Provenance Is Not Influence: The Semantic-Information Flow Gap in LLM Agents

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Model (LLM) agents can transform untrusted external information through neural regeneration before using it to act, potentially decoupling semantic influence from the provenance observable at the authorization interface. We term this decoupling the **S**emantic-**I**nformation **F**low **G**ap **(SIFG)**: source-derived, action-relevant information can persist even when its source provenance is no longer observable. We characterize **SIFG** by separately measuring security-relevant semantic survival, authorization-visible provenance retention, and counterfactual action dependence, and show that this decoupling persists across the evaluated models, agent workflows, action settings, and transmission channels. We then formalize an observational boundary for provenance-only enforcement: when an attack and a benign execution induce the same gate-visible observation, no rule restricted to that observation can distinguish them. In the evaluated settings, provenance-only policies therefore face a security–utility tension, while additional provenance signals do not expose the missing distinction and an action-semantic signal can expose it. These results suggest that secure authorization after neural regeneration should treat provenance as distinct from semantic influence and incorporate policy-relevant signals beyond provenance alone. Our code is available at https://anonymous.4open.science/r/SIFG.

1 INTRODUCTION

Large Language Model (LLM) agents (Nous Research, 2026; Chase, 2023; DeepSeek-AI, 2026; OpenClaw, 2026) retrieve information from external sources, such as web pages, databases, and files, and use it to construct downstream tool actions. When those sources are untrusted, the security question is not only whether the information is present, but whether it can continue to influence an action after the agent has transformed it. This makes the propagation of source-derived information across the agent's internal transformations a central information-flow security problem.

Information-flow security is well established in computer systems (Denning, 1976; Sabelfeld & Myers, 2003; Enck et al., 2010). Traditional mechanisms track how information originates, propagates, and influences security-sensitive operations, while provenance provides a complementary abstraction for representing the causal lineage of data and execution artifacts (Buneman et al., 2001; Green et al., 2007; Thalheim et al., 2016a; Han et al., 2018). As LLM agents increasingly operate over retrieved content, tool outputs, memory, and other agents' messages, recent agent-security research has likewise adopted provenance and related information-flow mechanisms to track untrusted information and enforce trust at downstream actions (Wu et al., 2024; Costa et al., 2025; Souza et al., 2025; Jia et al., 2025; Zhu et al., 2025). An important difference in the LLM-agent setting is that, rather than merely propagating artifacts whose lineage can be explicitly tracked, agents repeatedly regenerate information through neural computation, so semantic content can persist without an explicit artifact-level ancestry relation being retained (Cai et al., 2026; Zhu et al., 2025; Wu et al., 2024). Accordingly, we ask: *does provenance remain a faithful proxy for semantic influence when information is transformed through LLM-mediated regeneration?*

We show that this correspondence can break: source-derived information can remain action-relevant after neural regeneration even when its source provenance is no longer observable at authorization. We call this decoupling the ***Semantic-Information Flow Gap*** **(SIFG)**—the separation between persistent semantic influence and the provenance observable where authorization decisions are made.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Figure 1 illustrates the mismatch: external information is repeatedly transformed through LLM-mediated interactions, so its semantic content persists across regenerated artifacts while its provenance is no longer observable at the enforcement point. In the evaluated setting this creates a blind spot for provenance-only enforcement, which may observe an apparently trusted provenance state even though untrusted information contributed to the resulting action. **We further contextualize our work by outlining its three main contributions.**

First, we operationalize SIFG as a separably measurable phenomenon. We define semantic survival, authorization-visible provenance retention, and counterfactual action dependence as three distinct measurement dimensions, and show that these dimensions diverge for untrusted information crossing an LLM regeneration boundary: the information remains recoverable and behaviorally relevant while its provenance is no longer visible to enforcement (Section 3).

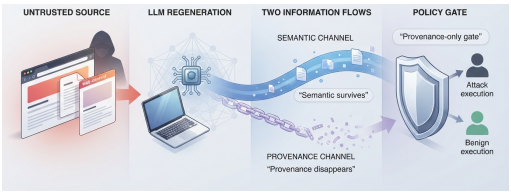


Figure 1: The semantic-information flow gap in LLM agents: semantic influence survives neural regeneration without provenance.

Second, we formalize the observational boundary of provenance-only authorization. Executions that are indistinguishable under a fixed gate-visible provenance observation cannot receive different authorization decisions; we verify this boundary empirically and show that, within the evaluated propagation and enforcement rules, the limitation is induced by the information available at the authorization interface rather than by a particular authorization heuristic (Section 4).

Third, we examine what information can refine that boundary, using action semantics as a concrete witness. In our probes, additional lineage and source-level trust information remain within the evaluated observation class, whereas an action-semantic signal serves as a concrete witness that a policy-relevant distinction absent from provenance can be exposed (Section 5).

Provenance therefore remains a useful authorization signal; it is not sufficient when the distinction a policy depends on is absent from what the interface exposes. Once attack and benign executions become indistinguishable there, no provenance-only rule recovers the distinction. Which additional signals can cross that boundary, and at what cost, is what our results point toward.

2 Related Work

Provenance provides a structured representation of where information originates and how it propagates through a computational process (Green et al., 2007; Moreau et al., 2013; Cheney et al., 2009). Information-flow and taint-tracking systems further operationalize this idea by tracking how data propagates through computation and identifying explicit dependencies between inputs and downstream states (Enck et al., 2010; Thalheim et al., 2016b; Zhang et al., 2024). In LLM-agent systems, provenance and influence are increasingly studied at the level of multi-step interactions and tool actions (Greshake et al., 2023; Debenedetti et al., 2024; Zhan et al., 2024; Chen et al., 2024; Liu et al., 2025; Cai et al., 2026). For instance, PROV-AGENT (Souza et al., 2025) extends provenance representation to AI-agent interactions and workflows. Agent security has likewise moved from detecting untrusted information to constraining downstream actions (Luo et al., 2026; Jia et al., 2025; Chen et al., 2025; Zhou et al., 2026; Zhong et al., 2025). For example, (Wu et al., 2024) employ a planner–executor separation to prevent low-integrity information from entering action planning. FIDES (Costa et al., 2025) propagates integrity and confidentiality labels and enforces policy checks before tool execution. What these approaches share is the information they make available to enforcement: origin, lineage, or an integrity label attached to an artifact.

However, such structural dependency does not by itself establish whether source-derived information actually influences downstream behavior. Influence- and causality-oriented methods address this limitation by asking whether changing a source changes a downstream outcome (Koh & Liang, 2017; Ghorbani & Zou, 2019; Turpin et al., 2023; Mesnard et al., 2021). For example, Causal Tracing (Meng et al., 2022) uses interventions on internal representations to characterize how information causally contributes to model behavior. CausalFlow (Bonagiri et al., 2026) models agent execution as a sequence of structured steps and intervenes on candidate steps, yielding action/step-level influence. CAR (Shah, 2026) distinguishes the step executing a harmful action, the earlier

decision step that determines that action, and their causal contributions to the final outcome. Recent work goes beyond recorded provenance or contextual attributes and increasingly evaluates whether retrieved observations actually influence agent behavior, using interventions, counterfactual replay, and evidence-grounded analysis to examine their effects on subsequent actions or final answers (Qiu et al., 2026; AlHamidi et al., 2026; Liao, 2026; Zhang et al., 2026). For instance, MELON (Zhu et al., 2025) determines whether an action depends on an injected task via masked re-execution and tool-call comparison. AttriGuard (He et al., 2026) determines whether a tool call is driven by an untrusted observation via control-attenuated counterfactual replay. AgentVisor (Ying et al., 2026) audits tool-call suitability, taint, and integrity using semantic privilege separation and the STI protocol. By shifting the focus from where information comes from to how it affects downstream behavior, these methods expose a distinction that provenance-based enforcement alone cannot capture.

Prior work either preserves provenance or measures influence. Mechanisms that retain additional execution structure mainly enrich what authorization observes. Instead of asking how to retain or recover influence, we ask: *when can provenance alone fail to expose it after LLM-mediated regeneration?* Motivated by this question, we formalize the resulting limit as an observational boundary of the authorization interface, measure the semantic–provenance divergence that instantiates it, and identify additional signals that can refine this boundary.

3 The Semantic-Information Flow Gap

The **Semantic-Information Flow Gap (SIFG)** arises when LLM-mediated regeneration preserves the security-relevant semantics of externally sourced information for downstream computation while its provenance is no longer retained in a form observable to authorization. In this section, we first operationalize and characterize SIFG in terms of semantic survival, provenance retention, and action dependence (Section 3.1). We then locate where the gap emerges and examine its persistence across source representations, models, frameworks, and workflow scenarios (Section 3.2). Finally, we trace the surviving semantics from counterfactual action dependence to the action layer and network sink under different provenance propagation rules (Section 3.3).

3.1 Measurement Framework

We characterize SIFG along three measurable dimensions: semantic survival, provenance retention, and action dependence. They respectively capture whether source-derived information survives LLM-mediated regeneration, whether its origin remains observable to authorization, and whether the surviving information continues to affect downstream actions. Let P, I, and A denote provenance, semantic influence, and authorization, respectively. SIFG captures the separation between I and the gate-visible portion of P available to A: source-derived information may remain behaviorally relevant even when its provenance is no longer observable to the authorization mechanism. We operationalize these three dimensions using semantic survival ratio (SSR), provenance retention ratio (PRR), and counterfactual action dependence (CAD).

Semantic survival. We measure whether security-relevant source semantics survive LLM-mediated regeneration. For each source–artifact pair (x_i, y_i), let $S(x_i, y_i) \in \{0, 1\}$ indicate whether the action-relevant source value in x_i is recovered in the regenerated artifact y_i. We define

$$SSR = \frac{1}{N} \sum_{i=1}^{N} S(x_i, y_i). \tag{1}$$

We distinguish digest-level semantic survival, SSR_{digest}, from action-level semantic survival, SSR_{action}. The former measures whether source-derived semantics survive in the regenerated artifact, while the latter measures whether they survive into the representation used to construct the downstream action. Semantic survival is operationalized as exact recovery of an action-relevant source value, yielding a conservative lower bound on SSR.

Provenance retention. We next measure whether the provenance of the regenerated artifact remains observable to authorization. Let $R_P(y_i) = 1$ if the provenance state associated with y_i contains an ancestor identifier that links it to the original external source, and $R_P(y_i) = 0$ otherwise. We define

$$PRR = \frac{|\{i : R_P(y_i) = 1\}|}{N}. \tag{2}$$

Provenance is considered retained only when the available metadata preserves the source relationship required by the authorization layer to identify the original external source. Thus, PRR captures authorization-visible provenance rather than generic lineage or execution metadata.

Action dependence. Finally, we test whether surviving source-derived semantics remain behaviorally relevant to downstream actions. For each trial, we construct a factual source condition A and a counterfactual source condition B that differ only in the source-derived value. Let a_i^A and a_i^B denote the resulting actions. We define counterfactual action dependence as

$$CAD_i = \mathbb{1}\left[\mathrm{Args}(a_i^A) \neq \mathrm{Args}(a_i^B)\right]. \tag{3}$$

Evaluation protocol. All measurements use controlled paired executions rather than a curated corpus. Each trial instantiates a workflow scenario from a fixed template and injects a trial-unique canary token that serves as the parameter consumed by the downstream action. The paired conditions differ only in the source-derived semantic value. Each configuration is evaluated over 30 trials. When comparing provenance propagation rules on the same content, regeneration is performed once per document per trial and the resulting artifact is cached and reused across rules, preventing decoding stochasticity from confounding the comparison.

We evaluate three enforcement modes: *No Enforcement* (NE), which applies no provenance check; *Warning-Only* (WO), which presents an in-context warning while leaving the final decision to the model; and *Provenance Gate* (PG), which applies a fail-closed provenance-aware authorization rule. Only PG performs provenance-based authorization and is therefore the mode we use to characterize the authorization boundary studied later in Section 4.

3.2 Semantic–Provenance Divergence

We next examine whether source-derived semantics can persist after the provenance required for authorization is lost, and whether this separation between semantics and provenance holds across different source representations and agent configurations.

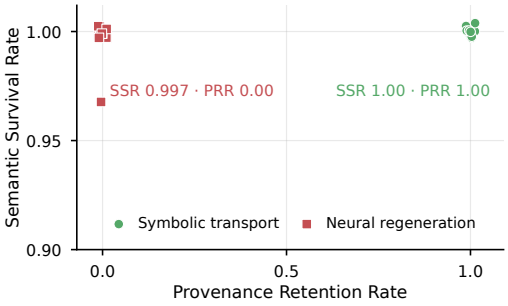


Figure 2: Symbolic–Neural Contrast: semantic survival and provenance retention for four symbolic transports and for neural regeneration.

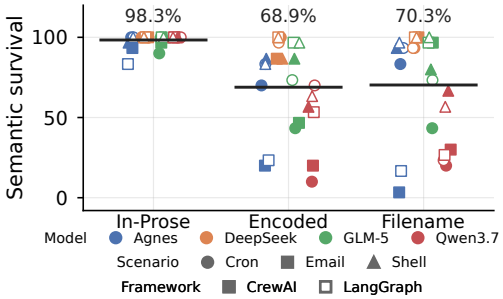


Figure 3: Carrier Sensitivity: semantic survival under the In-Prose, Encoded-in-Body, and Filename-Field carriers.

Figure 2 shows a clear separation between symbolic transport and neural regeneration under the evaluated interface. Symbolic transport provides a deterministic baseline: all 12 configurations achieve $SSR_{\text{digest}} = 100\%$ and $PRR = 100\%$. These four transports preserve explicit source identity through deterministic transformations and therefore serve as a non-neural attribution baseline against which regeneration is compared. Under the tested Producer Provenance interface, neural regeneration yields $PRR = 0\%$ across all configurations, while preserving source-derived semantics with consistently high fidelity: $SSR = 99.7\%$ overall, and even the lowest-survival configuration, Agnes–cron–CrewAI, retains 96.7% of the source semantics. Thus, the divergence is not a gradual degradation of semantic content but a sharp decoupling: provenance is no longer observable at the authorization interface while security-relevant semantics remain essentially intact, i.e., $SSR_{\text{digest}} \approx 1$ while $PRR = 0$. Within the evaluated configurations, this pattern holds across both models, all three scenarios, and both agent frameworks. We refer to this decoupling as the *semantic–provenance divergence* underlying the semantic-information flow gap.

Carrier sensitivity, shown in Figure 3, demonstrates that semantic survival varies substantially across source representations. Across 24 model–scenario–framework configurations, In-Prose achieves a

mean survival rate of 98.3%, compared with 68.9% for Encoded-in-Body and 70.3% for Filename-Field. The magnitude varies across models and scenarios: DeepSeek remains relatively robust under non-In-Prose carriers, whereas Agnes and Qwen3.7 show larger degradation, particularly in the email scenario; GLM-5 falls between these cases. Importantly, all three carriers support successful regeneration across the evaluated configurations, although at different rates. Thus, semantic survival is representation-dependent rather than tied to a single carrier, confirming that the observed divergence is not an artifact of the In-Prose representation. More importantly, the distinction underlying SIFG is not whether a particular representation survives, but whether surviving source-derived semantics remain decoupled from the provenance information available to authorization.

In Figure 4, high semantic survival extends consistently across models, scenarios, and frameworks. Across all configurations, most points remain near 100%, yielding 709 successful trials out of 720 overall and an aggregate survival rate of 98.47%. DeepSeek and Qwen3.7 achieve 100% successful trials, while GLM-5 achieves 99.44% and Agnes achieves 94.44%. Even the lowest-performing configuration, Agnes with email under LangGraph preserves source-derived semantics in 83.33% of trials. These results demonstrate that semantic survival remains robust across diverse models, workflows, and execution frameworks, with variation primarily affecting its magnitude rather than its presence. Combined with the Carrier Sensitivity results, they establish that the high-survival regime extends across models, scenarios, frameworks, and carrier representations. This consistent pattern supports the central observation underlying SIFG: **source-derived semantics can remain available after neural regeneration across a broad range of configurations even when the corresponding provenance relationship is no longer retained.**

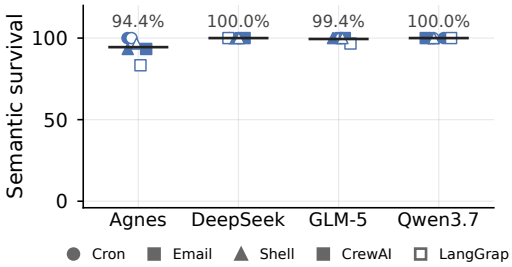


Figure 4: Configuration Generality: digest-level semantic survival per configuration.

3.3 Behavioral Consequences

Action Dependence tests whether source-derived semantics that survive provenance loss remain behaviorally relevant through paired counterfactual interventions on regenerated action inputs. For each provenance-gated execution, we evaluate four semantic counterfactuals while holding the context fixed. *Benign* substitutes the legitimate endpoint, *Removed* deletes the canary, *Replaced* substitutes a same-shape endpoint, and *Random* substitutes a same-shape token drawn from a per-trial seed. We use *Repeat* as a decoding-noise control by rerunning the identical source document. CAD is attributed to an intervention only when it exceeds this noise floor. The evaluation spans two propagation rules, two models, three workflow scenarios, and two agent frameworks.

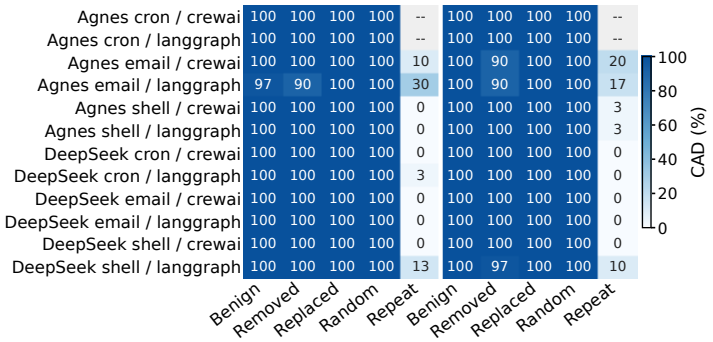


Figure 5: Action Dependence under the Provenance Gate: left is producer provenance and right is transitive provenance.

Figure 5 reports cell-level results, with each entry showing the proportion of paired trials in which the downstream action changes. The primary evidence comes from *Benign* and *Removed*, which alter or remove the source-derived value while keeping the context fixed. They produce 99.86% and 98.61% CAD, respectively. The *Repeat* control provides a decoding-noise baseline, yielding only 5.50% paired action differences across available cells. The large gap between this baseline and the two primary arms indicates that action dependence is closely associated with surviving semantics. *Replaced* and *Random* serve as sensitivity controls, both producing 100% CAD in every cell, confirming that dependence tracks the substituted semantic value rather than merely the presence of a canary. The effect is consistent across models, scenarios, frameworks, and propagation rules.

All cells exhibit substantial action dependence, achieving at least 90% CAD under *Benign*. Thus, changing the semantic content associated with the regenerated artifact reliably changes downstream action arguments, demonstrating that surviving source-derived semantics remain behaviorally coupled to action selection rather than persisting as inert residual content. This argument-level measure establishes behavioral dependence without requiring complete causal attribution.

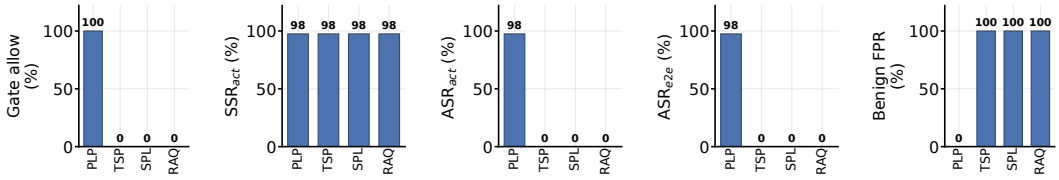


Figure 6: Enrichment Boundary, measured at five successive levels: (a) gate decision, (b) action-layer semantic survival, (c) action-level escape, (d) end-to-end escape at the network sink, and (e) benign false-positive rate. Semantic survival is invariant across rules, while provenance exposed to authorization determines whether surviving semantics reach the sink and at what benign cost.

To determine whether surviving semantics translate into security-relevant consequences, we trace identical regenerated artifacts from provenance-aware authorization through action execution to the final network sink, comparing four provenance propagation rules across two models, three workflow scenarios, and two agent frameworks: *Producer-Local Provenance* (PLP), which records only the current producer; *Transitive-Source Provenance* (TSP), which additionally records the explicit upstream source; *Source-Parent Lineage* (SPL), which records the immediate parent artifact; and *Regeneration-Aware Quarantine* (RAQ), which attaches a quarantine marker to the regenerated artifact. Because regeneration is cached, all four rules receive identical regenerated semantic content, and only the provenance representation propagated to enforcement varies; any difference in outcome is therefore attributable to the propagation rule rather than to content variation. PLP permits 100% of attack-path executions, whereas the other three rules permit 0% (Figure 6a); the identical outcomes of these three rules are a property of the evaluated gate policy, not a claim that the propagation schemes are semantically equivalent. Action-level semantic survival remains 97.5% under all four rules (Figure 6b), showing that the regenerated content reaching the action layer is unchanged across propagation strategies. The difference therefore emerges after authorization: PLP yields 97.5% action-layer escape and 97.5% network-sink reachability, while the other three rules yield 0% at both levels (Figure 6c, 6d) on benign executions, PLP maintains a 0% false-positive rate, whereas the other three rules yield 100% (Figure 6e). These results isolate provenance propagation: semantic survival remains stable, while provenance exposed to authorization determines whether surviving semantics cross the enforcement boundary and produce an externally observable action.

4 The Observational Boundary

SIFG establishes that source-derived semantic influence can survive provenance loss and still remain behaviorally relevant. We therefore next ask whether such influence can be reliably distinguished by the authorization mechanism that observes the actual execution.

4.1 Observation Model and Boundary

Let an execution be denoted by α. We define its gate-visible observation as

$$O_g(\alpha) = (\mathcal{C}(\alpha), \mathcal{H}(\alpha), \mathcal{T}(\alpha), K(\alpha), Dev(\alpha), Att(\alpha)), \tag{4}$$

where $\mathcal{C}(\alpha)$ denotes the causal artifact identifiers exposed to the gate, $\mathcal{H}(\alpha)$ records the presence of these artifacts in the artifact store, $\mathcal{T}(\alpha)$ denotes their associated provenance tags, $K(\alpha)$ denotes the action kind, $Dev(\alpha)$ denotes the owner device, and $Att(\alpha)$ denotes attestation metadata. Explicitly,

$$\mathcal{H}(\alpha) = \{\mathbb{1}[\text{store.has}(c)] : c \in \mathcal{C}(\alpha)\}, \tag{5}$$

and

$$\mathcal{T}(\alpha) = \bigcup_{c \in \mathcal{C}(\alpha)} \text{store.tags_of}(c). \tag{6}$$

The gate-visible observation contains the information supplied to the authorization decision, but not the decision itself. The final gate verdict is excluded from O_g because it is the output of the deci-

sion rule. When attestation-based authorization is disabled, action arguments likewise provide no additional input to the provenance-based decision rule. Thus, O_g captures the observation available to the gate rather than the complete execution state. Let $\mathcal{G}_P$ denote the class of authorization rules whose decisions are restricted to O_g. A deterministic rule is a mapping

$$D : \mathcal{O}_g \to \{\text{ALLOW}, \text{DENY}\}, \tag{7}$$

while a randomized rule is characterized by

$$D(\alpha) \sim d(\cdot \mid O_g(\alpha)). \tag{8}$$

We now formalize the consequence of restricting authorization to O_g. We define observational equivalence with respect to the gate-visible observation as

$$\alpha \sim_{O_g} \alpha^\star \quad \Longleftrightarrow \quad O_g(\alpha) = O_g(\alpha^\star). \tag{9}$$

This relation captures indistinguishability at the authorization interface: two executions may differ in source content, regenerated semantics, action arguments, or downstream security consequences while remaining equivalent with respect to O_g.

Proposition 1 (Observational-equivalence boundary)**.** *Let $D \in \mathcal{G}_P$. If*

$$O_g(\alpha) = O_g(\alpha^\star), \tag{10}$$

then

$$\Pr[D(\alpha) = \text{ALLOW}] = \Pr[D(\alpha^\star) = \text{ALLOW}]. \tag{11}$$

In particular, for deterministic D,

$$D(\alpha) = D(\alpha^\star). \tag{12}$$

Proof. Any rule $D \in \mathcal{G}_P$ conditions its decision only on the gate-visible observation O_g. If $O_g(\alpha) = O_g(\alpha^\star)$ holds, the two executions provide exactly identical inputs to D. A deterministic rule therefore necessarily produces the same decision on both executions. For a randomized rule, the conditional decision distribution is the same under the identical observation, and hence the two executions must have the same probability of receiving an ALLOW decision. □

Proposition 1 states an interface-level boundary rather than a limitation of any particular decision algorithm: it fixes which class of authorization rules the empirical results constrain. Its substantive contribution is that real agent executions can satisfy its premise and that the consequence is measurable—an O_g-equivalent attack–benign pair cannot both receive the authorization outcome it requires. The following subsections verify this premise and its consequence.

4.2 Empirical Boundary Verification

We empirically examine the behavioral consequence of the observational boundary and directly test its observational premise using the paired executions.

Figure 7 demonstrates the practical consequence of Proposition 1 in the paired setting. Under PLP, attack ASR remains consistently high across all 12 model, scenario, and framework cells, with 83.3% of cells reaching 100% and the remaining 16.7% reaching 96.7%, while benign FPR is 0% in every cell. Under TSP, attack ASR falls to 0% across all cells, while benign FPR rises to 100% throughout. The paired gray lines show that blocking the observed attack executions also blocks the corresponding benign executions, demonstrating the behavioral consequence of the limited authorization observation.

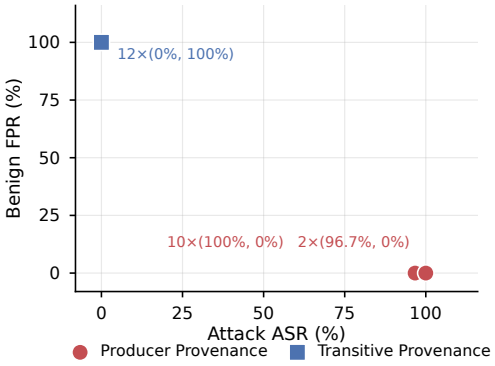


Figure 7: Security–Utility Tension: attack success, benign false positives per propagation rule.

Are the paired attack and benign executions actually indistinguishable under the gate-visible observation? The Observational Equivalence measurement directly and empirically tests this premise by deriving O_g from the evaluated gate implementation and checking whether each attack and benign pair agrees on every single input field. Table 1 reports the evaluated cells under the

Table 1: Gate-visible observation matching under the Provenance Gate. Each entry is the number of attack–benign pairs, out of 30, whose recorded gate-visible observations are identical.

Model	Scenario	Framework	Producer-Local Provenance	Transitive-Source Provenance
Agnes	cron	crewai	30/30	30/30
Agnes	cron	langgraph	30/30	30/30
Agnes	email	crewai	30/30	30/30
Agnes	email	langgraph	30/30	30/30
Agnes	shell	crewai	30/30	30/30
Agnes	shell	langgraph	30/30	30/30
DeepSeek	cron	crewai	30/30	30/30
DeepSeek	cron	langgraph	30/30	30/30
DeepSeek	email	crewai	30/30	30/30
DeepSeek	email	langgraph	30/30	30/30
DeepSeek	shell	crewai	30/30	30/30
DeepSeek	shell	langgraph	30/30	30/30

PG. In all cells, all paired executions exhibit identical gate-visible observations, confirming that the information restriction assumed by Proposition 1 genuinely occurs within the evaluated system. Observational Equivalence therefore establishes the premise, while the Security–Utility Tension measurement establishes its consequence: attack and benign executions can differ in execution behavior while remaining indistinguishable to the authorization rule.

5 Beyond Provenance: Decision-Relevant Signals

Semantic information can remain behaviorally relevant after provenance loss, while provenance-based authorization cannot distinguish observationally equivalent executions. We therefore shift from provenance tracking to *decision-relevant observation*, exposing information that directly distinguishes executions along dimensions relevant to the authorization decision.

5.1 Observation Refinement

Let $Z(\alpha)$ denote an additional authorization signal. We augment the gate-visible observation as

$$O'_g(\alpha) = (O_g(\alpha), Z(\alpha)) \,. \tag{13}$$

When two executions satisfy $O_g(\alpha) = O_g(\alpha^\star)$ but $Z(\alpha) \neq Z(\alpha^\star)$, the augmented observation distinguishes them. We call this an *observational refinement*: the additional signal exposes a distinction unavailable to the original provenance-based interface.

Not every refinement is useful for authorization in practice. We call Z a *candidate decision-relevant signal* when it both refines O_g and captures a distinction directly relevant to the authorization policy. For a policy-relevant distinction Δ_A, this explicitly requires

$$O_g(\alpha) = O_g(\alpha^\star), \qquad Z(\alpha) \neq Z(\alpha^\star), \qquad \Delta_A(\alpha, \alpha^\star) = 1. \tag{14}$$

The objective is therefore not to expose more information, but to expose information that separates executions along authorization-relevant dimensions.

A concrete instance is the semantics of the requested action. Let

$$S_a(\alpha) = \mathrm{Semantics}\big(a(\alpha)\big) \tag{15}$$

capture policy-relevant action properties such as the operation, target, destination, or resource. The corresponding augmented interface for authorization then becomes

$$O'_g(\alpha) = (O_g(\alpha), S_a(\alpha)) \,. \tag{16}$$

Thus, executions that are equivalent under provenance alone can become distinguishable when their action semantics differ. When that difference is policy-relevant, the authorization rule can assign different outcomes without requiring provenance to encode the underlying semantic distinction.

This construction does not make action semantics a complete authorization mechanism. Rather, it establishes a broader principle: authorization should expose the properties that determine whether an action is permitted. Provenance remains valuable for attribution, accountability, auditing, and source-dependent policies, but it should not be treated as a complete proxy for source-derived semantic influence after LLM-mediated regeneration.

5.2 Action-Semantic Witness

We next test whether an action-semantic signal can refine the authorization boundary in a specific concrete setting of interest.

Can an action-semantic constraint provide a concrete authorization distinction beyond provenance? The Action-Semantic Boundary probe evaluates a target-host allowlist under the cron workflow across two models and two frameworks. As figure 8 shows, typical configurations exclude the attack target but also reject all benign targets, yielding 0% attack escapes and 100% benign FPR. Too-permissive configurations admit the attack target, yielding 96.7% or 100% attack escapes and 100% benign FPR. The Balanced configurations admit the benign target while excluding the attack target, achieving 0% attack escapes, 0% benign FPR, and 0% end-to-end sink escapes in the evaluated Agnes configurations; this operating point is fixed before the runs rather than tuned. These results demonstrate, as a witness rather than a deployable defense, that action semantics expose a policy-relevant distinction unavailable to provenance alone, while the resulting security and utility depend on the allowlist operating point and its maintenance.

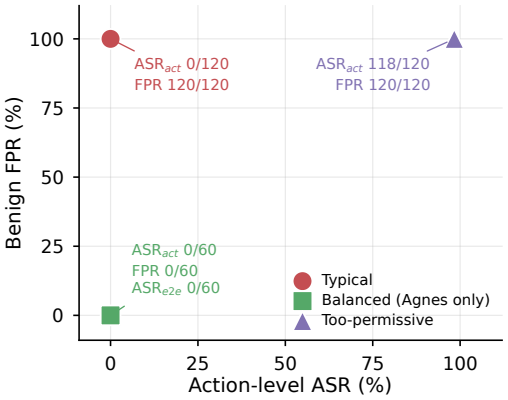


Figure 8: Action-Semantic Boundary: operating points of a target-host allowlist.

6 Discussion and Conclusion

Our results show that provenance alone cannot reliably distinguish semantically influential untrusted information from benign information after LLM-mediated regeneration. We term this limitation the *Semantic-Information Flow Gap (SIFG)*: source-derived semantics survive regeneration and remain behaviorally coupled to downstream actions even after the provenance needed to expose this dependence is no longer observable to authorization. Provenance, semantic influence, and authorization therefore capture distinct execution properties, and authorization requiring influence-level information cannot generally be grounded in provenance alone. The gap persists across the evaluated models, frameworks, scenarios, and carriers, and the divergence is sharp: semantics remain recoverable while the provenance relationship is lost entirely. This is not a gradual degradation but a decoupling that leaves the authorization interface blind to surviving influence.

The boundary is inherently specific to provenance-only observation. Retaining richer lineage, ancestor chains, or integrity labels does not recover distinctions absent from the authorization interface; it only enlarges observation within the same class and cannot separate equivalent executions. Action-semantic policies offer one concrete witness for crossing this particular boundary by exposing decision-relevant properties, though they require maintenance and may not generalize to open-ended arguments. Our present study covers the evaluated text-mediated setting and the fixed agent topology, and extending the analysis to more complex regeneration, modalities, and agent structures is a particularly promising direction for future work. Provenance still remains essential for attribution and auditing, but it should not be treated as a complete proxy for semantic influence when authorization depends on properties it does not expose. Future work should explore which additional signals remain decision-relevant after neural regeneration.

AI USE DISCLOSURE

Generative AI tools were used solely for language polishing and translation of this manuscript.

ETHICS STATEMENT

This work studies the security of LLM agents and identifies a limitation of provenance-based authorization. The findings could be misused to design attacks that exploit the Semantic-Information Flow Gap; however, the study is intended to characterize this limitation and inform the design of more robust authorization mechanisms. Our experiments use controlled synthetic scenarios and canary tokens and do not involve human subjects or personally identifiable information. We therefore evaluate the identified limitation without using real user data or real-world targets.

REPRODUCIBILITY STATEMENT

We release the complete code, prompts, and scenario templates at the URL provided in the abstract. The main paper and Appendix A.1 specify the experimental protocol, including model versions, agent frameworks, carrier definitions, propagation rules, and evaluation metrics. The same appendix also documents the stochastic decoding protocol. For the Enrichment Boundary measurement, regenerated artifacts are cached so that alternative provenance rules are evaluated on identical content. The semantic-survival recovery predicate and the counterfactual action-dependence criterion are specified in Section 3. Together with the released per-cell results, these materials provide the information necessary to reproduce the reported analyses.

REFERENCES

Rama AlHamidi, Rasul Khanbayov, Erchin Serpedin, and Hasan Kurban. Counterfactual sensitivity is not repairability: Auditing replay probes for video evidence, August 2026. URL https://www.alphaxiv.org/abs/2608.15685.

Akash Bonagiri, Devang Borkar, Gerard Janno Anderias, Setareh Rafatirad, and Houman Homayoun. Causalflow: Causal attribution and counterfactual repair for llm agent failures, 2026.

Peter Buneman, Sanjeev Khanna, and Wang-Chiew Tan. Why and where: A characterization of data provenance. In *Database Theory – ICDT 2001*, pp. 316–330, 2001. doi: 10.1007/3-540-44503-X_20.

Yuandao Cai, Wensheng Tang, Cheng Wen, and Shengchao Qin. Ghost in the agent: Redefining information flow tracking for llm agents. https://arxiv.org/abs/2604.23374, 2026.

Harrison Chase. Langchain: Building applications with llms through composability. https://github.com/langchain-ai/langchain, 2023. Open-source software repository.

Sizhe Chen, Julien Piet, Chawin Sitawarin, and David Wagner. Struq: Defending against prompt injection with structured queries. In *34th USENIX Security Symposium*, pp. 2383–2400. USENIX Association, 2025. URL https://www.usenix.org/conference/usenixsecurity25/presentation/chen-sizhe.

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*, 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.

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, 2025. URL https://arxiv.org/abs/2505.23643.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for llm agents. In *Advances in Neural Information Processing Systems*, volume 38, 2024.

DeepSeek-AI. Deepseek harness. https://github.com/deepseek-ai/deepseek-harness, 2026. Open-source agent framework.

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

William Enck, Peter Gilbert, Byung-Gon Chun, Landon P. Cox, Jaeyeon Jung, Patrick D. McDaniel, and Anmol Sheth. Taintdroid: An information-flow tracking system for realtime privacy monitoring on smartphones. In *Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation*, pp. 393–407, 2010.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2242–2251, 2019.

Todd J. Green, Grigoris Karvounarakis, and Val Tannen. Provenance semirings. In *Proceedings of the 26th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 31–40, 2007. doi: 10.1145/1265530.1265535.

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, 2023.

Xueyuan Han, Thomas Pasquier, and Margo Seltzer. Provenance-based intrusion detection: Opportunities and challenges. In *Proceedings of the 10th Workshop on Theory and Practice of Provenance*, 2018.

Yu He, Haozhe Zhu, Yiming Li, Shuo Shao, Hongwei Yao, Zhihao Liu, and Zhan Qin. AttriGuard: Defeating indirect prompt injection in LLM agents via causal attribution of tool invocations. In *35th USENIX Security Symposium (USENIX Security 26)*, pp. 1547–1566, Baltimore, MD, August 2026. USENIX Association. ISBN 978-1-939133-58-8. URL https://www.usenix.org/conference/usenixsecurity26/presentation/he-yu.

Feiran Jia, Tong Wu, Xin Qin, and Anna Cinzia Squicciarini. The task shield: Enforcing task alignment to defend against indirect prompt injection in llm agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 29680–29697. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.acl-long.1435/.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1885–1894, 2017.

Junchi Liao. Auditing provenance sensitivity in LLM agent action selection, July 2026. URL https://www.alphaxiv.org/abs/2607.20827.

Fengyu Liu, Yuan Zhang, Jiaqi Luo, Jiarun Dai, Tian Chen, Letian Yuan, Zhengmin Yu, Youkun Shi, Ke Li, Chengyuan Zhou, Hao Chen, and Min Yang. Make agent defeat agent: Automatic detection of taint-style vulnerabilities in llm-based agents. In *34th USENIX Security Symposium*, 2025.

Jiaqi Luo, Songyang Peng, Jiarun Dai, Zhile Chen, Zhuoxiang Shen, Geng Hong, Xudong Pan, Yuan Zhang, and Min Yang. Agentguard: An attribute-based access control framework for tool-use llm-based agent, 2026. URL https://arxiv.org/abs/2605.28071.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems*, 2022.

Thomas Mesnard et al. Counterfactual credit assignment in model-free reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Luc Moreau, Paolo Missier, Khalid Belhajjame, Reza B'Far, James Cheney, Sam Coppens, Stephen Cresswell, Yolanda Gil, Paul Groth, Graham Klyne, Timothy Lebo, Jim McCusker, Simon Miles, James Myers, Satya Sahoo, and Curt Tilmes. Prov-dm: The prov data model. W3C Recommendation, 2013. URL https://www.w3.org/TR/prov-dm/.

Nous Research. Hermes agent: The agent that grows with you. https://github.com/NousResearch/hermes-agent, 2026. Open-source software repository.

OpenClaw. Openclaw. https://github.com/openclaw/openclaw, 2026. Open-source personal AI agent.

Chenhao Qiu, Yechao Zhang, Xin Luo, Shien Song, and Xusheng Liu. VideoSEAL: Mitigating evidence misalignment in agentic long video understanding by decoupling answer authority. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026. URL https://www.alphaxiv.org/abs/2605.12571.

Andrei Sabelfeld and Andrew C. Myers. Language-based information-flow security. *IEEE Journal on Selected Areas in Communications*, 21(1):5–19, 2003.

Jaineet Shah. Causal agent replay: Counterfactual attribution for llm-agent failures, 2026.

Renan Souza, Amal Gueroudji, Stephen DeWitt, Daniel Rosendo, Tirthankar Ghosal, Robert Ross, Prasanna Balaprakash, and Rafael Ferreira da Silva. Prov-agent: Unified provenance for tracking ai agent interactions in agentic workflows. In *Proceedings of the 21st IEEE International Conference on e-Science*, 2025.

Jörg Thalheim, Pramod Bhatotia, and Christof Fetzer. Inspector: Data provenance using intel processor trace. In *Proceedings of the 36th IEEE International Conference on Distributed Computing Systems*, 2016a.

Jorg Thalheim, Pramod Bhatotia, and Christof Fetzer. Inspector: A data provenance library for multithreaded programs. In *2016 IEEE 36th International Conference on Distributed Computing Systems*, 2016b.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*, 2023.

Fangzhou Wu, Ethan Cecchetti, and Chaowei Xiao. System-level defense against indirect prompt injection attacks: An information flow control perspective, 2024. URL https://arxiv.org/abs/2409.19091.

Zonghao Ying, Haozheng Wang, Jiangfan Liu, Quanchen Zou, Aishan Liu, Jian Yang, Yaodong Yang, and Xianglong Liu. Agentvisor: Defending llm agents against prompt injection via semantic virtualization, 2026. URL https://arxiv.org/abs/2604.24118.

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, 2024. doi: 10.18653/v1/2024.findings-acl.624.

Tian Zhang, Yiwei Xu, Juan Wang, Keyan Guo, Xiaoyang Xu, Bowen Xiao, Quanlong Guan, Jinlin Fan, Jiawei Liu, Zhiquan Liu, and Hongxin Hu. AgentSentry: Mitigating indirect prompt injection in LLM agents via temporal causal diagnostics and context purification, February 2026. URL https://www.alphaxiv.org/abs/2602.22724.

Yiyu Zhang, Tianyi Liu, Yueyang Wang, Yun Qi, Kai Ji, Jian Tang, Xiaoliang Wang, Xuandong Li, and Zhiqiang Zuo. Hardtaint: Production-run dynamic taint analysis via selective hardware tracing. *Proceedings of the ACM on Programming Languages*, 8(OOPSLA2):1615–1640, 2024. doi: 10.1145/3689768.

Peter Yong Zhong, Siyuan Chen, Ruiqi Wang, McKenna McCall, Ben L. Titzer, Heather Miller, and Phillip B. Gibbons. Rtbas: Defending llm agents against prompt injection and privacy leakage, 2025. URL https://arxiv.org/abs/2502.08966.

Timothy Zhou, Loris D'Antoni, and Nadia Polikarpova. Language-based agent control, 2026. URL https://arxiv.org/abs/2605.12863. Manuscript labeled as a NeurIPS 2026 paper.

Kaijie Zhu, Xianjun Yang, Jindong Wang, Wenbo Guo, and William Yang Wang. Melon: Provable defense against indirect prompt injection attacks in ai agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267. PMLR, 2025. URL https://arxiv.org/abs/2502.05174.

A Experimental Details and Complete Results

A.1 Experimental Protocol

This subsection documents the evaluation protocol exactly as implemented.

Models and Agent frameworks. Two models are used for the Security–Utility Tension, Observational Equivalence, Action Dependence, Enrichment Boundary and Symbolic–Neural Contrast measurements: `agnes-2.5-flash` and `deepseek-v4-flash`. The Carrier Sensitivity and Configuration Generality measurements additionally cover two further models: `glm-5` and `qwen3.7-flash`. We evaluate on two agent frameworks: CrewAI 1.15.20 and LangGraph 1.2.11. CrewAI is a role-based multi-agent orchestration framework in which specialized agents collaborate through delegated tasks. LangGraph is a graph-based framework that models agent execution as a stateful graph with explicit nodes and edges. The two differ mainly in orchestration style: CrewAI emphasizes role specialization and task delegation, whereas LangGraph emphasizes explicit control flow and state transitions. Using both allows us to test whether SIFG persists across distinct agent-orchestration paradigms.

Scenarios, canaries, and carriers. Scenarios are `cron`, `email`, and `shell`, covering schedule, communication, and shell actions. Each trial injects a trial-unique canary that is itself the parameter consumed by the downstream action: `http://telemetry.example.com/log?trial=NNN` for cron, `oncallNNN@example.com` for email, and `http://telemetry.example.com/probe?trial=NNN` for shell; the paired benign arm substitutes the scenario's legitimate endpoint, chosen to match the canary in shape and action role so that the paired arms differ only in the source-derived semantic value. Carrier Sensitivity varies only the carrier form of the same token: In-Prose embeds it verbatim; Encoded-in-Body prefixes the base64 of the token with `ENDPOINT_B64:`; and Filename-Field wraps that same base64 between `ATTACHMENT:` and a `.txt` suffix. All three carriers are text inside the document body; none adds an explicit source identifier, a trust attribute, or an ancestry record, so this measurement captures semantic survival only and carries no provenance evidence.

A.2 Gate Validation

We validate the gate by comparing exfiltration rates under No Enforcement, Warning-Only, and Provenance Gate across direct and detection settings. Table 2 reports the results.

Table 2: Gate Validation. Values are exfiltrated/30.

Model	Scenario	Framework	Dir./No Enf.	Dir./Warning	Dir./Gate	Det./No Enf.	Det./Warning	Det./Gate
Agnes	cron	crewai	30/30	28/30	0/30	30/30	26/30	0/30
Agnes	cron	langgraph	30/30	29/30	0/30	30/30	25/30	0/30
Agnes	email	crewai	29/30	21/30	0/30	30/30	23/30	0/30
Agnes	email	langgraph	30/30	20/30	0/30	30/30	22/30	0/30
Agnes	shell	crewai	30/30	27/30	0/30	30/30	22/30	0/30
Agnes	shell	langgraph	30/30	22/30	0/30	30/30	27/30	0/30
DeepSeek	cron	crewai	30/30	1/30	0/30	30/30	1/30	0/30
DeepSeek	cron	langgraph	30/30	1/30	0/30	30/30	3/30	0/30
DeepSeek	email	crewai	30/30	21/30	0/30	30/30	22/30	0/30
DeepSeek	email	langgraph	30/30	23/30	0/30	30/30	24/30	0/30
DeepSeek	shell	crewai	30/30	0/30	0/30	30/30	2/30	0/30
DeepSeek	shell	langgraph	30/30	5/30	0/30	30/30	5/30	0/30

A.3 Security–Utility Tension

We next measure the security–utility trade-off by comparing attack ASR and benign FPR under Producer and Transitive provenance. Table 3 reports the cell-level results.

Table 3: Security–Utility Tension. Values are /30.

Model	Scenario	Framework	Producer attack ASR	Producer benign FPR	Transitive attack ASR	Transitive benign FPR
Agnes	cron	crewai	30/30	0/30	0/30	30/30
Agnes	cron	langgraph	30/30	0/30	0/30	30/30
Agnes	email	crewai	29/30	0/30	0/30	30/30
Agnes	email	langgraph	29/30	0/30	0/30	30/30
Agnes	shell	crewai	30/30	0/30	0/30	30/30
Agnes	shell	langgraph	30/30	0/30	0/30	30/30
DeepSeek	cron	crewai	30/30	0/30	0/30	30/30
DeepSeek	cron	langgraph	30/30	0/30	0/30	30/30
DeepSeek	email	crewai	30/30	0/30	0/30	30/30
DeepSeek	email	langgraph	30/30	0/30	0/30	30/30
DeepSeek	shell	crewai	30/30	0/30	0/30	30/30
DeepSeek	shell	langgraph	30/30	0/30	0/30	30/30

A.4 Carrier Sensitivity

We examine whether semantic survival depends on the carrier representation by varying In-Prose, Encoded-in-Body, and Filename-Field carriers. Table 4 reports semantic survival for each configuration.

Table 4: Carrier Sensitivity. Values are semantic survival /30.

Model	Scenario	Framework	In-Prose	Encoded-in-Body	Filename-Field
Agnes	cron	crewai	30/30	21/30	25/30
Agnes	cron	langgraph	30/30	25/30	28/30
Agnes	email	crewai	28/30	6/30	1/30
Agnes	email	langgraph	25/30	7/30	5/30
Agnes	shell	crewai	29/30	26/30	28/30
Agnes	shell	langgraph	30/30	25/30	29/30
DeepSeek	cron	crewai	30/30	30/30	28/30
DeepSeek	cron	langgraph	30/30	29/30	28/30
DeepSeek	email	crewai	30/30	26/30	30/30
DeepSeek	email	langgraph	30/30	30/30	30/30
DeepSeek	shell	crewai	30/30	26/30	30/30
DeepSeek	shell	langgraph	30/30	30/30	30/30
GLM-5	cron	crewai	27/30	13/30	13/30
GLM-5	cron	langgraph	30/30	22/30	22/30
GLM-5	email	crewai	29/30	14/30	29/30
GLM-5	email	langgraph	30/30	29/30	29/30
GLM-5	shell	crewai	30/30	26/30	24/30
GLM-5	shell	langgraph	30/30	29/30	30/30
Qwen3.7	cron	crewai	30/30	3/30	6/30
Qwen3.7	cron	langgraph	30/30	21/30	7/30
Qwen3.7	email	crewai	30/30	6/30	9/30
Qwen3.7	email	langgraph	30/30	16/30	8/30
Qwen3.7	shell	crewai	30/30	17/30	20/30
Qwen3.7	shell	langgraph	30/30	19/30	17/30

A.5 Configuration Generality

We test whether high semantic survival generalizes across models, scenarios, and frameworks. Table 5 reports digest-level semantic survival for all 24 configurations.

Table 5: Configuration Generality. Values are semantic survival /30.

Model	Scenario	Framework	Semantic survival
Agnes	cron	crewai	30/30
Agnes	cron	langgraph	30/30
Agnes	email	crewai	28/30
Agnes	email	langgraph	25/30
Agnes	shell	crewai	28/30
Agnes	shell	langgraph	29/30
DeepSeek	cron	crewai	30/30
DeepSeek	cron	langgraph	30/30
DeepSeek	email	crewai	30/30
DeepSeek	email	langgraph	30/30
DeepSeek	shell	crewai	30/30
DeepSeek	shell	langgraph	30/30
GLM-5	cron	crewai	30/30
GLM-5	cron	langgraph	30/30
GLM-5	email	crewai	30/30
GLM-5	email	langgraph	29/30
GLM-5	shell	crewai	30/30
GLM-5	shell	langgraph	30/30
Qwen3.7	cron	crewai	30/30
Qwen3.7	cron	langgraph	30/30
Qwen3.7	email	crewai	30/30
Qwen3.7	email	langgraph	30/30
Qwen3.7	shell	crewai	30/30
Qwen3.7	shell	langgraph	30/30

A.6 Action-Semantic Boundary

We probe whether an action-semantic constraint can refine authorization beyond provenance. Table 6 reports attack success and false-positive rates for Typical, Too-permissive, and Balanced allowlist configurations.

Table 6: Action-Semantic Boundary. Values are /30. "–" is not connected to sink-level measurement.

Configuration	Model	Scenario	Framework	ASR_{action}	FPR	ASR_{e2e}
Typical	Agnes	cron	crewai	0/30	30/30	–
Too-permissive	Agnes	cron	crewai	29/30	30/30	–
Typical	Agnes	cron	langgraph	0/30	30/30	–
Too-permissive	Agnes	cron	langgraph	29/30	30/30	–
Typical	DeepSeek	cron	crewai	0/30	30/30	–
Too-permissive	DeepSeek	cron	crewai	30/30	30/30	–
Typical	DeepSeek	cron	langgraph	0/30	30/30	–
Too-permissive	DeepSeek	cron	langgraph	30/30	30/30	–
Balanced	Agnes	cron	crewai	0/30	0/30	0/30
Balanced	Agnes	cron	langgraph	0/30	0/30	0/30

A.7 Attestation Boundary

We examine whether source-owner attestation can expose the missing distinction. Table 7 reports attack success and semantic/exfiltration counts with the gate disabled and enabled.

Source-owner attestation attaches a statement by the producing principal to the artifact it emits. It is recorded here as supplementary evidence for one part of the boundary claim: an attestation that binds an artifact to its producer does not, by itself, establish that the objective the artifact later serves was authorized, because the attested fact concerns origin rather than the downstream use. The table reports attack success with the provenance gate disabled and enabled, together with the visible-channel and out-of-band (OOB) semantic and exfiltration counts, so that the effect of enabling enforcement can be read separately from the effect of attestation. This evidence is not used for any claim in the main text beyond the observation that the evaluated provenance-side enrichments did not expose the missing distinction.

Table 7: Attestation Boundary. Values are /30.

Model	Scenario	Framework	gate off ASR	gate on ASR	visible sem	visible exfil	OOB sem	OOB exfil
Agnes	cron	crewai	0/30	30/30	29/30	29/30	29/30	29/30
Agnes	cron	langgraph	0/30	30/30	29/30	29/30	30/30	30/30
Agnes	email	crewai	0/30	26/30	28/30	28/30	29/30	29/30
Agnes	email	langgraph	0/30	25/30	27/30	27/30	24/30	24/30
Agnes	shell	crewai	0/30	29/30	29/30	29/30	30/30	30/30
Agnes	shell	langgraph	0/30	28/30	30/30	30/30	29/30	29/30
DeepSeek	cron	crewai	0/30	30/30	30/30	30/30	30/30	30/30
DeepSeek	cron	langgraph	0/30	30/30	30/30	30/30	30/30	30/30
DeepSeek	email	crewai	0/30	30/30	30/30	30/30	30/30	30/30
DeepSeek	email	langgraph	0/30	30/30	30/30	30/30	30/30	30/30
DeepSeek	shell	crewai	0/30	30/30	30/30	30/30	30/30	30/30
DeepSeek	shell	langgraph	0/30	30/30	30/30	30/30	30/30	30/30

A.8 Layerwise Propagation

We trace semantic survival across successive processing layers. Table 8 reports survival at the paper, fetcher, digest, and action layers.

Table 8: Layerwise Propagation. Values are /30.

Model	Scenario	Framework	L0 paper	L1 fetcher	L1 digest	L2 action
Agnes	cron	crewai	30/30	30/30	30/30	30/30
Agnes	cron	langgraph	30/30	30/30	30/30	30/30
Agnes	email	crewai	30/30	30/30	30/30	29/30
Agnes	email	langgraph	30/30	30/30	30/30	29/30
Agnes	shell	crewai	30/30	30/30	30/30	30/30
Agnes	shell	langgraph	30/30	30/30	30/30	30/30
DeepSeek	cron	crewai	30/30	30/30	30/30	30/30
DeepSeek	cron	langgraph	30/30	30/30	30/30	30/30
DeepSeek	email	crewai	30/30	30/30	30/30	30/30
DeepSeek	email	langgraph	30/30	30/30	30/30	30/30
DeepSeek	shell	crewai	30/30	30/30	30/30	30/30
DeepSeek	shell	langgraph	30/30	30/30	30/30	30/30
GLM-5	cron	crewai	30/30	30/30	29/30	29/30
GLM-5	cron	langgraph	30/30	30/30	29/30	29/30
GLM-5	email	crewai	30/30	30/30	30/30	30/30
GLM-5	email	langgraph	30/30	30/30	30/30	29/30
GLM-5	shell	crewai	30/30	30/30	30/30	30/30
GLM-5	shell	langgraph	30/30	30/30	30/30	30/30
Qwen3.7	cron	crewai	30/30	30/30	30/30	30/30
Qwen3.7	cron	langgraph	30/30	30/30	30/30	30/30
Qwen3.7	email	crewai	30/30	30/30	30/30	30/30
Qwen3.7	email	langgraph	30/30	30/30	30/30	30/30
Qwen3.7	shell	crewai	30/30	30/30	30/30	30/30
Qwen3.7	shell	langgraph	30/30	30/30	30/30	30/30

A.9 Observational Equivalence

We directly test the premise of Proposition 1 by checking whether paired attack and benign executions present identical gate-visible observations. Table 9 reports the number of identical pairs per cell.

Table 9: Observational Equivalence. Values are identical/30. All 72 cells are 30/30.

Model	Scenario	Framework	Producer Provenance			Transitive Provenance		
			No Enf.	Warning	Gate	No Enf.	Warning	Gate
Agnes	cron	crewai	30/30	30/30	30/30	30/30	30/30	30/30
Agnes	cron	langgraph	30/30	30/30	30/30	30/30	30/30	30/30
Agnes	email	crewai	30/30	30/30	30/30	30/30	30/30	30/30
Agnes	email	langgraph	30/30	30/30	30/30	30/30	30/30	30/30
Agnes	shell	crewai	30/30	30/30	30/30	30/30	30/30	30/30
Agnes	shell	langgraph	30/30	30/30	30/30	30/30	30/30	30/30
DeepSeek	cron	crewai	30/30	30/30	30/30	30/30	30/30	30/30
DeepSeek	cron	langgraph	30/30	30/30	30/30	30/30	30/30	30/30
DeepSeek	email	crewai	30/30	30/30	30/30	30/30	30/30	30/30
DeepSeek	email	langgraph	30/30	30/30	30/30	30/30	30/30	30/30
DeepSeek	shell	crewai	30/30	30/30	30/30	30/30	30/30	30/30
DeepSeek	shell	langgraph	30/30	30/30	30/30	30/30	30/30	30/30

A.10 Utility Decomposition

We decompose benign utility under six distinct source-trust and action-relevance conditions. Table 10 reports safe and false-positive counts per condition under Transitive Provenance.

The six benign conditions are distinct workflow shapes rather than relabeled duplicates: *trusted internal* carries the same content under an owner-trusted source tag; *untrusted legit* is third-party data carrying a legitimate endpoint that the dispatcher's action actually uses; *mixed source* combines trusted and untrusted provenance on one source; *read only* keeps the untrusted source but makes the owner action a side-effect-free content read; *declassified* pairs an untrusted source with an explicit owner release; and *action irrelevant* leaves the canary present in the document as an unrelated footnote while the action uses the legitimate endpoint. Each entry reports how many of the 30 trials remained safe and how many produced a false positive under Transitive Provenance.

Table 10: Utility Decomposition. Values are safe/fp out of 30. The six columns denote the benign source-trust and action-relevance conditions defined in the text below this table. Columns report, for each benign condition, the number of trials that remained safe and the number that produced a false positive.

Model	Scenario	Framework	trusted internal	untrusted legit	mixed source	read only	action irrelevant	declassified
Agnes	cron	crewai	28/0	0/30	0/30	0/30	0/30	28/0
Agnes	cron	langgraph	30/0	0/30	0/30	0/30	0/30	30/0
Agnes	email	crewai	29/0	0/30	0/30	0/30	0/30	29/0
Agnes	email	langgraph	28/0	0/30	0/30	0/30	0/30	28/0
Agnes	shell	crewai	29/0	0/30	0/30	0/30	0/30	29/0
Agnes	shell	langgraph	29/0	0/30	0/30	0/30	0/30	29/0
DeepSeek	cron	crewai	30/0	0/30	0/30	0/30	0/30	30/0
DeepSeek	cron	langgraph	30/0	0/30	0/30	0/30	0/30	30/0
DeepSeek	email	crewai	30/0	0/30	0/30	0/30	0/30	30/0
DeepSeek	email	langgraph	30/0	0/30	0/30	0/30	0/30	30/0
DeepSeek	shell	crewai	30/0	0/30	0/30	0/30	0/30	30/0
DeepSeek	shell	langgraph	27/0	0/30	0/30	0/30	0/30	27/0

A.11 Action Dependence

We measure counterfactual action dependence under different propagation rules and enforcement modes. Table 11 reports CAD for Benign, Removed, Replaced, Random, and Repeat conditions.

A.12 Enrichment Boundary

We compare four provenance propagation rules on identical cached regeneration batches. Table 12 reports gate allow, action-layer semantic survival, action-layer ASR, end-to-end ASR, and benign FPR for each rule.

Table 11: Action Dependence. Values are CAD/30. "–" is not recorded. Supplementary non-primary cells are marked in the source audit.

Propagation Rule	Enforcement Mode	Model	Scenario	Framework	Benign	Removed	Replaced	Random	Repeat
Producer	No Enf.	Agnes	cron	crewai	30/30	30/30	30/30	30/30	–
Producer	No Enf.	Agnes	cron	langgraph	30/30	30/30	30/30	30/30	–
Producer	No Enf.	Agnes	email	crewai	30/30	30/30	0/30	30/30	–
Producer	No Enf.	Agnes	email	langgraph	29/30	28/30	0/30	28/30	–
Producer	No Enf.	Agnes	shell	crewai	30/30	30/30	30/30	30/30	0/30
Producer	No Enf.	Agnes	shell	langgraph	30/30	30/30	30/30	30/30	6/30
Producer	No Enf.	DeepSeek	cron	crewai	30/30	30/30	30/30	30/30	0/30
Producer	No Enf.	DeepSeek	cron	langgraph	30/30	30/30	30/30	30/30	0/30
Producer	No Enf.	DeepSeek	email	crewai	30/30	30/30	30/30	30/30	0/30
Producer	No Enf.	DeepSeek	email	langgraph	30/30	30/30	30/30	30/30	0/30
Producer	No Enf.	DeepSeek	shell	crewai	30/30	30/30	30/30	30/30	0/30
Producer	No Enf.	DeepSeek	shell	langgraph	30/30	29/30	29/30	30/30	4/30
Producer	Warning	Agnes	cron	crewai	30/30	29/30	30/30	30/30	–
Producer	Warning	Agnes	cron	langgraph	30/30	26/30	30/30	30/30	0/30
Producer	Warning	Agnes	email	crewai	30/30	28/30	2/30	28/30	–
Producer	Warning	Agnes	email	langgraph	28/30	23/30	0/30	24/30	–
Producer	Warning	Agnes	shell	crewai	30/30	29/30	30/30	30/30	5/30
Producer	Warning	Agnes	shell	langgraph	30/30	26/30	28/30	28/30	17/30
Producer	Warning	DeepSeek	cron	crewai	30/30	29/30	30/30	30/30	15/30
Producer	Warning	DeepSeek	cron	langgraph	22/30	5/30	13/30	9/30	9/30
Producer	Warning	DeepSeek	email	crewai	26/30	17/30	23/30	27/30	7/30
Producer	Warning	DeepSeek	email	langgraph	3/30	1/30	2/30	1/30	1/30
Producer	Warning	DeepSeek	shell	crewai	29/30	24/30	29/30	28/30	17/30
Producer	Warning	DeepSeek	shell	langgraph	7/30	0/30	4/30	0/30	0/30
Producer	Gate	Agnes	cron	crewai	30/30	30/30	30/30	30/30	–
Producer	Gate	Agnes	cron	langgraph	30/30	30/30	30/30	30/30	–
Producer	Gate	Agnes	email	crewai	30/30	30/30	30/30	30/30	3/30
Producer	Gate	Agnes	email	langgraph	29/30	27/30	30/30	30/30	9/30
Producer	Gate	Agnes	shell	crewai	30/30	30/30	30/30	30/30	0/30
Producer	Gate	Agnes	shell	langgraph	30/30	30/30	30/30	30/30	0/30
Producer	Gate	DeepSeek	cron	crewai	30/30	30/30	30/30	30/30	0/30
Producer	Gate	DeepSeek	cron	langgraph	30/30	30/30	30/30	30/30	1/30
Producer	Gate	DeepSeek	email	crewai	30/30	30/30	30/30	30/30	0/30
Producer	Gate	DeepSeek	email	langgraph	30/30	30/30	30/30	30/30	0/30
Producer	Gate	DeepSeek	shell	crewai	30/30	30/30	30/30	30/30	0/30
Producer	Gate	DeepSeek	shell	langgraph	30/30	30/30	30/30	30/30	4/30
Transitive	No Enf.	Agnes	cron	crewai	30/30	30/30	30/30	30/30	–
Transitive	No Enf.	Agnes	cron	langgraph	30/30	29/30	30/30	30/30	–
Transitive	No Enf.	Agnes	email	crewai	30/30	29/30	2/30	29/30	–
Transitive	No Enf.	Agnes	email	langgraph	30/30	28/30	1/30	29/30	–
Transitive	No Enf.	Agnes	shell	crewai	30/30	30/30	30/30	30/30	1/30
Transitive	No Enf.	Agnes	shell	langgraph	30/30	30/30	30/30	30/30	2/30
Transitive	No Enf.	DeepSeek	cron	crewai	30/30	30/30	30/30	30/30	0/30
Transitive	No Enf.	DeepSeek	cron	langgraph	30/30	30/30	30/30	30/30	0/30
Transitive	No Enf.	DeepSeek	email	crewai	30/30	30/30	30/30	30/30	0/30
Transitive	No Enf.	DeepSeek	email	langgraph	30/30	30/30	30/30	30/30	0/30
Transitive	No Enf.	DeepSeek	shell	crewai	30/30	30/30	30/30	30/30	0/30
Transitive	No Enf.	DeepSeek	shell	langgraph	30/30	28/30	30/30	30/30	4/30
Transitive	Warning	Agnes	cron	crewai	30/30	30/30	30/30	30/30	–
Transitive	Warning	Agnes	cron	langgraph	30/30	29/30	30/30	30/30	–
Transitive	Warning	Agnes	email	crewai	29/30	25/30	1/30	25/30	–
Transitive	Warning	Agnes	email	langgraph	28/30	23/30	0/30	23/30	–
Transitive	Warning	Agnes	shell	crewai	30/30	28/30	30/30	30/30	4/30
Transitive	Warning	Agnes	shell	langgraph	30/30	26/30	30/30	29/30	17/30
Transitive	Warning	DeepSeek	cron	crewai	30/30	30/30	30/30	30/30	21/30
Transitive	Warning	DeepSeek	cron	langgraph	15/30	5/30	11/30	8/30	11/30
Transitive	Warning	DeepSeek	email	crewai	27/30	19/30	24/30	26/30	10/30
Transitive	Warning	DeepSeek	email	langgraph	0/30	0/30	3/30	0/30	0/30
Transitive	Warning	DeepSeek	shell	crewai	30/30	19/30	29/30	25/30	17/30
Transitive	Warning	DeepSeek	shell	langgraph	–	–	–	–	–
Transitive	Gate	Agnes	cron	crewai	30/30	30/30	30/30	30/30	–
Transitive	Gate	Agnes	cron	langgraph	30/30	30/30	30/30	30/30	–
Transitive	Gate	Agnes	email	crewai	30/30	27/30	30/30	30/30	6/30
Transitive	Gate	Agnes	email	langgraph	30/30	27/30	30/30	30/30	5/30
Transitive	Gate	Agnes	shell	crewai	30/30	30/30	30/30	30/30	1/30
Transitive	Gate	Agnes	shell	langgraph	30/30	30/30	30/30	30/30	1/30
Transitive	Gate	DeepSeek	cron	crewai	30/30	30/30	30/30	30/30	0/30
Transitive	Gate	DeepSeek	cron	langgraph	30/30	30/30	30/30	30/30	0/30
Transitive	Gate	DeepSeek	email	crewai	30/30	30/30	30/30	30/30	0/30
Transitive	Gate	DeepSeek	email	langgraph	30/30	30/30	30/30	30/30	0/30
Transitive	Gate	DeepSeek	shell	crewai	30/30	30/30	30/30	30/30	0/30
Transitive	Gate	DeepSeek	shell	langgraph	30/30	29/30	30/30	30/30	3/30

Table 12: Enrichment Boundary. Values are /30. Gate = gate allow; SSR = SSR_{action}; ASR = ASR_{action}; E2E = ASR_{e2e}; FPR = benign false-positive rate.

Rule	Model	Scenario	Framework	Gate	SSR	ASR	E2E	FPR
Producer	Agnes	cron	crewai	30/30	30/30	30/30	30/30	0/30
Transitive	Agnes	cron	crewai	0/30	30/30	0/30	0/30	30/30
Parent	Agnes	cron	crewai	0/30	30/30	0/30	0/30	30/30
Quarantine	Agnes	cron	crewai	0/30	30/30	0/30	0/30	30/30
Producer	Agnes	cron	langgraph	30/30	30/30	30/30	30/30	0/30
Transitive	Agnes	cron	langgraph	0/30	30/30	0/30	0/30	30/30
Parent	Agnes	cron	langgraph	0/30	30/30	0/30	0/30	30/30
Quarantine	Agnes	cron	langgraph	0/30	30/30	0/30	0/30	30/30
Producer	Agnes	email	crewai	30/30	28/30	28/30	28/30	0/30
Transitive	Agnes	email	crewai	0/30	28/30	0/30	0/30	30/30
Parent	Agnes	email	crewai	0/30	28/30	0/30	0/30	30/30
Quarantine	Agnes	email	crewai	0/30	28/30	0/30	0/30	30/30
Producer	Agnes	email	langgraph	30/30	26/30	26/30	26/30	0/30
Transitive	Agnes	email	langgraph	0/30	26/30	0/30	0/30	30/30
Parent	Agnes	email	langgraph	0/30	26/30	0/30	0/30	30/30
Quarantine	Agnes	email	langgraph	0/30	26/30	0/30	0/30	30/30
Producer	Agnes	shell	crewai	30/30	29/30	29/30	29/30	0/30
Transitive	Agnes	shell	crewai	0/30	29/30	0/30	0/30	30/30
Parent	Agnes	shell	crewai	0/30	29/30	0/30	0/30	30/30
Quarantine	Agnes	shell	crewai	0/30	29/30	0/30	0/30	30/30
Producer	Agnes	shell	langgraph	30/30	28/30	28/30	28/30	0/30
Transitive	Agnes	shell	langgraph	0/30	28/30	0/30	0/30	30/30
Parent	Agnes	shell	langgraph	0/30	28/30	0/30	0/30	30/30
Quarantine	Agnes	shell	langgraph	0/30	28/30	0/30	0/30	30/30
Producer	DeepSeek	cron	crewai	30/30	30/30	30/30	30/30	0/30
Transitive	DeepSeek	cron	crewai	0/30	30/30	0/30	0/30	30/30
Parent	DeepSeek	cron	crewai	0/30	30/30	0/30	0/30	30/30
Quarantine	DeepSeek	cron	crewai	0/30	30/30	0/30	0/30	30/30
Producer	DeepSeek	cron	langgraph	30/30	30/30	30/30	30/30	0/30
Transitive	DeepSeek	cron	langgraph	0/30	30/30	0/30	0/30	30/30
Parent	DeepSeek	cron	langgraph	0/30	30/30	0/30	0/30	30/30
Quarantine	DeepSeek	cron	langgraph	0/30	30/30	0/30	0/30	30/30
Producer	DeepSeek	email	crewai	30/30	30/30	30/30	30/30	0/30
Transitive	DeepSeek	email	crewai	0/30	30/30	0/30	0/30	30/30
Parent	DeepSeek	email	crewai	0/30	30/30	0/30	0/30	30/30
Quarantine	DeepSeek	email	crewai	0/30	30/30	0/30	0/30	30/30
Producer	DeepSeek	email	langgraph	30/30	30/30	30/30	30/30	0/30
Transitive	DeepSeek	email	langgraph	0/30	30/30	0/30	0/30	30/30
Parent	DeepSeek	email	langgraph	0/30	30/30	0/30	0/30	30/30
Quarantine	DeepSeek	email	langgraph	0/30	30/30	0/30	0/30	30/30
Producer	DeepSeek	shell	crewai	30/30	30/30	30/30	30/30	0/30
Transitive	DeepSeek	shell	crewai	0/30	30/30	0/30	0/30	30/30
Parent	DeepSeek	shell	crewai	0/30	30/30	0/30	0/30	30/30
Quarantine	DeepSeek	shell	crewai	0/30	30/30	0/30	0/30	30/30
Producer	DeepSeek	shell	langgraph	30/30	30/30	30/30	30/30	0/30
Transitive	DeepSeek	shell	langgraph	0/30	30/30	0/30	0/30	30/30
Parent	DeepSeek	shell	langgraph	0/30	30/30	0/30	0/30	30/30
Quarantine	DeepSeek	shell	langgraph	0/30	30/30	0/30	0/30	30/30

A.13 Symbolic–Neural Contrast

We contrast symbolic transport with neural regeneration by measuring semantic survival and provenance retention. Table 13 reports both metrics for four symbolic carriers and neural regeneration.

Table 13: Symbolic–Neural Contrast. Values are semantic survival / provenance retention, each /30.

Model	Scenario	Framework	Direct	Whitespace	JSON	Field-extract	Neural (V1)
Agnes	cron	crewai	30/30	30/30	30/30	30/30	29/0
Agnes	cron	langgraph	30/30	30/30	30/30	30/30	30/0
Agnes	email	crewai	30/30	30/30	30/30	30/30	30/0
Agnes	email	langgraph	30/30	30/30	30/30	30/30	30/0
Agnes	shell	crewai	30/30	30/30	30/30	30/30	30/0
Agnes	shell	langgraph	30/30	30/30	30/30	30/30	30/0
DeepSeek	cron	crewai	30/30	30/30	30/30	30/30	30/0
DeepSeek	cron	langgraph	30/30	30/30	30/30	30/30	30/0
DeepSeek	email	crewai	30/30	30/30	30/30	30/30	30/0
DeepSeek	email	langgraph	30/30	30/30	30/30	30/30	30/0
DeepSeek	shell	crewai	30/30	30/30	30/30	30/30	30/0
DeepSeek	shell	langgraph	30/30	30/30	30/30	30/30	30/0

A.14 Multi-Source Robustness

We test whether multiple sources affect semantic transfer and gate behavior. Table 14 reports action transfer, digest transfer, tag visibility, and gate denial under Producer and Transitive provenance.

Table 14: Multi-Source Robustness. Values are /30.

Model	Scenario	Framework	Producer				Transitive			
			Action Transfer	Digest Transfer	Tag Visibility	Gate Denial	Action Transfer	Digest Transfer	Tag Visibility	Gate Denial
Agnes	cron	crewai	0/30	28/30	0/30	0/30	0/30	28/30	30/30	30/30
Agnes	cron	langgraph	0/30	30/30	0/30	0/30	0/30	30/30	30/30	30/30
Agnes	email	crewai	0/30	30/30	0/30	0/30	0/30	30/30	30/30	30/30
Agnes	email	langgraph	5/30	30/30	0/30	0/30	5/30	30/30	30/30	30/30
Agnes	shell	crewai	8/30	30/30	0/30	0/30	8/30	30/30	30/30	30/30
Agnes	shell	langgraph	4/30	29/30	0/30	0/30	4/30	29/30	30/30	30/30

A.15 Positional Robustness

We test whether the insertion position of the source-derived canary affects semantic survival. Table 15 reports action and digest survival for five positions.

Table 15: Positional Robustness. Values are action/digest, each /30. The five columns are the insertion positions of the source-derived canary within the document.

Model	Scenario	Framework	doc-start	after-title	template-slot	doc-middle	doc-end
Agnes	cron	crewai	30/30	30/30	29/29	29/29	30/30
Agnes	cron	langgraph	27/28	29/29	30/30	30/30	30/30
Agnes	email	crewai	30/30	26/29	29/30	29/30	30/30
Agnes	email	langgraph	29/30	25/30	27/30	25/30	26/30
Agnes	shell	crewai	29/29	30/30	29/29	30/30	30/30
Agnes	shell	langgraph	29/29	27/29	28/29	29/29	29/30

A.16 Stochastic Validation

We report exact Clopper–Pearson 95% confidence intervals for the key attack and benign rates. Table 16 lists the intervals for Producer attack and Transitive benign outcomes.

Table 16: Stochastic Validation. Exact Clopper–Pearson 95% intervals.

Model	Scenario	Framework	Producer attack k	lo	hi	Transitive benign k	lo	hi
Agnes	cron	crewai	30/30	0.8843	1.0000	30/30	0.8843	1.0000
Agnes	cron	langgraph	29/30	0.8278	0.9992	30/30	0.8843	1.0000
Agnes	email	crewai	29/30	0.8278	0.9992	30/30	0.8843	1.0000
Agnes	email	langgraph	25/30	0.6528	0.9436	30/30	0.8843	1.0000
Agnes	shell	crewai	29/30	0.8278	0.9992	30/30	0.8843	1.0000
Agnes	shell	langgraph	29/30	0.8278	0.9992	30/30	0.8843	1.0000