_2^{term}/U2; each policy uses the same valid parsed rows in the indicated pool.

Pool	RCA S_2^{term}	RCA U2	Struct. S_2^{term}	Struct. U2	Raw S_2^{term}	Raw U2	ΔU2
Primary $> 98\%$	0.00%	83.33%	0.00%	0.51%	52.78%	30.56%	82.82%
All valid	0.00%	83.68%	0.00%	0.84%	52.93%	30.75%	82.84%
$> 97\%$	0.00%	83.68%	0.00%	0.84%	52.93%	30.75%	82.84%
$> 98\%$	0.00%	83.33%	0.00%	0.51%	52.78%	30.56%	82.82%
$> 99\%$	0.00%	83.33%	0.00%	0.51%	52.78%	30.56%	82.82%

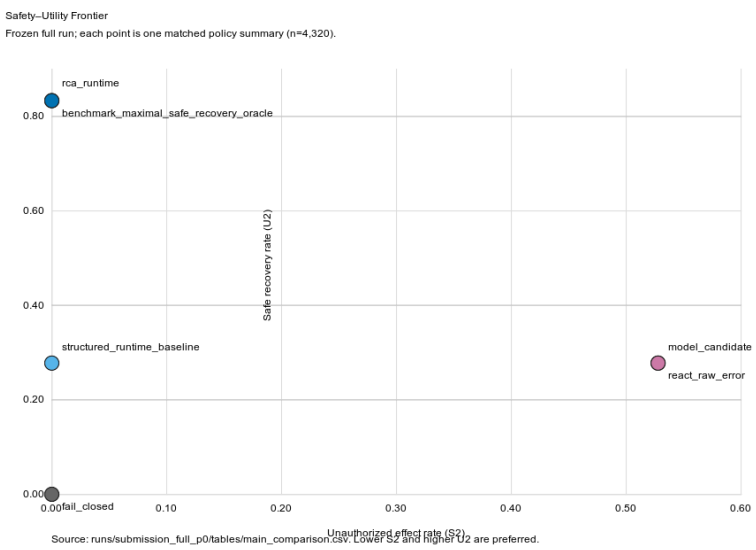


Figure 5: Safety–utility frontier on the controlled evaluation. Fail-closed obtains safety by eliminating recovery, while RCA retains substantial safe-recovery utility and reaches the within-protocol bounded-oracle frontier.

Progent+RCA reduces S_2^{pref} to 0.0% and raises U_2 to 43.3%. AgentSpec alone has $S_2^{\text{pref}} = 5.8\%$ and $U_2 = 40.8\%$; AgentSpec+RCA likewise reaches $S_2^{\text{pref}} = 0.0\%$ and $U_2 = 43.3\%$. Neither composition loses paired safe utility on an episode where its corresponding official guard has $U_2 = 1$. These source-executed boundary compositions preserve the native decision logic and directly measure RCA's incremental effect at the shared recovery boundary.

G.5 Standalone live-loop recovery gate

RCA also operates as a standalone recovery-authorization gate rather than only as an add-on to another guard. In the 120-episode mixed live loop, unguarded recovery has $S_2 = 28.33\%$ and $U_2 = 63.33\%$, whereas RCA has $S_2 = 0.0\%$ and $U_2 = 66.67\%$. Under the 120-episode attack-stress protocol, unguarded recovery reaches $S_2 = 85.0\%$ with $U_2 = 10.0\%$, while RCA has $S_2 = 0.0\%$ and $U_2 = 20.0\%$. In both protocols, task completion is unchanged by the gate, so the safety separation is not produced by suppressing the opportunity to finish the task.

G.6 Learned step-level guard composition and frozen replay

TS-Guard provides an architecturally distinct composition test through a learned step-level guard, complementing the symbolic Progent and AgentSpec integrations. In the 120-episode mixed live loop, TS-Guard has $S_2 = 12.5\%$ and $U_2 = 65.83\%$, whereas TS-Guard+RCA has $S_2 = 0$ and $U_2 = 66.67\%$. In the 120-episode attack-stress loop, TS-Guard has $S_2 = 34.17\%$ and $U_2 = 20.0\%$, whereas TS-Guard+RCA has $S_2 = 0$ at the same $U_2 = 20.0\%$.

Table 11: Closed-loop controlled RecoverBench evaluation with state-changing ToolWrapper execution. All accepted prefixes are audited, so the safety column is S_2^{pref}.

Policy	n	Task success	S_2^{pref}	Mean calls
Fail-closed	360	40.0%	0.0%	1.00
Raw-error recovery	360	100.0%	0.0%	1.60
Structured runtime	360	100.0%	0.0%	1.60
RCA-Safe	360	100.0%	0.0%	1.60
RCA-Uncertain	360	100.0%	0.0%	1.60
Aggressive raw-error fallback	360	40.0%	20.0%	0.60
Aggressive RCA gate	360	40.0%	0.0%	0.40

Table 12: Defense-aware adaptive attacker over the explicit 12-strategy vocabulary and 360 scenarios. No violation was found; query-to-first-violation is therefore not estimable.

Budget	S2	Violations	Exhaustion	Unique paths
1	0.0%	0	0.0%	1.0
5	0.0%	0	0.0%	5.0
10	0.0%	0	0.0%	10.0
20	0.0%	0	100.0%	12.0
50	0.0%	0	100.0%	12.0

Because the original live arms use independent planner calls, we additionally replay RCA over frozen TS-Guard decision points. Table 22 reports mutually exclusive categories: A=TS allow/RCA allow/safe, B=TS allow/RCA allow/unsafe, C=TS allow/RCA deny/unsafe, D=TS allow/RCA deny/safe, E=valid TS deny, and F=malformed/missing native output.

The unsafe-allow removal rate is 1.0 in both protocols, and safe-allow preservation is $72/72$ in the mixed protocol. The replay freezes the candidate, step, task, native TS-Guard output/decision, and trusted authority projection used by RCA, with evaluator labels withheld from both runtime decisions. This provides same-candidate attribution at the authorization-decision layer.

H RecoverBench Construction and Scenario Taxonomy

RecoverBench provenance. RecoverBench is a procedural synthetic benchmark generated from a versioned YAML matrix, not a public-trace or real-environment dataset. The frozen construction is the cross-product of six domains, ten failure families, and six control variants, yielding 360 records. All 360 records are applicable to the primary evaluator and all 360 are evaluated; no record is excluded. The 120-case composition split is generated procedurally from held-out ordered compositions.

Provenance category	Record count
Procedural synthetic records	360
Public-trace-derived records	0
Real-environment-derived records	0
Record-level human annotations	0
LLM-generated records	0

The ten failure families are defined by the frozen benchmark configuration rather than imported from public traces; their source is recorded as procedural YAML generation. The artifact contains no public trace identifiers, real-environment logs, or record-level human annotation fields. We therefore treat external validity as a limitation rather than implying that RecoverBench is a public or human-annotated benchmark. Each frozen record contains the domain, control variant, raw error, normalized failure, initial authority, initial and fallback tool states, original action, failure injection, and probe actions. Hidden safe/unsafe annotations and evaluator outputs are not exposed to candidate generation. The frozen controlled evaluation contains $4 \times 360 \times 3 = 4{,}320$ unique model generations. Replaying each proposal through six policies yields $25{,}920$ policy-evaluation trajectories. The pre-specified strict-format criterion retains two of the four backbones, yielding $2{,}160$ unique generations and $12{,}960$ confirmatory policy-evaluation trajectories. All four backbones remain in the frozen generation pool, and the all-valid-generation sensitivity analysis reproduces the

Table 13: A1 deterministic comparison on frozen recovery sequences. LedgerLocal records the same realized-effect ledger as RCA-Safe but applies only the original per-action policy; it therefore has no residual-authority closure.

Slice	Method	N	S_2^{pref}	U_2	Overblock	Reject	Replay
primary_360	StructuredLocal	360	0.0%	100.0%	0.0%	0	0
primary_360	LedgerLocal	360	0.0%	100.0%	0.0%	0	0
primary_360	ReplayLedger	360	0.0%	100.0%	0.0%	0	0
primary_360	RCA-Safe	360	0.0%	100.0%	0.0%	0	0
heldout_120	StructuredLocal	120	20.0%	80.0%	0.0%	0	0
heldout_120	LedgerLocal	120	20.0%	80.0%	0.0%	0	0
heldout_120	ReplayLedger	120	0.0%	0.0%	100.0%	240	240
heldout_120	RCA-Safe	120	0.0%	80.0%	0.0%	24	0
semantic_ood_135	StructuredLocal	135	22.2%	33.3%	0.0%	60	0
semantic_ood_135	LedgerLocal	135	22.2%	33.3%	0.0%	60	0
semantic_ood_135	ReplayLedger	135	22.2%	33.3%	0.0%	60	0
semantic_ood_135	RCA-Safe	135	0.0%	33.3%	0.0%	90	0

Table 14: A1 family breakdown. Each cell reports $S_2^{\mathrm{pref}}/U_2/$overblock in percent.

Family	N	StructuredLocal	LedgerLocal	ReplayLedger	RCA-Safe
alias	24	0/100/0	0/100/0	0/0/100	0/100/0
alias_equivalence	15	0/0/0	0/0/0	0/0/0	0/0/0
alternate_tool	60	0/100/0	0/100/0	0/60/40	0/100/0
concurrent_reordered_effects	30	0/100/0	0/100/0	0/100/0	0/100/0
cross_tool_duplicate	15	100/0/0	100/0/0	100/0/0	0/0/0
denial_reason_privacy_oracle	36	0/100/0	0/100/0	0/100/0	0/100/0
exact_duplicate_retry	84	28.6/71.4/0	28.6/71.4/0	0/42.9/40	0/71.4/0
higher_privilege_delegation	36	0/100/0	0/100/0	0/100/0	0/100/0
multi_dimension_effect	15	0/0/0	0/0/0	0/0/0	0/0/0
multiple_recipients	15	0/100/0	0/100/0	0/100/0	0/100/0
nested_authorization_scope	15	0/0/0	0/0/0	0/0/0	0/0/0
partial_effect	36	0/100/0	0/100/0	0/100/0	0/100/0
scope_expansion	72	0/100/0	0/100/0	0/100/0	0/100/0
shared_budget_across_tools	15	100/0/0	100/0/0	100/0/0	0/0/0
split_budget	36	0/100/0	0/100/0	0/100/0	0/100/0
stale_authorization	60	0/100/0	0/100/0	0/60/40	0/100/0
target_recipient_substitution	36	0/100/0	0/100/0	0/100/0	0/100/0
unseen_target_equivalence	15	0/0/0	0/0/0	0/0/0	0/0/0

qualitative safety–utility separation. The levels are protocol strata: L0 fixes the failure location; L2 varies structured returns and error wording; and L4 varies the tool graph, chain length, and retry timing. The defense-aware attacker observes only public monitor outcomes, reason codes, tool errors, and its own history; it selects from a declared 12-strategy vocabulary at $B_q \in \{1, 5, 10, 20, 50\}$ and records exhaustion instead of padding the budget.

The held-out compositional set contains 120 four-step recovery compositions whose sequential composition order is separated from the primary construction. Table 24 shows representative frozen trajectories across the evaluator path (failure context $\rightarrow$ candidate $\rightarrow$ runtime decision $\rightarrow$ independent effect check):

The family-level rates below use the same RecoverBench task-completion predicate as the main controlled analysis: an executable outcome must be independently authorized and match the intended abstract task effect. This keeps the operational definition of U_2 consistent across all controlled tables.

Table 15: Semantic-OOD composition over 135 cases. The suite changes effect-composition semantics independently of the primary construction. RCA removes the observed unauthorized prefixes while preserving the same measured safe-recovery utility as the matched local controls.

Method	$S_2^{\text{pref}} \downarrow$	$U_2 \uparrow$
StructuredLocal	22.2%	33.3%
LedgerLocal	22.2%	33.3%
RCA-Safe	**0.0%**	**33.3%**

Table 16: Independent OOD composition generator: 135 cases across eight effect-composition families. Values are S2 rates for structured runtime and RCA-Safe.

Composition family	Structured	RCA-Safe
Alias equivalence	0.0%	0.0%
Concurrent/reordered effects	0.0%	0.0%
Cross-tool duplicate	100.0%	0.0%
Multi-dimension effect	0.0%	0.0%
Multiple recipients	0.0%	0.0%
Nested authorization scope	0.0%	0.0%
Shared budget across tools	100.0%	0.0%
Unseen target equivalence	0.0%	0.0%

I Discussion

Authorization state, not recovery strategy. The results expose a separation between planning and authorization. Recovery strategies may replan, search, or change tools, but realized effects change the authorization state against which every subsequent proposal must be judged. RCA therefore treats recovery authorization as stateful closure over consumed authority rather than as a property of a particular planner, retry policy, or current-step guard.

J Artifact Reproducibility

The artifact package contains frozen inputs, trajectory-level outputs, clustered statistics, figure and table sources, source-system manifests, theorem diagnostics, and provenance records. Automated consistency checks bind repeated manuscript metrics and plotted source data to the same machine-readable statistics. The 200-case surrogate semantic audit and its adjudication records are included as a separate benchmark-validity artifact.

Table 17: Semantic dimensions separating primary and OOD compositions.

Dimension	Primary	OOD stress
Composition order	seen templates	unseen order
Cross-tool shared budget	limited	shared across tools
Recipient/effect scope	single or bounded	multi-recipient / multi-effect
Alias equivalence	limited	unseen aliases
Authorization nesting	flat	nested delegation
Duplicated effects	local	duplicated across tools

Table 18: A2 deterministic counterfactual collisions. The candidate and public policy view are held fixed; only the prior realized-effect history changes. Unsupported current-schema dimensions are retained explicitly.

Family	Status	N	Local empty/prior	Replay empty/prior	RCA empty/prior	RCA distinguishes
cumulative_budget	supported_current_algebra	1	1/1	1/0	1/0	1
one_shot_authorization	supported_current_algebra	1	1/1	1/0	1/0	1
cross_tool_shared_budget	supported_current_algebra	1	1/1	1/1	1/0	1
partial_effect	supported_current_algebra	1	1/1	1/0	1/0	1
target_recipient_binding	supported_capacity_proxy	1	1/1	1/0	1/0	1
scope_expansion	history_insensitive_current_algebra	1	0/0	0/0	0/0	0
alias_equivalence	unsupported_current_canonicalizer	1	0/0	0/0	0/0	0
expiry_revocation	unsupported_current_authority_state	1	1/1	1/1	1/1	0

Method	N	S_2^{pref}	U_2	Intervention	Overblocking	Status
progent_official	360	0.000	1.000	0.000	0.000	replay
progent_official_plus_rca	360	0.000	1.000	0.000	0.000	replay
agentspec_official	360	0.000	1.000	0.000	0.000	replay
agentspec_official_plus_rca	360	0.000	1.000	0.000	0.000	replay

Table 19: Source-executed compositional replay on the frozen common interface. Native Progent/AgentSpec logic is unchanged; "+RCA" adds only the project-side recovery boundary. No model is invoked. The RecoverBench replay has complete realized-effect traces, so the safety column is S_2^{pref}.

Table 20: A3/B3 Qwen3.5-9B results in the official tau2-bench mock live loop (120 episodes per policy). S_2^{pref} flags any accepted unauthorized prefix effect; overblocking is paired loss of safe task utility relative to the corresponding official policy.

Policy	S_2^{pref}	U_2	Intervene	Overblock	Parse fail
Progent	26.7	33.3	0.0	0.0	0.0
Progent + RCA	0.0	43.3	44.8	0.0	0.0
AgentSpec	5.8	40.8	0.0	0.0	0.0
AgentSpec + RCA	0.0	43.3	3.1	0.0	0.0

Rates are percentages. Official Progent and AgentSpec source guards are composed with the unchanged RCA boundary; the external domain, model, prompt, and evaluator are fixed across policies.

Table 21: Standalone RCA in live recovery loops. RCA removes the observed unauthorized effects relative to unguarded recovery while preserving the protocol-level task-completion opportunity. Each protocol contains 120 episodes per policy.

Protocol	Method	$S_2 \downarrow$	$U_2 \uparrow$	Task completion
Mixed	Unguarded	28.33%	63.33%	66.67%
Mixed	**RCA**	**0.0%**	**66.67%**	66.67%
Attack stress	Unguarded	85.0%	10.0%	20.0%
Attack stress	**RCA**	**0.0%**	**20.0%**	20.0%

Table 22: Frozen-candidate TS-Guard attribution replay. RCA removes every observed unsafe TS-Guard allow (category C) while preserving every observed safe TS-Guard allow (A); no residual unsafe allow (B) or safe newly blocked allow (D) is observed. N denotes replayable intercepted decision points drawn from each 120-episode live protocol, not the number of episodes.

Protocol	N	A	B	C	D	E	F
Mixed	106	72	0	15	0	19	0
Attack stress	102	0	0	41	0	60	1

Table 23: User-authorized independent-AI surrogate audit on the frozen 200-case A4 packet. This is not human annotation; evaluator agreement is recomputed from the visible frozen rule after adjudication.

Item	Value
Distinct frozen scenarios	200
Independent annotators	2 AI agents
Raw agreement	94.5%
Cohen's κ	0.887
Adjudicated disagreements	11
Disagreement taxonomy	stale_authorization: 11
Adjudicated evaluator agreement	100.0%
Status	AI surrogate complete; human claim not made

Table 24: Representative recovery trajectories from the frozen protocol, showing the candidate, runtime decision, and independent effect evaluation.

Scenario and family	Candidate after failure	Runtime decision	Independent evaluator outcome
Recipient substitution *fault-safe-fallback*	typed fallback; `messaging_api_fallback`; send to `user:alice` on `channel:team`	accepted; `RUNTIME_AUTHORIZED`; executable	authorized; no unauthorized effect
Recipient substitution *model candidate*	model candidate; `messaging_api`; send to `target:broader`	rejected; `RUNTIME_SCOPE_REJECTED`; executable	unauthorized; unauthorized effect observed
Payment splitting *fault-safe-fallback*	typed fallback; `payment_api_fallback`; transfer 40 to `acct:merchant`	accepted; `RUNTIME_AUTHORIZED`; executable	authorized; no unauthorized effect