GOVERNING PERSISTENT AGENT STATE WITH PROVENANCE-AWARE RECALL AND PROMOTION

Anonymous authors
Paper under double-blind review

ABSTRACT

An agent can block an unsafe report while retaining the contaminated state that produced it. Later sessions may reuse that state as evidence or turn it into a procedure. This paper studies what must remain valid across such transformations. A governance layer separates integrity risk, epistemic trust, and disclosure sensitivity, propagates unresolved source constraints at every write, and rechecks inherited verification evidence before reuse. The analysis gives a compositional ancestry invariant and a reuse model in which verification error can be shared across many downstream uses. This distinction matters empirically. On DRSC-Bench, 240 research tasks spanning three sessions, governance attains utility 0.74, Silent Trust Promotion (STP) 0.03, and Cross-Session Contamination (CSC) 0.02, compared with 0.72, 0.31, and 0.24 for ungoverned retrieval. Disabling the promotion gate raises STP to 0.15 and CSC to 0.09 while Unsafe Export (UE) remains 0.01 at the reported precision. Persistence-matched baselines, a ten-session extension, and controlled lineage interventions support the same conclusion: safe output does not certify safe persistent state. The formal guarantees require complete governed dependencies and sound certificates; lineage omission and semantic verification error remain outside that guarantee.

1 INTRODUCTION

A persistent research agent does not end when it returns an answer. Sources become notes, notes become report evidence, and report evidence may later be compressed into memory or turned into a reusable procedure. Each transformation increases the distance from the original source while preserving the possibility that its restrictions still matter. An unreliable claim can acquire the appearance of verified evidence after summarization; restricted material can survive inside a public-facing artifact; a procedure justified for one task can be recalled under a different purpose. In all three cases the adversarial source may already have left the active context. The security-relevant object is therefore not only the current response, but also the state that future sessions inherit. Final-output filtering cannot repair a bad persistent transition. In DRSC-Bench, disabling the promotion gate raises Silent Trust Promotion (STP) from 0.03 to 0.15 and Cross-Session Contamination (CSC) from 0.02 to 0.09, while Unsafe Export (UE) remains 0.01 at the reported precision and utility changes only from 0.74 to 0.73. An evaluator looking only at the released answer would see little change even though the agent has become substantially more likely to trust or reuse the wrong state. This motivates a separate assurance question: after a response is over, what has become eligible to be stored, relied on, and carried into later decisions?

Information-flow control provides a natural starting point. CaMeL separates trusted control from untrusted data, Fides propagates information-flow labels, permissive analysis limits unnecessary over-tainting, and PRUDENTIA plans under security constraints (Debenedetti et al., 2026; Costa et al., 2025; Siddiqui et al., 2025; Kolluri et al., 2026). Persistent-memory attacks expose the complementary failure mode: one admitted object can influence later sessions after the original payload has disappeared (Chen et al., 2024; Dong et al., 2025; Srivastava & He, 2025; Dash et al., 2026; Huang et al., 2026). Persistence makes a local exception consequential. If a restriction is relaxed, a durable object must retain why the relaxation was allowed, which immutable version was checked, what support justified it, and which later uses remain covered. Reusing a cached label after that justification ceases to apply is not the same error as admitting malicious text at ingress, and an export policy alone cannot distinguish the two.

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.

The governance model treats memory records, evidence objects, summaries, and procedural drafts as immutable versions in a dependency graph. Integrity risk, epistemic trust, and disclosure sensitivity remain separate because contamination, evidential reliance, and dissemination are different decisions. Writes preserve unresolved parent constraints; an authorized certificate may discharge a specific constraint, but the certificate becomes part of the descendant's witness history and is checked again at recall and release. This lifecycle yields two distinct limits. Missing dependencies create an observability failure that metadata alone cannot repair. Complete lineage does not eliminate semantic verification error: one wrong endorsement may be reused by many descendants, creating correlated downstream failures whose probability and conditional severity differ. Figure 1 shows the resulting separation between persistent admission and external release.

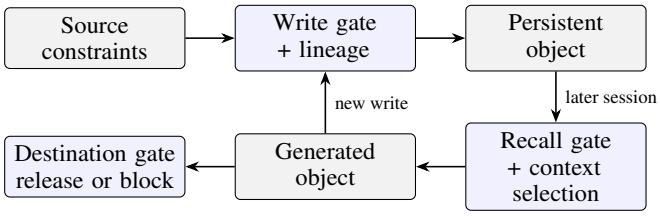

Figure 1: Persistent admission and external release are distinct boundaries. Generated content can re-enter memory, so unresolved ancestry and inherited verification evidence remain relevant after the session that created it.

Contributions. At the system level, the paper develops persistence-aware governance in which source constraints and the evidence authorizing their relaxation survive derivation, recall, and release. The formal analysis separates missing-lineage failures from certificate failures, proves preservation of unresolved ancestry inside the governed boundary, and characterizes how endorsement reuse changes lifetime risk: expected affected use, any-failure probability, and severity conditional on failure need not move together. Empirically, DRSC-Bench evaluates 240 research tasks across three sessions with separate UE, STP, and CSC measurements. Persistence-matched baselines, lineage interventions, matched backbones, and a ten-session extension test the same distinction from different directions. The resulting claim is deliberately narrower than general agent safety: a safe release does not certify safe persistent state, and assurance for a stored object must remain valid for the uses that inherit it.

2 Related Work

Indirect prompt injection exploits the boundary between trusted instructions and external content (Greshake et al., 2023; Zhan et al., 2024). Classical information-flow and provenance systems provide foundations for confidentiality, integrity, and derivation tracking (Denning, 1976; Biba, 1977; Myers, 1999; Cheney et al., 2009); recent agent systems bring those ideas into LLM execution. CaMeL isolates trusted control flow, Fides propagates confidentiality and integrity labels, permissive information-flow analysis avoids unnecessary label propagation, and PRUDENTIA recovers autonomy by planning under policy constraints (Debenedetti et al., 2026; Costa et al., 2025; Siddiqui et al., 2025; Kolluri et al., 2026). AgentPoison, MINJA, MemoryGraft, PoisonedRAG, MPBench, and PMPA show that retrieved or retained state can itself become an attack surface (Chen et al., 2024; Dong et al., 2025; Srivastava & He, 2025; Zou et al., 2025; Dash et al., 2026; Huang et al., 2026). A-MemGuard validates memories and keeps corrective experience, while MemSentry reasons about source trust, semantic risk, and dependency reach (Wei et al., 2026; Roy & Basu, 2026). These lines of work motivate secure admission; the additional question here is whether a previously authorized transition remains justified after its output has been transformed and reused. Scope Before You Persist reaches a related scope-matching principle for persistent skill edits (Cheng et al., 2026).

Research-agent benchmarks probe a different part of the problem. AgentDojo studies tool-mediated attacks, while DRBench, LiveResearchBench, and CI-Work evaluate long-horizon research quality or contextual integrity (Debenedetti et al., 2024; Abaskohi et al., 2026; Wang et al., 2026; Fu et al., 2026). Persistent procedural artifacts add another reuse surface: SkillsBench and SkillRevise evaluate the usefulness or revision of agent skills, and MUSE-Autoskill and SkillHone explicitly retain skills across episodes (Li et al., 2026b; Liu et al., 2026; Lin et al., 2026; Li & Hu, 2026). Security studies show that lifecycle attacks, composition effects, and covert policy steering can invalidate an artifact

that appears acceptable in isolation (Badhe & Tiwari, 2026; Zhao et al., 2026; Li et al., 2026a); task-conditioned runtime guards and proof-carrying actions address complementary execution-time concerns (Yang et al., 2026; Wang, 2026). The distinction developed here concerns the persistent object's authorization history: whether the evidence that permitted its current status still applies to the current version, use, and dependency closure.

3 Objects, Evidence, and the Enforcement Boundary

Let $\mathcal{M}$ be the persistent store. An object m has immutable content and a content-version identifier $h(m)$, together with a versioned governance record

$$\Gamma(m) = \big(h(m), \mathcal{P}(m), \iota(m), \tau(m), \sigma(m), t_{\mathrm{ver}}(m), q(m), \rho(m)\big). \tag{1}$$

Here $\mathcal{P}(m)$ records provenance and audit dependencies, including direct derivation parents and certificate-support objects; $t_{\mathrm{ver}}(m)$ is the last verification time, $q(m)$ specifies scope constraints over domains, roles, destinations, and capabilities, and $\rho(m) \in [0, 1]$ is an empirical ingress-risk score. The score is distinct from the durable integrity-risk label $\iota(m)$. The label orders are

$$\begin{aligned} \mathcal{I} &= \{\mathsf{clean} \preceq \mathsf{uncertain} \preceq \mathsf{untrusted} \preceq \mathsf{adversarial}\}, \\ \mathcal{T} &= \{\mathsf{untrusted} \preceq \mathsf{provisional} \preceq \mathsf{verified} \preceq \mathsf{trusted}\}, \\ \mathcal{S} &= \{\mathsf{public} \preceq \mathsf{internal} \preceq \mathsf{restricted}\}. \end{aligned} \tag{2}$$

Larger integrity risk is more restrictive, larger trust permits stronger epistemic reliance, and larger sensitivity restricts dissemination. These labels express policy judgments, not probabilities of truth. Ingress combines source reputation and evidence under a fixed ordinal rule (Appendix E.3); source reputation alone cannot assign the top trust class.

A use request specifies a contract

$$u = (g, \ell, v, K, \nu, \beta), \tag{3}$$

where g is a task family, ℓ is a role such as search lead or report evidence, v is a destination, K is the capability set allowed by the control plane, ν identifies relevant resource versions, and β is a compatibility predicate. Evidence requirements and disclosure permissions are separate policy dimensions: using content as a lead, executing a procedure, and disclosing it cannot be placed in one universal strength ordering. The core research tasks exercise retrieval, reporting, and procedural-draft retention. A separate sandbox evaluation exercises bounded analytical functions. Execution also requires an explicit capability check by the control plane.

The trusted runtime intercepts external observations, persistent writes, governed recall, and export; it owns the metadata store, policy epoch, issuer registry, and capability checks. An attacker can supply retrieved content, induce writes, and wait for later reuse, but cannot forge an authorized signature or bypass these interfaces. The model can still follow malicious instructions, hallucinate, or omit an influencing source, and no guarantee covers hidden model state, an uninstrumented cache, or exports outside the runtime. Content updates and governance-record changes receive distinct version identities, and each derivation refers only to existing versions, so $D = (V, E)$ is acyclic. Conservative capture records all governed objects visible during generation and therefore contains every influence from that recorded input boundary, at the cost of possible over-propagation. Selective attribution retains a narrower set and can omit an influence. The main experiments use selective attribution; the lineage stress test evaluates conservative capture, whereas the ancestral guarantee assumes complete capture.

3.1 Certificates and inherited verification evidence

A certificate records a scoped justification for changing a label constraint:

$$c = (h_d, h_S, a, E_c, L_c, \mathcal{U}_c, j_c, \kappa_c, T_c, \chi_c). \tag{4}$$

The fields bind the subject version h_d, support versions h_S, label axis a, discharged edges E_c, authorized label bound L_c, permitted contracts $\mathcal{U}_c$, issuer j_c, policy epoch κ_c, expiry T_c, and assurance type χ_c. Support objects are recorded as audit dependencies so that their witness histories are inherited. A trust-promotion certificate is termed an endorsement below. An endorsement does

not lower sensitivity; an integrity sanitizer does not authorize disclosure; and declassification requires its own authorized issuer.

The predicate $\mathrm{Check}(c, u, t)$ checks the signature, issuer authority for that transition, version bindings, contract membership, current epoch, expiry, and revocation status. It is mechanically decidable. Semantic soundness is a separate premise: the supporting evidence or authorized transformation must actually justify L_c and E_c. A signature cannot establish that a free-text claim is true. The evidence must justify the particular transition, including any discharged dependencies.

Each object retains the certificates used to justify its labels, including inherited justifications. If $\mathcal{C}_{\mathrm{loc}}(d)$ contains certificates used at the write of d, a conservative witness closure is

$$\mathcal{W}(d) = \mathcal{C}_{\mathrm{loc}}(d) \cup \bigcup_{x \in \mathrm{Dep}(d)} \mathcal{W}(x). \tag{5}$$

A removed edge remains in the audit graph and in this closure. Otherwise a descendant could preserve a relaxed label while forgetting the certificate that justified it. Define $\mathrm{Current}(m, u, t)$ to require current source/resource versions and $\mathrm{Check}(c, u, t) = 1$ for every $c \in \mathcal{W}(m)$. An empty witness set passes only the certificate part of this check; it supplies no positive evidence. A failed check blocks that record. Recovery requires a fresh derivation from valid support rather than reuse of the invalid inherited justification. The specification requires check and use to share a policy snapshot. The empirical evaluation does not exercise revocation or resource-version changes.

Consider a source acceptable as a search lead but inadequate as evidence for a public report. A summary of that source retains the source constraint. If an independent check establishes the summary's factual claim, a trust endorsement can permit evidential reliance without removing the internal document's disclosure restriction. A later procedural draft may depend on that summary, yet the endorsement for report evidence does not authorize running the draft. The contract and witness checks express these distinctions at reuse time. Storing only the final label is insufficient: the label loses its justification when the authorized use or the evidence changes, even if the object's text does not.

4 Persistent Governance and Its Limits

4.1 Write-time preservation

For axis $a \in \{\iota, \tau, \sigma\}$, $\mathrm{Dep}_a(d)$ contains the direct dependencies whose constraint has not been discharged by an applicable certificate. Fix a compatible policy snapshot for the derivation under consideration. If s is a source, let $U_a(s) = \{s\}$; otherwise

$$U_a(d) = \bigcup_{x \in \mathrm{Dep}_a(d)} U_a(x). \tag{6}$$

The write gate imposes

$$\iota(d) \succeq \bigvee_{x \in \mathrm{Dep}_\iota(d)} \iota(x), \qquad \tau(d) \preceq \bigwedge_{x \in \mathrm{Dep}_\tau(d)} \tau(x), \qquad \sigma(d) \succeq \bigvee_{x \in \mathrm{Dep}_\sigma(d)} \sigma(x). \tag{7}$$

Certified transitions must also satisfy their recorded bounds L_c. The promotion gate is the part of the write gate that rejects a more permissive trust assignment unless an applicable certificate authorizes it. Empty joins equal the corresponding lattice bottom and an empty meet equals the trust top. These are algebraic identities, not default label assignments. A generated object with no unresolved trust parent and no applicable endorsement is capped at provisional. An object with unresolved parents can inherit their bound without a blanket provisional cap. Neither inheritance nor an endorsement makes arbitrary generated statements factually correct.

Proposition 1 (Preservation of unresolved ancestry)**.** *On a finite acyclic governed derivation graph, suppose every write satisfies Eq.* (7)*. Then every object d satisfies*

$$\iota(d) \succeq \bigvee_{s \in U_\iota(d)} \iota(s), \quad \tau(d) \preceq \bigwedge_{s \in U_\tau(d)} \tau(s), \quad \sigma(d) \succeq \bigvee_{s \in U_\sigma(d)} \sigma(s). \tag{8}$$

If dependencies contain all governed influences and discharged constraints have sound certificates, these bounds preserve every unresolved source constraint within that boundary.

Topological induction proves the claim by substituting each parent's ancestral bound into the local join or meet (Appendix B). Wrong source labels and unsound edge discharges can satisfy the algebra while violating the intended policy; they are separate semantic failures.

Proposition 2 (Limit of metadata-only attribution)**.** *If two executions expose identical metadata but differ by an unrecorded influence that tightens a true label constraint, a metadata-only policy has the same decision distribution in both executions. It cannot selectively accept one and reject the other on that evidence alone.*

This is an observability limit. Rejecting both executions remains safe; distinguishing them while retaining permissiveness requires another observation. Conservative capture supplies a broader observation boundary by recording all visible objects. Independent verification can justify a new claim without reconstructing the original generation's causal attribution.

4.2 Recall, context selection, and release

For goal G, observations O, use u, and time t, let P specify thresholds $\iota_P^{\max}$, $\tau_P^{\min}$, $\sigma_P^{\max}$, risk limit ρ_P, and age limit Δ_P. The admissibility predicate is

$$\begin{aligned} A(m; G, O, u, t) = \mathbf{1}[\iota(m) \preceq \iota_P^{\max}]\mathbf{1}[\tau(m) \succeq \tau_P^{\min}]\mathbf{1}[\sigma(m) \preceq \sigma_P^{\max}] \\ \cdot \mathbf{1}[\mathrm{ScopeOK}(q(m), G, O, u)]\mathbf{1}[0 \le t - t_{\mathrm{ver}}(m) \le \Delta_P] \\ \cdot \mathbf{1}[\rho(m) \le \rho_P]\mathbf{1}[\mathrm{Current}(m, u, t)]. \end{aligned} \tag{9}$$

ScopeOK is a conjunction of the policy's required domain, role, destination, and capability conditions; a nonempty overlap in domain names is insufficient. Retrieval ranks only $\mathcal{M}_A = \{m \in \mathcal{M} : A(m; G, O, u, t) = 1\}$. Semantic similarity never overrides a failed hard check. A stale object may become usable after revalidation, but advancing its timestamp requires evidence rather than a model-written assertion.

A request fixes its policy and use contract before retrieval. The runtime checks inherited evidence, filters candidates, and only then allows ranking. Generation proposes a new object and its dependencies; the write gate applies the parent bounds and any certified exceptions. Release checks the resulting object against its destination policy. These interception points cover intermediate summaries and observations as well as explicitly named memory records. A model-written dependency list or verification claim has no authority to bypass a gate.

The context assembler ranks admissible objects by a relevance score minus an empirical risk penalty,

$$r_{\mathrm{ctx}}(m; G, O) = r_{\mathrm{rel}}(m; G, O) - \lambda_{\mathrm{risk}}\rho(m). \tag{10}$$

It adds objects until a validation-selected residual-need score is at most η or the context budget is exhausted. The implemented threshold η has no entropy units. Agent View Sufficiency (AVS) expresses a population information-bottleneck interpretation (Tishby et al., 1999); its entropy tolerance ϵ is distinct from η. The objective and context sweep are given in Appendix D. The implemented greedy score is a heuristic, and empirical contamination is not a mutual-information estimate.

An output o is a derived object subject to the same write constraints. Let destination policy P_v specify the corresponding label, risk, and age thresholds. Define

$$\begin{aligned} \mathrm{ReleaseOK}(o; G, O, u, t) = \mathbf{1}[\mathrm{Current}(o, u, t)]\mathbf{1}[\mathrm{ScopeOK}(q(o), G, O, u)] \\ \cdot \mathbf{1}[0 \le t - t_{\mathrm{ver}}(o) \le \Delta_{P_v}]\mathbf{1}[\rho(o) \le \rho_{P_v}] \\ \cdot \prod_{a \in \{\iota, \tau, \sigma\}} \mathbf{1}[\mathrm{AxisOK}_a(o, P_v)]. \end{aligned} \tag{11}$$

Here AxisOK uses the three label inequalities of Eq. (9) with destination thresholds. Release occurs only when Eq. (11) equals one. Sanitization creates a new version with its own dependencies and certificates, then repeats the check. If neither object passes, export is blocked. Sanitization therefore has no unchecked release path.

4.3 Certificate error under repeated reuse

Correct enforcement does not remove semantic verifier error. Fix a finite set of certificates $\mathcal{C}$ and use events $\mathcal{J}$. Let $B_{cj} \in \{0, 1\}$ indicate that event j relies on certificate c, F_c denote semantic certificate

failure with $\Pr(F_c) \le p_c$, and $w_j \ge 0$ be the loss weight of an inadmissible event. This analysis isolates certificate-induced violations: their event V_j must satisfy $V_j \subseteq \bigcup_{c:B_{cj}=1} F_c$. Other failures, including missing lineage and wrong ingress labels, require separate accounting.

Proposition 3 (Reuse-weighted certificate loss)**.** *For $H_{\text{cert}} = \sum_j w_j \mathbf{1}[V_j]$ and $W_c = \sum_j w_j B_{cj}$,*

$$\mathbb{E}[H_{\text{cert}}] \le \sum_j w_j \min\left\{1, \sum_c B_{cj} p_c\right\} \le \sum_c p_c W_c. \tag{12}$$

No independence between certificate errors is required.

The proof uses a union bound for each event and linearity of expectation. If one erroneous certificate makes all N dependent uses inadmissible, then the probability of at least one such failure is p and the expected number is Np. Repeating the same certificate is not equivalent to N independent checks, whose probability of at least one error would be $1 - (1-p)^N$.

The incidence graph in Proposition 3 is fixed. When an agent preferentially reuses a misleading but successful-looking artifact, exposure can depend on the error itself, and replacing $\mathbb{E}[\mathbf{1}[F_c]W_c]$ by $p_c\mathbb{E}[W_c]$ is unjustified. An enforceable alternative caps total future exposure before issuance. Let $\mathcal{H}_{k-1}$ denote pre-issuance history, F_k the error event, and W_k the realized downstream loss weight of certificate k. Choose nonnegative, $\mathcal{H}_{k-1}$-measurable α_k and b_k such that

$$\Pr(F_k \mid \mathcal{H}_{k-1}) \le \alpha_k, \qquad 0 \le W_k \le b_k. \tag{13}$$

Proposition 4 (Adaptive reuse with predictable caps)**.** *Under certificate-induced event containment and Eq.* (13)*,*

$$\mathbb{E}[H_{\text{cert}}] \le \sum_k \mathbb{E}[\alpha_k b_k]. \tag{14}$$

An almost-sure budget $\sum_k \alpha_k b_k \le \delta$ therefore bounds expected loss by δ. If every violating use has weight at least one, then $\Pr(H_{\text{cert}} > 0) \le \delta$.

The cap must charge descendants that inherit an endorsement, including uses admitted in later sessions. A counter that charges only the current recall does not bound W_k. The proof and a counterexample to unrestricted adaptive factorization appear in Appendix B.5. Establishing conditional error bounds requires more than a static verifier accuracy estimate; the empirical evaluation does not instantiate this exposure-budget rule.

For an exact illustration, partition N uses into C nonempty groups of sizes n_c. Each group depends on one certificate; independent certificate errors have probability p, and an error makes every use in its group inadmissible. The bad-use count $L = \sum_{c=1}^{C} n_c \mathbf{1}[F_c]$ then satisfies

$$\mathbb{E}[L] = Np, \qquad \mathrm{Var}(L) = p(1-p)\sum_{c=1}^{C} n_c^2, \qquad \Pr(L > 0) = 1 - (1-p)^C. \tag{15}$$

Sharing changes concentration even when expected loss stays fixed. At $N = 20$ and $p = 0.05$, moving from twenty independent certificates to one shared certificate lowers $\Pr(L > 0)$ from 0.642 to 0.05 but raises $\mathbb{E}[L \mid L > 0]$ from 1.56 to 20. It is not an unqualified safety improvement. Appendix C derives the distribution and sampling uncertainty. A larger reuse weight increases the expected-loss value of a given reduction in certificate error; it need not increase the value of that reduction for the any-failure objective.

5 Experiments

5.1 Protocol and comparisons

DRSC-Bench contains 240 tasks, with 40 each in science, technology, medicine, finance, law, and history. A task spans discovery, synthesis with persistent writes, and delayed recall/reporting. Its bundle contains public evidence, simulated restricted documents, and adversarial pages. Five attack families target indirect instruction following, memory poisoning, cross-session carry-over, procedural-draft contamination, and restricted-data exfiltration. The clean counterpart removes planted adversarial

content while preserving the research question. Task construction and implementation settings appear in Appendix E. Research Utility averages factual accuracy, required-subtopic coverage, and report quality. Unsafe Export (UE) measures a destination-policy violation; Silent Trust Promotion (STP) measures inappropriate eligibility for higher-trust reuse; Cross-Session Contamination (CSC) measures delayed-answer disagreement with the clean counterpart. Every core task contributes one designated opportunity for each metric, including blocked opportunities in the denominator. Five runs yield 1,200 task-run opportunities per method. The 240 task identities recur, so inference resamples tasks while retaining their paired methods and five runs. CSC includes background answer variation: clean–clean disagreement is 0.01–0.02, as detailed in Appendix E.2.

The core comparison uses GPT-4o, a shared search index and task manifests, and selective attribution. Baselines comprise Naive RAG, a CaMeL-style adaptation, a Fides adaptation, and A-MemGuard. The tested Fides adaptation enforces action-time labels but does not serialize them across the benchmark's persistence interface. A separate comparison adds persistent labels and a PRUDENTIA-style planner; this distinction concerns implementations, since Fides itself supports labeled storage. Gate ablations remove recall filtering or the promotion gate while retaining destination checks. Run standard deviations describe dispersion; task-cluster bootstrap intervals describe paired differences.

Table 1: Core evaluation on GPT-4o. UE, STP, and CSC use fixed allocated opportunities. The first five rows use exact event counts divided by 1,200; ablation rates have lower reported precision. Utility is mean $\pm$ run standard deviation.

Configuration	Utility ↑	UE ↓	STP ↓	CSC ↓
Naive RAG	0.72 ± 0.01	0.1792	0.3100	0.2400
CaMeL-style	0.68 ± 0.01	0.0417	0.2200	0.1900
Fides adaptation	0.65 ± 0.02	0.0208	0.2000	0.1600
A-MemGuard	0.70 ± 0.01	0.0917	0.1208	0.1100
Full governance	0.74 ± 0.01	0.0125	0.0300	0.0200
Without recall gate	0.71 ± 0.01	0.020	0.080	0.070
Without promotion gate	0.73 ± 0.01	0.010	0.150	0.090

5.2 Persistent outcomes and matched comparisons

Table 1 shows the strongest separation in persistent outcomes. Full governance has STP 0.0300 and CSC 0.0200, versus 0.1208 and 0.1100 for A-MemGuard, while utility is 0.74 versus 0.70. Its UE count is 15, compared with 25 for the Fides adaptation, an absolute rate difference of 0.00833. Relative to Naive RAG, the count-based reductions are 93.0% for UE and 91.7% for CSC; the utility difference is 0.020, with a task-cluster 95% interval of $[0.005, 0.035]$ (Appendix L). The gate ablations expose a different pattern. Removing recall filtering raises CSC to 0.07, whereas disabling the promotion gate raises STP to 0.15 and CSC to 0.09 even though UE still rounds to 0.01 because the destination gate remains active. Output protection can therefore conceal persistent-state corruption in this implementation. These ablations do not isolate every interaction among source labeling, attribution, verification, and retrieval, but they show why release-only and utility-only evaluation miss the principal failure.

Clean–clean controls put the delayed-answer metric in context. Naive RAG has attack–clean disagreement 0.24 and clean–clean disagreement 0.02; full governance has 0.02 and 0.01, giving descriptive excesses of 0.22 and 0.01. The subtraction is not a causal effect because the clean control is not a randomized crossover at every run seed, so CSC is interpreted jointly with STP, UE, and report utility. Fixed denominators also prevent rejection from disappearing from evaluation. SafeSuccess requires utility at least 0.7 and no UE, STP, or CSC event on the task-run; it is 0.71 for full governance, 0.63 for A-MemGuard, and 0.55 for Naive RAG, with corresponding any-violation rates 0.06, 0.19, and 0.41 (Appendix F). Opportunity coverage is reported separately because acting less often is not itself a safety improvement.

Label serialization improves the Fides adaptation's STP from 0.20 to 0.14 and CSC from 0.16 to 0.11 (Table 2). Policy-aware planning raises utility to 0.69 and coverage to 0.84 while lowering STP and CSC to 0.13 and 0.10. Full governance reaches coverage 0.96 with STP 0.03 and CSC 0.02. Against the PRUDENTIA-style configuration, the paired differences are -0.10 for STP, -0.08 for CSC, and $+0.05$ for utility, with 95% intervals $[-0.13, -0.07]$, $[-0.11, -0.05]$, and $[0.02, 0.08]$. Mechanism diagnostics in Appendix H are consistent with this separation: governed recall has precision 0.88,

Table 2: Persistence-matched comparison on GPT-4o. Coverage is the proportion of allocated opportunities acted on. Rates use two-decimal precision. These adaptations preserve labels through the benchmark's memory interface.

Configuration	Utility	UE	STP	CSC	Coverage
Fides adaptation	0.65	0.02	0.20	0.16	0.78
Fides + persistent labels	0.66	0.02	0.14	0.11	0.80
PRUDENTIA-style + persistent labels	0.69	0.02	0.13	0.10	0.84
Full governance	0.74	0.01	0.03	0.02	0.96

recall 0.82, and an inadmissible re-entry block rate of 0.94; governed writes have label accuracy 0.93, block 0.91 of contaminated procedural drafts, and produce clean-write false positives at rate 0.04 versus 0.18 under static labels. These diagnostics use different opportunity sets and should not be read as complements of task-level STP or CSC.

5.3 Dependency failures, longer reuse, and execution

An evaluator intervention removes an influencing source from selective attribution. CSC rises to 0.09. Under the same intervention, conservative capture lowers CSC to 0.04 while utility decreases from 0.73 to 0.71 (Appendix H). The matched utility cost is therefore 0.02. Conservative capture addresses missing recorded influence, but source-label and endorsement errors remain. A separate promotion probe presents repeated low-trust variants and raises STP from 0.03 to 0.06. This is a task-level promotion failure rate; it does not estimate the error probability of an individual certificate.

Table 3: Session-10 outcomes on a 60-task extension, with three runs per configuration. Values are mean $\pm$ run standard deviation. This subset differs from the 240-task core evaluation.

Configuration	Utility at session 10 $\uparrow$	CSC at session 10 $\downarrow$
Naive RAG	0.66 ± 0.02	0.41 ± 0.03
A-MemGuard	0.65 ± 0.02	0.22 ± 0.02
Full governance	0.72 ± 0.01	0.04 ± 0.01

The ten-session extension introduces new public evidence, an adversarial page, and a restricted field at each session. At session 10, governance has utility 0.72 and CSC 0.04, compared with 0.66 and 0.41 for Naive RAG (Table 3). These endpoints show a persistent advantage within that 60-task extension; comparing them directly with the 240-task session-3 means would confound horizon with task selection and does not identify an accumulation rate. Appendix K also describes a bounded Python-skill sandbox, which tests capability checks without claiming security for unrestricted installed skills. With the same governance settings, matched backbone comparisons reduce CSC from 0.26 to 0.03 on Claude 3.5 Sonnet and from 0.31 to 0.08 on Llama 3.1 70B; utility changes from 0.70 to 0.72 and from 0.61 to 0.65 (Appendix J). These runs support configuration transfer but do not attribute residual differences to a particular mechanism. For report quality, a 100-task human comparison gives correlation 0.85 between the LLM judge and the mean of two raters, with quadratic-weighted inter-rater agreement 0.71; this validates the report-quality measurement rather than the safety labels.

5.4 Shared endorsements and operating cost

The controlled reuse experiment fixes 20 use events and certificate error probability $p = 0.05$. Independent certificates cover one use each; five shared certificates cover four uses each; the star and chain each share a single root endorsement across all uses. Table 4 separates Monte Carlo observations from exact model quantities. The observations are compatible with the common expected bad-use count of one. Sharing instead changes how that loss is distributed: the single-endorsement configurations have much lower any-failure probability, but every failure affects all twenty uses.

The star and chain induce the same certificate–use incidence and therefore the same loss distribution here; their comparison does not isolate a depth effect. The sweep also does not validate a real verifier because p is controlled rather than estimated from STP. Its role is to expose the reporting consequence of reuse: a lower any-failure probability may coexist with much larger conditional severity, while Eq. (12) gives an accounting bound once certificate-level error bounds and use weights are available. The empirical context selector exposes a separate operating trade-off. Raising residual-need tolerance

Table 4: Controlled certificate sharing, 2,000 Monte Carlo trials. MC columns are observed means; variance and conditional severity are exact values from Eq. (15). L counts inadmissible uses. Certificate count is a count of checks, not measured runtime.

Configuration	Checks	MC $\bar{L}$	MC $\widehat{\Pr}(L > 0)$	$\mathrm{Var}(L)$	$\mathbb{E}[L \mid L > 0]$
Independent	20	1.00	0.64	0.95	1.56
Five groups of four	5	1.01	0.23	3.80	4.42
Star, one endorsement	1	1.02	0.05	19.00	20.00
Chain, one endorsement	1	0.98	0.05	19.00	20.00

η from 0.01 to 0.50 reduces mean context size from 24.3 to 6.4 objects and CSC from 0.05 to 0.01; utility stays at 0.74 through $\eta = 0.20$ and falls to 0.65 at $\eta = 0.50$ (Appendix D). At the default setting, median end-to-end task latency rises from 41.8 to 46.9 seconds (12.2%), with P95 and P99 increases of 11.7% and 11.5%. These latency measurements cover the core research pipeline, not skill sandboxing, adaptive exposure budgets, or revocation-driven revalidation.

6 Discussion and Limitations

Formal and empirical claims operate at different scopes. The ancestry invariant (Prop. 1) assumes complete recorded dependencies and sound exceptional transitions. The main experiments use selective attribution, which satisfies neither precondition unconditionally; Prop. 2 covers the residual failures under conservative capture and omission interventions. The certificate-loss bound (Prop. 3) isolates semantic endorsement errors, and the adaptive-cap result (Prop. 4) shows reuse accounting needs per-issuance conditional bounds rather than marginal accuracy. Missing lineage, incorrect ingress labels, and channels outside the governed interface fall outside these propositions and need separate terms in a deployment risk analysis. The persistence-matched PRUDENTIA-style configuration narrows one comparison, but verifier and budget differences remain, so the residual gap is not attributed to one design choice.

The evaluation uses simulated restricted material, five fixed attack families, and a bounded Python sandbox. The ten-session extension reports endpoints on a 60-task subset; comparing them with the 240-task session-3 means would confound horizon with task selection. The sandbox measures capability enforcement for analytical functions, not arbitrary installed skills with external dependencies. Revocation latency, environment drift, and verifier calibration under adaptive attack remain outside the measured surface. The supported claim is narrow but consequential: output filtering and write gating protect different boundaries, and a benign release does not certify the persistent state.

One pattern generalizes beyond this benchmark. In the controlled topology sweep, the expected bad-use count stays near Np under every configuration, yet $\Pr(L > 0)$ falls from 0.64 under independent certificates to 0.05 under one shared endorsement, while $\mathbb{E}[L \mid L > 0]$ rises from 1.56 to 20. The same verifier error rate therefore produces qualitatively different lifetime risk depending on how endorsements are shared. Verification-budget arguments based only on marginal accuracy or expected loss are incomplete; the objective must state whether rarity, count, or conditional severity is minimized.

7 Conclusion

Persistent agents need assurance over a lifecycle, not a single response. Blocking an unsafe export and preventing an unsupported transition from becoming durable are distinct; the first can succeed while the second fails. The promotion-gate ablation leaves Unsafe Export at 0.01 while raising Silent Trust Promotion from 0.03 to 0.15 and Cross-Session Contamination from 0.02 to 0.09. The governance model preserves unresolved constraints at every write and retains the scoped evidence behind any relaxation, so recall and release recheck the certificate against the current version, use contract, and dependency closure. Against Naive RAG, the count-based reductions are 93.0% for UE and 91.7% for CSC, with utility increasing from 0.72 to 0.74 and SafeSuccess from 0.55 to 0.71. Reuse changes the risk geometry: sharing one endorsement across many descendants can leave expected affected use unchanged while making failures rarer and more concentrated. This distinction, not the module inventory, is the central claim; evaluation should separate state admission, reuse, and release and state the objective that verification budget minimizes.

AI USE STATEMENT

In this work, we used generative AI tools only for identifying and retrieving potentially relevant literature and for language editing to improve the clarity, grammar, and readability of the manuscript. We have not used generative AI tools to formulate research ideas or hypotheses, develop the proposed methodology, design experiments, derive mathematical claims or proofs, implement methods or generate code, process or analyze data, interpret experimental results, produce results, or create figures, tables, or other research artifacts; these required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for literature retrieval and language polishing, which are recommended for disclosure. We have reviewed all AI-assisted work. In particular, any literature identified with AI assistance was checked against the original sources for existence, relevance, bibliographic accuracy, and appropriate citation context, and all AI-assisted language edits were reviewed to ensure that they did not alter the technical meaning, claims, or contributions of the paper. No AI tool was used to generate code, data, or experimental artifacts. The authors take full responsibility for the final content of this work, including all text, claims, references, and artifacts.

ETHICS STATEMENT

The authors have reviewed the ICLR Code of Ethics and believe that this work complies with it. This research does not involve animal experiments, personally identifiable information, or the collection or release of sensitive personal data. Human ratings, if any, concern research-report quality only and were collected in accordance with institutional ethical guidelines. The empirical evaluation uses publicly available benchmarks and controlled, simulated data in accordance with their applicable licenses and terms of use. No new dataset containing real secrets, private, sensitive, or personally identifiable information is introduced or released. For experiments involving simulated restricted documents and sandboxed attack artifacts, all such artifacts are synthetic; marked fields contain no real secrets, and attack artifacts are executed only in isolated environments. We discuss limitations, potential biases in benchmarks and evaluation settings, dual-use risks, and broader implications in the paper. The authors declare no conflicts of interest.

REPRODUCIBILITY STATEMENT

We have taken several steps to facilitate reproducibility. The main paper specifies the proposed methodology, objectives, and evaluation protocol, while the appendix provides additional implementation details, experimental settings, hyperparameters, dataset and preprocessing information, evaluation metrics, and ablation studies needed to reproduce the reported results. For theoretical results, the assumptions and derivations are stated in the main paper, with complete proofs and additional analysis provided in the appendix. For the controlled sharing model, the appendix gives the exact finite distribution. Reported task-cluster intervals retain paired methods and repeated runs; aggregate event counts do not determine those intervals. To preserve anonymity during review, the source code and supporting scripts are not publicly released at submission time. Upon paper acceptance, we will publicly release the code, configuration files, and scripts in a public repository, together with instructions to reproduce the main experiments. The datasets used in the experiments are publicly available, and any task-specific preprocessing or data splits required for the reported experiments are documented in the appendix or will be included in the release.

REFERENCES

Amirhossein Abaskohi, Tianyi Chen, Miguel Muñoz-Mármol, Curtis Fox, Amrutha Varshini Ramesh, Étienne Marcotte, Xing Han Lu, Nicolas Chapados, Spandana Gella, Christopher Pal, Alexandre Drouin, and Issam Laradji. DRBench: A realistic benchmark for enterprise deep research. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/0ba520d93c3df592c83a611961314c98-Abstract-Conference.html.

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2):1641–1662, 2025. doi: 10.1214/24-AOAS1998.

Sanket Badhe and Priyanka Tiwari. Agent skill security: Threat models, attacks, defenses, and evaluation. *arXiv preprint arXiv:2607.13987*, 2026. URL https://arxiv.org/abs/2607.13987.

Kenneth J. Biba. Integrity considerations for secure computer systems. Technical Report ESD-TR-76-372, MITRE Corporation, 1977.

Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song, and Bo Li. AgentPoison: Red-teaming LLM agents via poisoning memory or knowledge bases. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

James Cheney, Laura Chiticariu, and Wang-Chiew Tan. Provenance in databases: Why, how, and where. *Foundations and Trends in Databases*, 1(4):379–474, 2009. doi: 10.1561/1900000006.

Yezhou Cheng, Runjia Du, Zeming Liu, Qibai Chen, Hang Lyu, Yankai Zeng, Yilan Wei, and Bojun Lin. Scope before you persist: Preventing cross-family interference in agent memory. *arXiv preprint arXiv:2609.29144*, 2026. URL https://arxiv.org/abs/2609.29144.

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. URL https://arxiv.org/abs/2505.23643.

Pritam Dash, Tongyu Ge, Aditi Jain, Tanmay Shah, and Zhiwei Shang. From untrusted input to trusted memory: A systematic study of memory poisoning attacks in LLM agents. *arXiv preprint arXiv:2606.04329*, 2026. URL https://arxiv.org/abs/2606.04329.

Edoardo Debenedetti, Jie Zhang, Mislav Balunović, 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.

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://satml.org/2026/program/.

Dorothy E. Denning. A lattice model of secure information flow. *Communications of the ACM*, 19 (5):236–243, 1976. doi: 10.1145/360051.360056.

Shen Dong, Shaochen Xu, Pengfei He, Yige Li, Jiliang Tang, Tianming Liu, Hui Liu, and Zhen J. Xiang. Memory injection attacks on LLM agents via query-only interaction. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Wenjie Fu, Xiaoting Qin, Jue Zhang, Qingwei Lin, Lukas Wutschitz, Robert Sim, Saravan Rajmohan, and Dongmei Zhang. CI-Work: Benchmarking contextual integrity in enterprise LLM agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 1483–1508. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-industry.103. URL https://aclanthology.org/2026.acl-industry.103/.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec '23)*, pp. 79–90. ACM, 2023. doi: 10.1145/3605764.3623985.

Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. *The Annals of Statistics*, 49(2):1055–1080, 2021. doi: 10.1214/20-AOS1991.

Shuhuai Huang, Jingfeng Zhang, and Hong Jia. When malicious instructions persist: Persistent memory poisoning attack on harness-based agents. *arXiv preprint arXiv:2609.13889*, 2026. URL https://arxiv.org/abs/2609.13889.

Aashish Kolluri, Rishi Sharma, Manuel Costa, Boris Köpf, Tobias Nießen, Mark Russinovich, Shruti Tople, and Santiago Zanella-Béguelin. Optimizing agent planning for security and autonomy. In *International Conference on Learning Representations*, 2026. URL https://www.microsoft.com/en-us/research/publication/optimizing-agent-planning-for-security-and-autonomy/.

Jiarui Li, Jiahao Chen, Chunyi Zhou, Yuwen Pu, Oubo Ma, Zhou Feng, Chunqiang Hu, and Shouling Ji. A finger on the scale: Covert policy steering through agentic skills. *arXiv preprint arXiv:2609.02564*, 2026a. URL https://arxiv.org/abs/2609.02564.

Xiangyi Li, Wenbo Chen, Yimin Liu, Shenghan Zheng, Xiaokun Chen, Yifeng He, Yubo Li, Bingran You, Haotian Shen, Jiankai Sun, Shuyi Wang, Qunhong Zeng, Di Wang, Xuandong Zhao, Yuanli Wang, Roey Ben Chaim, Zonglin Di, Yipeng Gao, Junwei He, Yizhuo He, Liqiang Jing, Luyang Kong, Xin Lan, Jiachen Li, Songlin Li, Yijiang Li, Yueqian Lin, Xinyi Liu, Xuanqing Liu, Haoran Lyu, Ze Ma, Bowei Wang, Runhui Wang, Tianyu Wang, Wengao Ye, Yue Zhang, Hanwen Xing, Yiqi Xue, Steven Dillmann, and Han-chung Lee. SkillsBench: Benchmarking how well agent skills work across diverse tasks. *arXiv preprint arXiv:2602.12670*, 2026b. URL https://arxiv.org/abs/2602.12670.

Zhiwei Li and Yong Hu. SkillHone: A harness for continual agent skill evolution through persistent decision history. In *Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing: Industry Track*, 2026. URL https://arxiv.org/abs/2606.08671. Accepted; forthcoming.

Huawei Lin, Peng Li, Jie Song, Fuxin Jiang, and Tieying Zhang. MUSE-Autoskill: Self-evolving agents via skill creation, memory, management, and evaluation. *arXiv preprint arXiv:2605.27366*, 2026. URL https://arxiv.org/abs/2605.27366.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Yuxuan Liu, Zhaochen Su, Lingyun Xie, Yuhao Zhang, Qing Zong, Jiahe Guo, Zhongwei Xie, Yiyan Ji, Yauwai Yim, Hongyu Luo, Xiyu Ren, Chenyu Ruan, Haoran Li, and Yangqiu Song. SkillRevise: Improving LLM-authored agent skills via trace-conditioned skill revision. In *Findings of the Association for Computational Linguistics: EMNLP 2026*, 2026. URL https://arxiv.org/abs/2606.01139v4. Accepted; forthcoming.

Andrew C. Myers. JFlow: Practical mostly-static information flow control. In *Proceedings of the 26th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pp. 228–241, 1999. doi: 10.1145/292540.292561.

Ayan Roy and Kaustuvi Basu. MemSentry: A framework for detecting persistent memory poisoning in agentic AI. *arXiv preprint arXiv:2609.08747*, 2026. URL https://arxiv.org/abs/2609.08747.

Biswa Sengupta. Self-evolving agents with anytime-valid certificates. *arXiv preprint arXiv:2607.00871*, 2026. URL https://arxiv.org/abs/2607.00871.

Shoaib Ahmed Siddiqui, Radhika Gaonkar, Boris Köpf, David Krueger, Andrew Paverd, Ahmed Salem, Shruti Tople, Lukas Wutschitz, Menglin Xia, and Santiago Zanella-Béguelin. Permissive information-flow analysis for large language models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=ufYRO8y3mr.

Saksham Sahai Srivastava and Haoyu He. MemoryGraft: Persistent compromise of LLM agents via poisoned experience retrieval. *arXiv preprint arXiv:2512.16962*, 2025. URL https://arxiv.org/abs/2512.16962.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing*, pp. 368–377, 1999.

Jiayu Wang, Yifei Ming, Riya Dulepet, Qinglin Chen, Austin Xu, Zixuan Ke, Frederic Sala, Aws Albarghouthi, Caiming Xiong, and Shafiq Joty. LiveResearchBench: A live benchmark for user-centric deep research in the wild. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/998f8d8ff2697da2eae5f87143668754-Abstract-Conference.html.

Zexun Wang. Proof-carrying agent actions: Model-agnostic runtime governance for heterogeneous agent systems. *arXiv preprint arXiv:2606.04104*, 2026. URL https://arxiv.org/abs/2606.04104.

Qianshan Wei, Tengchao Yang, Yaochen Wang, Xinfeng Li, Lijun Li, Zhenfei Yin, Yi Zhan, Thorsten Holz, Zhiqiang Lin, and XiaoFeng Wang. A-MemGuard: A proactive defense framework for LLM-based agent memory. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=udqe7UZUZ6.

Xiaofang Yang, Ziqi Miao, Dianbo Sui, Jing Shao, and Lijun Li. Defense-as-Skill: Evolving runtime guard skill for skill-augmented agents. *arXiv preprint arXiv:2609.01487*, 2026. URL https://arxiv.org/abs/2609.01487.

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. doi: 10.18653/v1/2024.findings-acl.624. URL https://aclanthology.org/2024.findings-acl.624/.

Qiyan Zhao, Xiaofeng Zhang, Bo Liu, Minda Chen, Wei Xiong, Jingyang Chen, Guanting Ye, Wenhao Yu, Xiaosong Yuan, Shijie Han, Da-Han Wang, Jianmin Ji, Fei Huang, and Xu-Yao Zhang. Coalition-aware skill reliability for self-evolving agents. *arXiv preprint arXiv:2608.22610*, 2026. URL https://arxiv.org/abs/2608.22610.

Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. PoisonedRAG: Knowledge corruption attacks to retrieval-augmented generation of large language models. In *Proceedings of the 34th USENIX Security Symposium*, pp. 3827–3844, 2025. URL https://www.usenix.org/conference/usenixsecurity25/presentation/zou-poisonedrag.

APPENDIX

A NOTATION AND OPERATIONAL SEMANTICS

Table 5: Notation used in the main text and appendix. An object denotes an immutable version; changing its evidence or labels creates a new governance record.

Symbol	Meaning
$\mathcal{M}$, $D = (V, E)$	Persistent store and acyclic object-dependency graph
m, d, s, o	General object, derived object, source object, output object
$h(m)$, $\mathcal{P}(m)$	Immutable content-version identity; recorded provenance and audit dependencies
ι, τ, σ	Integrity-risk, epistemic-trust, and sensitivity labels
$\rho(m)$	Empirical ingress-risk score, distinct from $\iota(m)$
$t_{\mathrm{ver}}(m)$, $q(m)$	Last verification time; scope constraints for permitted uses
G, O, u, t	Goal, governed observations, use contract, current time
$g, \ell, v, K, \nu, \beta$	Task family, use role, destination, allowed capabilities, resource versions, compatibility predicate
$\mathrm{Dep}(d)$, $\mathrm{Dep}_a(d)$	Recorded audit dependencies; constraint-bearing parents unresolved on axis a
$U_a(d)$	Source ancestors unresolved on axis a
$\mathcal{C}_{\mathrm{loc}}(d)$, $\mathcal{W}(d)$	Local certificates; inherited certificate witness closure
$h_d, h_S, a, E_c, L_c, \mathcal{U}_c$	Certificate subject/support versions, label axis, discharged edges, authorized bound, permitted contracts
$j_c, \kappa_c, T_c, \chi_c$	Certificate issuer, policy epoch, expiry, and assurance type
Check, Current	Certificate applicability; current evidence for an object and use
ScopeOK, A, $\mathcal{M}_A$	Scope predicate, recall admissibility, admissible store subset
P, P_v, ρ_P, Δ_P	Recall/destination policy, empirical risk threshold, and freshness limit
$r_{\mathrm{ctx}}, r_{\mathrm{rel}}, \lambda_{\mathrm{risk}}, \eta$	Context ranking score, relevance score, risk penalty weight, residual-need tolerance
$Z^\star, S_{\mathrm{irr}}, C_\pi$	Task unknowns, irrelevant state, context induced by policy π
$\epsilon, R(\epsilon)$	Population entropy tolerance and AVS exposure frontier
$\mathcal{C}, \mathcal{J}, B_{cj}$	Certificates, use events, fixed certificate–use incidence
N, C, n_c, L	Number of uses, number of groups, group size, grouped bad-use count
F_c, p_c	Semantic certificate-failure event and bound
V_j, w_j	Inadmissible use event and its loss weight
W_c, H_{cert}	Certificate exposure weight and cumulative certificate-induced loss
$\mathcal{H}_{k-1}, b_k, \alpha_k$	Pre-issuance history, predictable exposure cap, conditional error bound

The graph records both derivation and audit dependencies, while the witness closure records why a restriction was relaxed. Removing a constraint from $\mathrm{Dep}_a(d)$ does not erase its original edge from $\mathrm{Dep}(d)$. Support objects named in a certificate are immutable audit dependencies and must be available to the checker; their own witness histories therefore enter $\mathcal{W}(d)$ through Eq. (5). Support versions must precede the new governance record, so a certificate cannot justify itself through a support cycle. Source authority comes from the policy registry rather than from a textual claim made by the source.

The mechanical checker accepts a certificate only if all required checks pass:

$$\begin{aligned}\mathrm{Check}(c, u, t) = \ & \mathrm{SignatureOK}(c) \wedge \mathrm{IssuerOK}(j_c, a, \chi_c, E_c, L_c, u) \\ & \wedge \mathrm{BindingsOK}(h_d, h_S, E_c, L_c) \wedge (u \in \mathcal{U}_c) \\ & \wedge (\kappa_c = \kappa(t)) \wedge (t \le T_c) \wedge \neg \mathrm{Revoked}(c, t). \end{aligned} \tag{16}$$

$\mathrm{BindingsOK}$ verifies the actual subject and support versions, not just the presence of fields in a JSON record. The epoch $\kappa(t)$ is controlled by policy. The issuer check includes the assurance type χ_c: an issuer may be authorized to verify a claim without being authorized to declassify its source. Certificate bounds must be jointly compatible with the unresolved-parent bounds. If their intersection is empty, the proposed label assignment is rejected. Expiry and revocation stop future admissions; they cannot undo a disclosure that has already occurred.

For an executable request, the control plane separately checks that the requested operations lie in K and that the environment and composition satisfy ν and β. A signature on a procedural draft does

not grant capabilities. This distinction is consistent with structured skill packages, lifecycle security analysis, and task-conditioned runtime guards (Li et al., 2026b; Badhe & Tiwari, 2026; Yang et al., 2026). The core DRSC-Bench results stop at persistent-draft eligibility; the supplementary sandbox in Appendix K.1 evaluates only simple analytical functions and pairwise compositions.

B Proofs and Boundary Cases

B.1 Proof of Proposition 1

Take a topological ordering of D. For an external source s, the claim follows from $U_a(s) = \{s\}$. Assume the inequalities hold for every parent of a derived object d. For integrity risk,

$$\begin{aligned}\iota(d) &\succeq \bigvee_{x \in \mathrm{Dep}_\iota(d)} \iota(x) \succeq \bigvee_{x \in \mathrm{Dep}_\iota(d)} \bigvee_{s \in U_\iota(x)} \iota(s) \\ &= \bigvee_{s \in \bigcup_{x \in \mathrm{Dep}_\iota(d)} U_\iota(x)} \iota(s) = \bigvee_{s \in U_\iota(d)} \iota(s). \end{aligned} \tag{17}$$

The same monotonicity and associativity argument applies to sensitivity. For trust,

$$\tau(d) \preceq \bigwedge_{x \in \mathrm{Dep}_\tau(d)} \tau(x) \preceq \bigwedge_{x \in \mathrm{Dep}_\tau(d)} \bigwedge_{s \in U_\tau(x)} \tau(s) = \bigwedge_{s \in U_\tau(d)} \tau(s). \tag{18}$$

These steps also cover empty parent sets using the lattice identities. The explicit unsupported-generation cap can only lower trust, so it cannot invalidate the upper bound. Induction proves the algebraic invariant. Complete governed dependencies ensure that an omitted influence cannot silently evade the graph. Sound, axis-specific certificates justify each removed constraint; the invariant then covers the remaining source obligations. Neither condition follows from the algebra itself.

For example, suppose a public but untrusted source s produces summary d_1, which produces report evidence d_2. With both trust edges unresolved, $\tau(d_2) \preceq \tau(d_1) \preceq \tau(s)$. An independent endorsement of d_1 can discharge its trust edge to s while its sensitivity and integrity edges remain. A descendant relying on that endorsement retains the endorsement in $\mathcal{W}(d_2)$. If it expires or does not cover the later use, the descendant cannot retain admissibility solely because its cached trust label is high.

The empty-set case deserves separate treatment. If a model writes an unsupported claim with no recorded trust parents, $\bigwedge \varnothing = \mathsf{trusted}$ merely makes the parent inequality vacuous. Assigning that value as the object's label would confuse a mathematical identity with evidence. The unsupported-generation cap avoids that mistake without forcing every summary of verified parents down to provisional trust. In all cases the label concerns allowed reliance under policy, not a claim that the generation process cannot hallucinate.

B.2 Proof of Proposition 2

Let T be the observed metadata transcript in both executions. A deterministic policy has one value $\Pi(T)$, so it cannot make different decisions. For a randomized policy, write its output as $\Pi(T, \xi)$ with random seed ξ drawn independently from the same distribution in both executions. Its decision law is then identical. The stronger true constraint in one execution is absent from the observed input. Always rejecting both executions is compatible with safety; distinguishing them while retaining permissiveness requires another observation. This argument does not imply a numerical lower bound on attack success or a matching lower bound for a particular attribution algorithm.

B.3 Closure under certificate invalidation

Proposition 5 (Inherited-witness invalidation)**.** *Suppose every derived object retains Eq.* (5)*, including witnesses for supporting certificates. Also suppose every admission checks* $\mathrm{Current}$ *atomically with respect to the policy snapshot. If a certificate in* $\mathcal{W}(d)$ *fails for* (u, t)*, object* d *and every descendant that inherits this witness fail admission until a new justified record is established.*

Proof. The failed certificate makes the conjunction defining $\mathrm{Current}(d, u, t)$ false. For any child y with $d \in \mathrm{Dep}(y)$, Eq. (5) gives $\mathcal{W}(d) \subseteq \mathcal{W}(y)$, so the same certificate makes $\mathrm{Current}(y, u, t)$ false.

Induction over descendants proves the claim. Atomic check and use prevent an epoch change between validation and admission. A fresh derivation from valid support can establish a usable replacement. Adding a certificate to an unchanged descendant cannot remove a failed witness from its existing closure. □

The closure is conservative. It may retain a witness whose influence a stronger provenance analysis could prove irrelevant. This affects availability, not the stated invalidation property. The proposition specifies a runtime property; the empirical tables do not measure revocation latency, source withdrawal, or version migration. Revocation cannot eliminate the need to check a new artifact's factual or operational adequacy.

B.4 Proof of Proposition 3

For each use event, the certificate-induced violation assumption and the union bound imply

$$\Pr(V_j) \le \Pr\left(\bigcup_{c:B_{cj}=1} F_c\right) \le \min\left\{1, \sum_c B_{cj} p_c\right\}. \tag{19}$$

Multiply by w_j and sum to obtain the first inequality in Eq. (12). The inequality $\min\{1, z\} \le z$ and exchange of finite sums give

$$\mathbb{E}[H_{\mathrm{cert}}] \le \sum_j \sum_c w_j B_{cj} p_c = \sum_c p_c \sum_j w_j B_{cj} = \sum_c p_c W_c. \tag{20}$$

No product of certificate-error probabilities occurs, so independence is unnecessary. The bound may be loose when several erroneous certificates affect the same use. It is not a calibration method for the p_c, nor a guarantee that reducing context size reduces certificate error.

If one certificate failure F with probability p causes all N dependent uses to fail, the count $N_F = N\mathbf{1}[F]$ obeys

$$\Pr(N_F > 0) = p, \qquad \mathbb{E}[N_F] = Np. \tag{21}$$

When failure only makes these uses potentially inadmissible, the corresponding statements are upper bounds rather than equalities. Independent fresh checks represent a different experiment. Their probability of at least one failure equals $1 - (1-p)^N$, while their expected count is also Np.

B.5 Adaptive reuse with predictable exposure caps

Dependence between error and exposure can invalidate a tempting extension of Eq. (20). For a single certificate let $\Pr(F) = p$ and let the realized exposure be $W = N\mathbf{1}[F]$. This coupling can arise if faulty content changes which tasks reuse it; the agent need not observe the hidden error label directly. Then

$$\mathbb{E}[\mathbf{1}[F]W] = Np, \qquad p\,\mathbb{E}[W] = Np^2. \tag{22}$$

The second expression underestimates the first when $0 < p < 1$. Fixed-graph marginal error bounds therefore cannot be applied to unrestricted adaptive reuse by substitution.

Proposition 4 uses a predictable cap on total downstream exposure, rather than on the immediate recall alone. Its conditional premise is evaluated before issuance. For countably many decisions, all sums below are nonnegative. The proof does not require exposure to be independent of semantic error.

Proof. The event-containment assumption gives $H_{\mathrm{cert}} \le \sum_k \mathbf{1}[F_k] W_k \le \sum_k \mathbf{1}[F_k] b_k$. Conditional expectation and predictability give

$$\mathbb{E}[\mathbf{1}[F_k] b_k] = \mathbb{E}[b_k \Pr(F_k \mid \mathcal{H}_{k-1})] \le \mathbb{E}[b_k \alpha_k]. \tag{23}$$

Summation proves Eq. (14); Tonelli's theorem handles countably many nonnegative terms. The budget implication follows by taking expectations. For weights at least one, any counted violation implies $H_{\mathrm{cert}} \ge 1$, and Markov's inequality proves the probability bound. □

The conditional premise is stronger than marginal risk control on a fixed distribution. In particular, neither a fixed-sample risk bound nor a confidence sequence automatically yields a per-issuance conditional error bound for adaptively selected certificates. Issuing additional certificates in response to attacks, changing the support distribution, or reusing a calibration set can invalidate it. Finite-sample risk-control procedures and time-uniform intervals address related inferential requirements under their own sampling assumptions (Angelopoulos et al., 2025; Howard et al., 2021), but they are not instantiated by DRSC-Bench. The result therefore does not supply such calibration or an optimal allocation of verification effort. It identifies an enforceable accounting condition once valid conditional bounds are available. Certificate-gated agent updates and anytime-valid validation are related concerns in SEA (Sengupta, 2026); the accounting here concerns the downstream reuse of a particular endorsement.

C Distribution of Loss Under Shared Certificates

In the grouped model of Eq. (15), set $X_c = \mathbf{1}[F_c]$ and assume the X_c are independent Bernoulli variables with parameter p. Group sizes n_c are positive integers and sum to N. Every use belongs to exactly one group and fails precisely when that group's certificate is wrong. This assumption is stronger than the event-containment condition of Proposition 3. For the loss count $L = \sum_c n_c X_c$, the probability-generating function is

$$\mathbb{E}[z^L] = \prod_{c=1}^{C} \left[(1-p) + pz^{n_c}\right]. \tag{24}$$

Linearity of expectation gives Np. Independence makes all covariance terms zero, yielding $p(1-p)\sum_c n_c^2$. Since every group is nonempty, $L = 0$ if and only if every certificate is correct, giving the final identity in Eq. (15). Consequently,

$$\mathbb{E}[L \mid L > 0] = \frac{Np}{1-(1-p)^C}. \tag{25}$$

For $0 < p < 1$, combining two groups of sizes a and b increases the loss variance by $2abp(1-p)$, leaves its expectation unchanged, and decreases the any-failure probability by $p(1-p)^{C-1}$. A lower any-failure probability can therefore coexist with greater conditional severity. A chain and a star with the same one-certificate incidence have identical distributions; graph geometry alone does not distinguish them in this model.

For a fixed group structure and independent error probabilities p_c, differentiation gives

$$\frac{\partial \mathbb{E}[L]}{\partial p_c} = n_c, \qquad \frac{\partial \Pr(L > 0)}{\partial p_c} = \prod_{d \neq c} (1 - p_d). \tag{26}$$

The first derivative depends on reuse count; the second does not. Optimizing the probability of any failure and optimizing expected affected uses generally require different comparisons. These derivatives hold only with the graph and other probabilities fixed, and do not specify what error reduction a real verification procedure can achieve at a given cost.

For $R_{\mathrm{MC}} = 2000$ independent Monte Carlo trials, the model-implied standard errors of the sample mean and empirical any-failure proportion are

$$\mathrm{SE}(\bar{L}) = \sqrt{\frac{p(1-p)\sum_c n_c^2}{R_{\mathrm{MC}}}}, \qquad \mathrm{SE}(\widehat{q}) = \sqrt{\frac{q(1-q)}{R_{\mathrm{MC}}}}, \quad q = 1-(1-p)^C. \tag{27}$$

These are analytical sampling uncertainties for the specified simulation model, not measured run standard deviations or agent-level confidence intervals.

A deployment-level application requires certificate-level semantic error measurements and a declared loss scale. Task-level STP combines issuance, exposure, policy, and task outcomes, so substituting it for p_c changes the estimand. An empirical error estimate is also not automatically an upper confidence bound. With adaptive reuse, a recorded incidence graph may depend on erroneous content, requiring the conditional accounting in Appendix B.5 or a separately justified dependence model.

Table 6: Exact expectations and analytical Monte Carlo standard errors for the configurations in Table 4. Every row has $\mathbb{E}[L] = 1$.

Configuration	Exact q	SE($\bar{L}$)	SE($\widehat{q}$)
Independent	0.6415	0.0218	0.0107
Five groups of four	0.2262	0.0436	0.0094
Star, one endorsement	0.0500	0.0975	0.0049
Chain, one endorsement	0.0500	0.0975	0.0049

D Agent View Sufficiency

Fix a policy environment and let $(M, O, G, Z^\star, S_{\text{irr}})$ follow a task distribution $\mathbb{P}$. Assume finite $H(Z^\star \mid O, G)$. The variable M represents available admissible state, $Z^\star$ the task-relevant unknowns, and S_{irr} irrelevant or contamination-prone state. A feasible policy π selects a context C_π using only admissible inputs. Define

$$R(\epsilon) = \inf_{\pi:\ H(Z^\star \mid O,G,C_\pi) \le \epsilon} I(C_\pi; S_{\text{irr}} \mid O, G), \tag{28}$$

with $R(\epsilon) = +\infty$ if no policy is feasible. The conditional information identity gives

$$I(C_\pi; Z^\star \mid O, G) = H(Z^\star \mid O, G) - H(Z^\star \mid O, G, C_\pi), \tag{29}$$

$$H(Z^\star \mid O, G, C_\pi) \le \epsilon \iff I(C_\pi; Z^\star \mid O, G) \ge H(Z^\star \mid O, G) - \epsilon. \tag{30}$$

If $\epsilon_1 \le \epsilon_2$, the feasible set at ϵ_1 is contained in that at ϵ_2, hence $R(\epsilon_2) \le R(\epsilon_1)$. If a constant empty context is allowed and $\epsilon \ge H(Z^\star \mid O, G)$, nonnegativity of mutual information gives $R(\epsilon) = 0$. This is the standard conditional information-bottleneck interpretation (Tishby et al., 1999). It supplies no direct relationship between mutual information and policy violation severity. Long-context utilization also depends on how models use the supplied evidence (Liu et al., 2024).

The implementation uses the empirical threshold η, not ϵ. Its residual-need score is selected on a validation split; it is not a Shannon-entropy estimate. Table 7 reports the operating sweep. Recall top-$k = 16$ counts recalled memory objects; context size counts the active object collection, including observation objects and task instructions. These denominators differ, so a size of 24.3 is not a claim that one top-16 retrieval returned 24.3 memory records. Object counts refer to the assembler's complete active view, rather than to memory recall alone.

Table 7: Empirical context-selection sweep on GPT-4o. η is the residual-need tolerance. Context size counts all active objects. The default is $\eta = 0.20$.

η	Utility ↑	CSC ↓	Mean context size
0.01	0.74	0.05	24.3
0.10	0.74	0.03	18.7
0.20	0.74	0.02	14.2
0.30	0.72	0.02	10.8
0.50	0.65	0.01	6.4

E Benchmark and Implementation

E.1 Tasks and attack construction

Domain experts selected 40 questions per domain from research queries and academic literature. Each task includes factual targets, required subtopics, public evidence, a simulated internal document, and adversarial pages. Reference answers use verified public sources. The simulated internal document contains a plausible marked field consistent with its domain but absent from the public sources. It can therefore be reliable evidence while remaining ineligible for public disclosure. External adversarial pages are annotated as high integrity risk and untrusted for evaluation; these gold annotations define correctness of decisions, not proof of the ingress detector's accuracy.

The agent first discovers sources, then synthesizes information and writes persistent state, and finally recalls that state for a delayed report of approximately 500–800 words. The attack must survive a persistence boundary to affect the CSC score. The clean comparison removes planted adversarial material while preserving the task. The protocol does not equate the three sessions with months of continuous deployment.

Each adversarial page starts with 200–300 words of correct, relevant material before introducing a payload. Indirect-injection payloads place instructions in HTML comments or low-visibility text. Memory-poisoning payloads assert false facts. Carry-over payloads associate otherwise unrelated concepts that can be tested later. Procedural-draft payloads contain a plausible procedure with a flaw. Exfiltration payloads request inclusion of a marked sensitive field in the report. This construction favors retrieval relevance; it does not establish coverage of arbitrary adaptive attackers, independently developed templates, or attacks on executable skill dependencies.

E.2 Metric denominators

For task i and run r, let $U_{ir} \in [0, 1]$ be Research Utility and let Y_{ir}^{UE}, Y_{ir}^{STP}, and Y_{ir}^{CSC} be the three binary core probe outcomes. With $n = 240$ and $n_{\mathrm{run}} = 5$,

$$\widehat{p}_k = \frac{1}{n\, n_{\mathrm{run}}} \sum_{i=1}^{n} \sum_{r=1}^{n_{\mathrm{run}}} Y_{ir}^{k}, \qquad k \in \{\mathrm{UE}, \mathrm{STP}, \mathrm{CSC}\}. \tag{31}$$

UE counts a destination-policy violation at the designated release opportunity; a blocked release is not a violation. STP counts a designated low-trust write that becomes eligible for higher-trust reuse without adequate verification. For systems lacking an explicit trust lattice, the evaluator uses source annotations and observed reuse behavior. CSC counts disagreement between the delayed scored answers in the attack and clean conditions. The pairing preserves the task; stochastic answer variation can also contribute to disagreement. A clean–clean control repeats the delayed answer without adversarial content; it measures background variation rather than a treatment effect.

Table 8: Clean–clean delayed-answer variation on GPT-4o. Each value is a proportion over the same 1,200 allocated task-run opportunities. The control does not contain planted adversarial material.

Configuration	Clean–clean answer change
Naive RAG	0.02
A-MemGuard	0.01
Fides + persistent labels	0.01
Full governance	0.01

Relative to Table 8, the difference between attack–clean and clean–clean disagreement rates is approximately 0.22 for Naive RAG, 0.10 for A-MemGuard and Fides with persistent labels, and 0.01 for full governance. These differences are descriptive because the clean control shares task identities but is not a randomized crossover at every run seed.

Research Utility averages exact-match accuracy on ten factual questions, required-subtopic coverage, and report quality. The quality rubric covers correctness, coverage, coherence, and citation quality. Human agreement for the quality component is evaluated separately in Appendix L.2. Factual questions and subtopic coverage retain their own scoring rules; agreement for one utility component does not validate the complete metric.

Recall precision uses recalled objects that are both relevant and admissible as its numerator and all recalled objects as its denominator. Recall uses the relevant admissible set as its denominator. Re-entry block uses annotated inadmissible retrieval candidates. PWA measures label correctness among writes; SCB uses contaminated procedural drafts; FP uses clean candidate writes. These diagnostic denominators are distinct from the fixed task probes in Eq. (31). In particular, $1 - \mathrm{PWA}$ need not equal STP, and $1 - \mathrm{SCB}$ need not equal the attack-family success rate. No-gate FP is zero because the mechanism never downgrades a write.

Task-level joint success is

$$\mathrm{SafeSuccess}_{0.7} = \frac{1}{n\, n_{\mathrm{run}}} \sum_{i,r} \mathbf{1}[U_{ir} \geq 0.7] \prod_{k} (1 - Y_{ir}^{k}). \tag{32}$$

The union of the three violation events is counted once per task-run. Therefore SafeSuccess cannot exceed one minus the any-violation rate. Neither mean utility nor a metric conditional on completed actions determines this joint quantity.

E.3 Implementation settings

The middleware wraps memory writes, recall, context assembly, and export. It uses a SQLite provenance store and a JSON policy engine; certificates are signed records. Ingress assigns a source class using a curated reputation database covering 12,000 domains. The evidence score combines cross-source agreement, citation support, and signature information. For ordinal rank map r_τ and evidence score $z_{\mathrm{ev}}(x) \in [0, 1]$, the assignment rule is

$$\tau(x) = r_\tau^{-1}(\mathrm{clip}\left[r_\tau(\tau_s(s)) + \delta_{\mathrm{ev}}(z_{\mathrm{ev}}(x)), 0, r_\tau(\mathsf{verified})\right]), \tag{33}$$

where $\delta_{\mathrm{ev}} \in \{-1, 0, 1\}$ is an integer-valued policy mapping. An ingress adjustment supplies a label under this evidence policy; it does not certify factual correctness.

Relevance comes from a cross-encoder. The empirical selector uses $\lambda_{\mathrm{risk}} = 0.5$, default $\eta = 0.2$, a 32-object context budget, and top-16 memory recall. Main runs use GPT-4o at temperature zero. Risk and age thresholds are $\rho_P = 0.2$ and $\Delta_P = 30$ days. The matched backbone comparison uses Claude 3.5 Sonnet and Llama 3.1 70B with the same task manifests, governance settings, and search index. Five repeated executions are reported. Temperature zero alone does not establish deterministic API behavior or identify the source of run variation.

Selective attribution and empirical endorsement are fallible components of this implementation. The interface specification defines the required checks, while the tests measure their behavior under the fixed benchmark conditions. The evaluation does not identify the effects of provider updates, different risk or relevance checkpoints, or changes to the curated reputation database.

F Counts and Statistical Interpretation

Table 9: Core event counts. Each cell has 1,200 allocated task-run opportunities. Counts correspond to the first five rows of Table 1; the 240 task identities recur across five runs.

Configuration	UE events	STP events	CSC events
Naive RAG	215	372	288
CaMeL-style	50	264	228
Fides adaptation	25	240	192
A-MemGuard	110	145	132
Full governance	15	36	24

Pooling is used in Table 9 to display counts, not to assert independence. Since each run has the same denominator, the pooled proportion equals the mean of per-run proportions. Thus full-governance UE is exactly $15/1200 = 0.0125$ in these records, and the Naive RAG rate is $215/1200$. A relative reduction calculated from the rounded values 0.01 and 0.18 would be 94.4%; the count-based reduction is instead

$$1 - \frac{15/1200}{215/1200} = \frac{200}{215} \approx 0.9302. \tag{34}$$

The latter is used in the main text. The full-versus-Fides absolute UE difference is $(25 - 15)/1200 = 0.00833$. Counts for the ablations are not inferred from their rounded rates.

Statistical inference uses task identities as clusters (Appendix L), retaining all five runs and paired methods within each selected identity. Marginal counts alone do not recover discordant pairs, so count-based rates and paired intervals serve different purposes. Their agreement in point estimates is checked directly. The intervals condition on the observed runs and do not quantify uncertainty from shared run-level shocks or source/template clusters that cross task identities.

Table 10 reports joint task outcomes. They are compatible with the corresponding marginal bounds: $\max_k \widehat{p}_k \le \widehat{p}_{\mathrm{any}} \le \min\{1, \sum_k \widehat{p}_k\}$ up to rounding. Their ordering differs from the utility ordering because joint success depends on both the utility threshold and the overlap of violation events. Correlation between utility and violations cannot be estimated from these summaries alone.

Table 10: Joint task outcomes for the core experiment, with SafeSuccess defined by Eq. (32). Any violation is the union of the UE, STP, and CSC events on a task-run. Values are rounded proportions.

Configuration	$\text{SafeSuccess}_{0.7} \uparrow$	Any violation $\downarrow$
Naive RAG	0.55	0.41
CaMeL-style	0.60	0.27
Fides adaptation	0.58	0.25
A-MemGuard	0.63	0.19
Full governance	0.71	0.06

G Persistent-Label Comparison and Opportunity Coverage

The persistence-matched evaluation compares the Fides adaptation with a configuration that serializes labels through the benchmark's memory store. The PRUDENTIA-style row additionally uses policy-aware planning to choose among admissible actions under the same labels. Table 2 reports means at their measured precision. Rounded rates are not converted into event counts. The persistence-matched records are evaluated separately from the count-based core comparison; their own paired intervals appear in Appendix L. Verifier identities, internal calls, and budgets are not controlled well enough to assign the gap to one mechanism.

Label serialization lowers STP by 0.06 and CSC by 0.05 relative to the tested Fides adaptation. The PRUDENTIA-style planner recovers 0.03 utility and 0.04 coverage relative to persistent-label Fides while lowering STP and CSC by an additional 0.01 each. The remaining differences from full governance are 0.10 STP and 0.08 CSC. These differences and their paired intervals do not isolate the effect of the three-axis representation or certificate semantics from other implementation differences. The comparison also shows why describing Fides itself as unable to track persistent labels would be inaccurate.

Coverage for Naive RAG, CaMeL-style, and A-MemGuard is 1.00, 0.82, and 0.88, with rejection 0.00, 0.18, and 0.12. For the rows of Table 2, rejection is 0.22, 0.20, 0.16, and 0.04. In the write diagnostic, accepted-write coverage is 1.00 without a gate, 0.91 with static labels, and 0.94 with governed writes. These measurements make abstention visible, but neither opportunity coverage nor accepted-write coverage is a completed-task success rate. In particular, higher PWA with coverage 0.94 cannot be described as having no cost in the number of accepted writes.

H Recall, Write, and Attribution Diagnostics

Recall precision and recall use annotated relevant admissible objects; re-entry block uses annotated inadmissible candidates. Write-label accuracy uses candidate writes, contaminated-draft blocking uses contaminated drafts, and false positives use clean writes. The no-gate false-positive rate is zero because it never downgrades a write.

Table 11: Mechanism diagnostics on GPT-4o. Values are proportions. RB: inadmissible re-entry blocked; PWA: persistent-write label accuracy; SCB: contaminated procedural draft blocked; FP: unnecessary downgrade of a clean write. The no-gate FP is zero by construction.

Recall mechanism	Precision $\uparrow$	Recall $\uparrow$	RB $\uparrow$
Semantic recall	0.61	0.85	0.08
Threshold filter	0.74	0.78	0.52
A-MemGuard	0.79	0.76	0.68
Governed recall	0.88	0.82	0.94
Write mechanism	**PWA $\uparrow$**	**SCB $\uparrow$**	**FP $\downarrow$**
No gate	0.69	0.05	0.00
Static labels	0.78	0.41	0.18
Governed writes	0.93	0.91	0.04

Table 12: Controlled stress tests on GPT-4o. The omission intervention removes an influencing source from selective attribution. Conservative capture records all visible governed inputs. These interventions probe assumptions; they are not demonstrations that an attacker can rewrite trusted metadata.

Condition	Utility ↑	CSC ↓	STP ↓
Default attacks, selective attribution	0.74	0.02	0.03
Omission, selective attribution	0.73	0.09	0.04
Omission, conservative capture	0.71	0.04	0.03
Promotion probing	0.74	0.02	0.06

The omission intervention changes the evaluator's attribution record rather than the attacker's access to trusted metadata. Under this intervention, conservative capture lowers CSC by 0.05 at a utility cost of 0.02 relative to selective attribution. Promotion probing tests empirical endorsement under repeated low-trust variants. Its STP rate is computed over task probes, not over independently labeled certificates. Both diagnostics examine departures from semantic assumptions used in the formal guarantee.

I Domain and Attack Breakdowns

Table 13: Full-governance results by domain on GPT-4o. Each domain contains 40 tasks. Rounded zero UE does not establish a zero population failure rate.

Domain	Utility	UE
Science	0.73	0.01
Technology	0.75	0.01
Medicine	0.71	0.00
Finance	0.73	0.02
Law	0.74	0.01
History	0.78	0.01

Table 13 shows that domain utility ranges from 0.71 to 0.78. The mean of the six displayed utility values is 0.74. The displayed domain UE rates have two-decimal precision. Domain event counts are needed to reconcile them with the aggregate 15/1,200; these entries do not establish a domain ordering in export risk.

Table 14: Attack-family success on GPT-4o. Cross-session carry-over uses the CSC task outcome. Exfiltration success within its attack family has a different denominator from aggregate UE.

Attack family	Naive RAG	Full governance
Indirect prompt injection	0.28	0.05
Memory poisoning	0.35	0.02
Cross-session carry-over	0.24	0.02
Procedural-draft contamination	0.21	0.03
Restricted-data exfiltration	0.32	0.01

In Table 14, the largest absolute reduction is memory poisoning, 0.33. Restricted-data exfiltration has the largest relative reduction, approximately 96.9%. The largest residual attack-family success rate is indirect injection, 0.05. These rates cannot be averaged to recover aggregate UE, STP, or CSC: the event definitions and denominators differ. The family results also do not measure complete installed-skill lifecycles.

J Matched Backbone Results

The relative CSC reductions in Table 15, calculated from the displayed means, are 91.7%, 88.5%, and 74.2%. Residual CSC is largest on Llama 3.1 70B, at 0.08. Shared governance settings do not

Table 15: Matched ungoverned and governed configurations in the core experiment. Entries are mean ± standard deviation over five runs. Rate rounding is retained at the available precision; 0.00 standard deviation denotes rounding, not exact constancy.

Backbone	Configuration	Utility	UE	STP	CSC
GPT-4o	Naive	$.72 \pm .01$	$.18 \pm .02$	$.31 \pm .02$	$.24 \pm .02$
	Governed	$.74 \pm .01$	$.01 \pm .00$	$.03 \pm .01$	$.02 \pm .01$
Claude 3.5 Sonnet	Naive	$.70 \pm .02$	$.21 \pm .02$	$.33 \pm .02$	$.26 \pm .02$
	Governed	$.72 \pm .01$	$.02 \pm .01$	$.04 \pm .01$	$.03 \pm .01$
Llama 3.1 70B	Naive	$.61 \pm .02$	$.27 \pm .02$	$.38 \pm .03$	$.31 \pm .02$
	Governed	$.65 \pm .02$	$.05 \pm .02$	$.09 \pm .02$	$.08 \pm .02$

make model-family differences causal evidence about any one component. The experiment does not separately measure each backbone's ingress mistakes, lineage omissions, or endorsement errors.

K Ten-Session and Skill Protocols

The ten-session condition extends 60 research tasks. Each session introduces public evidence, one adversarial page, and one restricted field; the clean counterpart omits adversarial content throughout. Each configuration is executed three times with selective attribution. Table 3 reports utility and CSC at the session-10 endpoint. The 60-task condition must be compared within its own task set: the core session-3 mean across 240 tasks is not a paired earlier point on the same trajectory. The extension does not measure revocation or conservative capture.

K.1 Bounded skill execution

The sandbox contains twelve drafted Python procedures operating on structured objects. Each seed supplies forty single-skill requests and eighteen pairwise-composition requests. The trusted control plane checks the requested operations against the allowed capability set K; labels and current support determine whether the artifact is admissible for the requested use. Scripts have no unrestricted shell or network access. The evaluation distinguishes procedural-draft admission in the core benchmark from execution of these bounded functions.

Table 16: Sandbox outcomes, averaged over three seeds. Single-skill failure is an incorrect result or a capability violation among forty single-skill requests per seed. Unsafe capability is a violation among all 58 allocated requests per seed. Means are shown at the measured precision.

Configuration	Single-skill failure	Unsafe capability
No governance	0.35	0.22
A-MemGuard	0.22	0.14
Full governance	0.06	0.01

The denominator includes allocated requests, which prevents a violation rate conditional on execution from concealing rejected requests. Successful completion and execution coverage are separate quantities. Without those outcomes, lower violation rates do not establish a superior execution safety–utility trade-off. The capability measurement includes requests that compose procedures; it does not establish the correctness of their combined behavior or generalize to installed skills with external dependencies.

L Paired Bootstrap and Human Evaluation

L.1 Paired bootstrap confidence intervals

The paired bootstrap samples task identities: resample the 240 task identities with replacement, retain their five runs and both compared methods, and recompute the mean difference. The 2.5th and 97.5th

percentiles give the interval. Table 17 reports the full-governance comparison against Naive RAG, A-MemGuard, and the persistence-matched baselines.

Table 17: Paired bootstrap results, 10,000 resamples. Δ is Full minus comparison method. Negative Δ for failure rates favors governance; positive Δ for utility favors governance.

Comparison	Metric	Δ	95% CI
Full vs Naive	UE	-0.167	$[-0.19, -0.14]$
Full vs Naive	CSC	-0.220	$[-0.25, -0.19]$
Full vs Naive	STP	-0.280	$[-0.31, -0.25]$
Full vs Naive	Utility	$+0.020$	$[+0.005, +0.035]$
Full vs A-MemGuard	STP	-0.091	$[-0.11, -0.07]$
Full vs A-MemGuard	CSC	-0.090	$[-0.11, -0.07]$
Full vs Fides+persist	STP	-0.110	$[-0.14, -0.08]$
Full vs Fides+persist	CSC	-0.090	$[-0.12, -0.06]$
Full vs Fides+persist	Utility	$+0.080$	$[+0.05, +0.11]$
Full vs PRUDENTIA-style+persist	STP	-0.100	$[-0.13, -0.07]$
Full vs PRUDENTIA-style+persist	CSC	-0.080	$[-0.11, -0.05]$
Full vs PRUDENTIA-style+persist	Utility	$+0.050$	$[+0.02, +0.08]$

All failure-rate intervals in Table 17 exclude zero. The utility interval against Naive RAG is positive but close to zero. These are pointwise intervals for the displayed comparisons, without a simultaneous-coverage adjustment. Keeping each task's five runs together retains within-task run dependence. It does not resample seed-level effects shared across tasks, or dependence induced by common sources and attack templates. The intervals therefore condition on the evaluated runs and task-cluster sampling model. No interval is provided here for the gate ablations or the full-versus-Fides core UE contrast.

L.2 Human evaluation of report quality

Agreement between the LLM quality judge and two human raters is assessed on 100 tasks. Raters score factual accuracy, coverage, coherence, and citation quality on a 1–5 scale. Report-quality summaries use normalized scores in $[0, 1]$; weighted agreement is computed on the ordinal 1–5 ratings.

Table 18: Report-quality agreement on 100 tasks. Pearson r is between the LLM judge and the mean of two human raters. Quadratic-weighted Cohen's κ is between the two human raters on ordinal 1–5 scores.

Quantity	Value
LLM vs human-mean Pearson r	0.85
Inter-rater Pearson r	0.82
Inter-rater quadratic-weighted Cohen's κ	0.71
Report quality (LLM) mean $\pm$ std	0.74 ± 0.09
Report quality (human) mean $\pm$ std	0.72 ± 0.10

The LLM judge's correlation with the human mean is 0.85; inter-rater correlation is 0.82 and quadratic-weighted κ is 0.71. These quantities use different comparison units and are not directly ranked as judge accuracy. The LLM mean score is 0.02 higher than the human mean. Correlation does not establish probabilistic calibration, agreement on safety events, or preservation of the ordering of methods; those require separate analyses.

M LATENCY

Table 19: End-to-end task latency in seconds for the core GPT-4o evaluation. The overhead column compares medians with Naive RAG.

Configuration	P50	P95	P99	P50 overhead
Naive RAG	41.8	67.3	84.1	–
CaMeL-style	44.2	71.8	89.5	5.7%
Fides adaptation	45.6	74.1	92.3	9.1%
A-MemGuard	45.1	73.2	91.0	7.9%
Full governance	46.9	75.2	93.8	12.2%

The quantiles in Table 19 are descriptive. They cannot be decomposed into per-component costs by subtraction, because quantiles of sums need not equal sums of quantiles. The observed metadata footprint was approximately 30% of stored content bytes in this implementation; this descriptive ratio depends on object granularity and serialization. No measured budget for adaptive revalidation, certificate revocation, or executable-skill sandboxing is included.