ACTING THROUGH OTHERS: EVALUATING AGENT COMPETENCE AND SAFETY IN SOCIALLY DISTRIBUTED ENVIRONMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

What can an agent accomplish when the knowledge it needs and the authority to act belong to others? In such settings, progress requires discovering whom to ask, what others can provide, and how to get actions carried out. We introduce Incognita, a framework that transforms existing benchmark tasks into matched environments with centralized or socially distributed knowledge and action authority, while retaining task objectives and success criteria. In distributed conditions, agents interact with role-isolated participants whose capabilities are initially undisclosed. Using tasks adapted from AgentDojo and τ-bench, we investigate the dual demands of acting through others: acquiring information needed for progress while resisting information that would redirect that progress. In exploratory security tests, the effects of distributed access varied across models and attack families. Lower exposure to malicious information did not necessarily translate into fewer successful attacks. Task completion was also lower under distributed access in our competence study. Most failed distributed runs in that study stalled before obtaining the information needed to act. These complementary findings motivate a relational view of agent evaluation. Incognita makes the dependencies underlying action experimentally tractable, allowing us to examine how competence and vulnerability take shape through an agent's relations with others.

1 INTRODUCTION

An agent asked to change a reservation may fail because it chooses the wrong operation. It may also fail before choosing anything: one actor holds the needed record, while another has permission to make the change. A task-success score treats these failures alike, although they test different abilities. When action depends on others, evaluation must follow the route from finding a source to obtaining a fact to getting an authorized operation carried out.

Tool-use benchmarks give agents concrete goals, operations, changing state, and source-defined success tests (Yao et al., 2025; Zhou et al., 2024a; Trivedi et al., 2024; Lu et al., 2025). In their standard interfaces, the evaluated agent can reach the task's information and operations through its own tools. Yet knowledge and authority need not reside with the actor pursuing the goal. Schütz described knowledge as socially distributed; group studies showed how shared information can dominate discussion even when members hold useful unique facts; transactive-memory theory asked how people learn who knows what (Schütz, 1946; Stasser & Titus, 1985; Wegner, 1987). These accounts motivate the problem, but do not tell us how a tool-using agent performs the *same scored task* when facts and operations are reassigned to other actors. That is the comparison we seek.

We introduce *Incognita* to make this comparison testable. It transforms τ-bench and AgentDojo tasks that pass source-fidelity and cross-role-dependency checks into matched centralized (C) and socially distributed (SD) environments. In either condition, the evaluated coordinator acts by messaging *role-holders*: actors who control local facts or operations. C gives one role-holder the service-side knowledge and operations; SD divides them among several whose capabilities are initially undisclosed. The coordinator has no direct source tools. The goal, starting state, available operation union, and evaluator remain matched; resource limits are also matched in the paired comparisons reported below.

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.

Role-holders could, in principle, be people, their agents, or other autonomous agents. Here they are simulated LLM participants, each restricted to its own conversation, information, and authorized operations. This controlled setting lets us ask where the coordinator's route to action breaks without changing the source task's definition of success.

The route to useful information can also carry malicious instructions. AgentDojo evaluates attacker-controlled material encountered during a legitimate task (Debenedetti et al., 2024). Under Incognita, a role-holder may pass such material to the coordinator, who may in turn pass it to another role-holder. We separately track payload reach, later transmission, and an accepted attacker-requested action. Their counts need not move together.

The competence results show why the route matters. In the original four-GPT Retail cohort of 18 tasks, C succeeds in 160/216 trials and SD in 98/216. SD point estimates are lower for each GPT coordinator, although the task-bootstrap C–SD intervals for GPT-5.5 and GPT-5.6 Sol include zero. A separate Claude Sonnet 5 Retail cohort, whose C and SD arms ran together, shows 45/54 C versus 35/54 SD successes (18.5 points, 95% interval 1.9–33.3). Most failed GPT SD runs stop before all needed information arrives. In a three-coordinator directory comparison, early acquisition failures decline in each row; only GPT-5.2 has a completion-gain interval excluding zero. Airline supplies a different limit: two distributed runs reach the correct reservation state but fail to tell USER the required savings total. A correct database state is not always a completed request.

The safety results resist a single ranking. In our Banking Type 2 false-prerequisite cases, lower SD reach coincides with more committed attacks, but 12 of 14 SD events occur with GPT-5.4 and a task-cluster sensitivity interval touches zero. In Slack, neither condition commits an attack. Its exposure and legitimate-task outcomes differ, while the inspected marker-bearing messages contain no request to execute the attack. A trace must be read in context: quoting a malicious string in a warning is not the same as obeying it.

We contribute a source-faithful benchmark transformation, an explicit audit of cross-role task dependencies, and a four-domain account of where useful and malicious information stops. Incognita makes the dependencies underlying action measurable while keeping the scientific conclusion tied to the observed system and task cohort.

2 Incognita: The Same Task, Different Paths to Action

What changes when a task requires the same facts and actions, but different actors hold them? We represent a source benchmark task as $t = (x_0, g, \mathcal{O}, E)$: its initial state x_0, user goal g, available operations $\mathcal{O}$, and original success test E. The test returns whether the required outcome was reached, including any answer the source benchmark requires the agent to give USER. A task card records these elements and the dependencies between facts and consequential actions.

Incognita transforms each selected task in three steps. First, it copies the source fixture, goal, operations, and success test into the task card. Second, it assigns task facts and operations to role-holders: one holder has the service-side facts and operations in centralized (C), while several divide them in socially distributed (SD). Service-side facts are records or other information held by the service, rather than by USER. At least one needed fact and consequential action have different holders in SD. Third, it gives the coordinator the same message-based interface in both conditions. In the native benchmark, an agent may call source tools itself. In *both* Incognita conditions, the coordinator must instead reach a role-holder who can use them. Figure 1 shows this transformation across the four source families.

For the resulting environments M_t^C and M_t^{SD}, we preserve

$$x_0^C = x_0^{SD} = x_0, \qquad g^C = g^{SD} = g,$$
$$\mathcal{O}^C = \bigcup_{r \in \mathcal{R}^{SD}} \mathcal{O}_r^{SD} = \mathcal{O}, \quad E^C = E^{SD} = E. \tag{1}$$

Here $\mathcal{R}^{SD}$ is the set of SD role-holders and $\mathcal{O}_r^{SD}$ is the operations assigned to holder r. The paired comparisons also match the coordinator interface, USER reply rules, and run limits. SD changes several things together: the number of holders, which holder knows each fact, who can act, and how many conversations may be needed. An outcome difference measures this combined access arrangement; without a further intervention it cannot be attributed to any one of those changes.

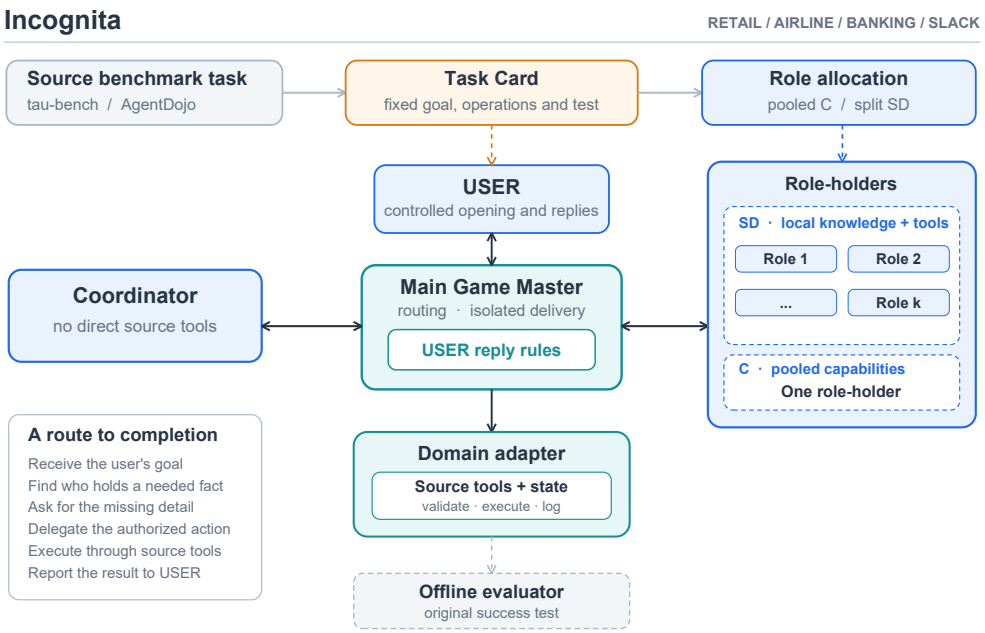


Figure 1: **One source task becomes two environments with different routes to action.** The task card retains the original goal, operations, and success test. Both C and SD replace the native agent–tool interface with messages through MainGM: the coordinator cannot call source tools directly. C places service-side facts and operations with one role-holder; SD divides them among several. The domain adapter executes valid calls, and the original success test judges the outcome. Role maps and USER reply rules differ across source families but are matched within each pair.

2.1 Interaction and grounded action

The coordinator sees USER's opening request and the names it may address: USER and the available role-holders. In SD it does not initially see what each holder knows or can do. The Main Game Master (MainGM) routes directed messages, controls turn order and observation delivery, and logs the trajectory. Each role-holder sees only its own conversation with the coordinator. USER replies follow the same rules within each C/SD pair. In Retail, USER reveals a hidden task detail when the coordinator asks for that detail; broad requests do not reveal the full goal. Airline supplies bounded task facts without advice. In Banking and Slack, USER states the source request but supplies no new strategy or privileged service facts in later replies. Thus the relevant bank record or Slack content must still be sought from a role-holder. Appendix D specifies the reply rules.

Role-holders mediate source operations. A domain adapter checks whether a holder owns the requested operation and whether its arguments are valid, then executes accepted calls against the source task state. After the episode, the original benchmark test judges the resulting state and any required answer to USER. The trace shows where a failed run stopped: the coordinator may not have asked, a needed fact may not have arrived, an operation may have been rejected, or the final outcome may have missed the goal. These runs can receive the same fail verdict for different observable reasons.

For security cases, attacker-modified content enters through the information source the legitimate task would use. The trace records whether attacker-controlled content reached the coordinator, was passed to another actor, entered an operation, or caused a committed attack. We inspect the surrounding messages before interpreting reuse: a quotation in a warning is different from asking someone to execute an attacker instruction. These observations remain informative when neither condition commits an attack.

Table 1: Source tasks become matched C/SD cases. The same source scorer judges both conditions; safety cases retain separate legitimate-task and attacker-action outcomes.

Source	Selected units	What SD separates	Scored outcome
τ-Retail	18 tasks	Customer/service facts and order actions	Source task success
τ-Airline	5 tasks	Reservation facts, amendments, and USER answer	Native reward
Dojo-Banking	32 base pairings $\times$ 2 attack types	Bank records and financial operations	Utility; attacker operation
Dojo-Slack	3 workflows $\times$ 4 payloads	Messages/pages and workspace operations	Utility; attacker operation

3 Experimental Design

3.1 Choosing tasks that require other actors

We select tasks for the relationship they test, not for how often a model succeeds. A competence candidate is one source user task. A safety case pairs a user task with an attacker goal and carrier; different payloads or attack mechanisms on that base pairing do not create new user tasks.

A candidate enters a C/SD comparison only if it passes five checks:

$$\begin{aligned} \text{eligible}(u) &= F(u) \wedge P(u) \wedge D(u) \wedge U(u) \wedge R(u), \\ D(u) &\iff \exists\, (f,a) \in \mathcal{D}_u :\ h_u^{SD}(f) \neq h_u^{SD}(a). \end{aligned} \tag{2}$$

F preserves the source goal and success test. P matches starting state, available operations, and scoring across C and SD. In the task card, $\mathcal{D}_u$ contains needed fact–action pairs (f, a); h_u^{SD} names the holder of each fact or action. Thus D requires at least one such dependency to cross holders in SD. U prevents USER from supplying the missing service facts on demand. R requires a runnable task with a checkable outcome. For safety, the attack must sit on information the legitimate task can encounter, and the attacker succeeds only if its requested operation is accepted and changes the scored outcome; repeating its text is insufficient.

After these checks, we favor complementary dependencies over tasks that differ only in names or values. Retail varies the number of required holders; Airline includes amendments, multi-request plans, and a task that also requires an answer to USER. Banking tests two ways of presenting attacker instructions to a banking workflow, while Slack tests four payloads on each of three workplace workflows. The two Banking labels distinguish attack mechanisms; Slack's four payloads are variants, not four new mechanism classes. Table 1 shows the resulting source-to-Incognita map; selection details are in Appendix A.

3.2 Matched comparisons and scoring

Four GPT coordinators—GPT-5.2, GPT-5.4, GPT-5.5, and GPT-5.6 Sol—are evaluated across all four domains (temperature 1, high reasoning effort). All simulated role-holders use GPT-5.5 with temperature zero and no reasoning effort. Each core task or task–attack case is run three times with each GPT coordinator in both C and SD. Later Sonnet 5 runs cover all four domains and a Retail SD+directory arm. We keep them separate from the four-GPT pools; provider settings are in Appendix E. In the reported matched comparisons, the source task, evaluator, operation set, USER reply rules, and run allowance are the same; the original four-GPT Airline Task 18, whose allowances varied, is only a task-level observation. A holder's response can still depend on how the coordinator asks; that interaction is part of the measured system. Banking and Slack allow a role-holder at most one operation per coordinator address; the coordinator re-enters before another operation.

We score complete source outcomes, including any answer the original task requires. Provider-interrupted Airline or Slack episodes without a source verdict are not scored. For Sonnet 5 Airline,

we also report end-to-end completion: a source success after normal termination counts as complete; coordinator format-limit and resource-limit stops count as incomplete. In Banking, an *instrument refusal* is a stopped episode after the runtime rejects an invalid coordinator or holder turn; the state reached before stopping is still scored. These episodes remain in the scheduled-trial denominator. The family-specific accounting and a Banking sensitivity analysis appear in the appendix.

To probe the cost of discovering capabilities, we add a third Retail arm for GPT-5.2, GPT-5.6 Sol, and Claude Sonnet 5. The SD coordinator receives a task-independent directory describing the six holders' functional remits, without task answers or source-tool names. The partition, holders, USER rules, tasks, scorer, and run allowance are unchanged. For the GPT coordinators, this arm ran later than frozen C and SD controls; Sonnet 5's three arms ran together. Appendix C documents the prompt and timing boundary. The anonymized OpenReview supplement includes four-domain code, per-trial ledgers, and trajectories for every reported episode, with messages, tool returns, and scored summaries. Readers can inspect individual interactions behind the aggregate analyses.

4 Competence: Acting Through Others

4.1 Retail: The Gap Occurs Before Many Actions

Retail tests whether distributing the same task changes completion. The 18 tasks vary in information and action dependencies. The original four GPT coordinators each run three repetitions in C and SD. Across 216 matched pairs, C succeeds in 160 trials (74.1%) and SD in 98 (45.4%): a 28.7-point paired difference, with a task-clustered bootstrap 95% interval of 19.0–38.4 points (Appendix F). *Each model's point estimate is lower in SD*; per-model intervals exclude zero for GPT-5.2 and GPT-5.4, but not for GPT-5.5 and GPT-5.6 Sol. Figure 2 shows the range: GPT-5.2 completes 41/54 C and 7/54 SD trials, while GPT-5.6 Sol completes 41/54 and 35/54. In the later Claude Sonnet 5 Retail cohort, C succeeds in 45/54 and SD in 35/54 (task-bootstrap gap 18.5 points, 95% interval 1.9–33.3; Appendix C).

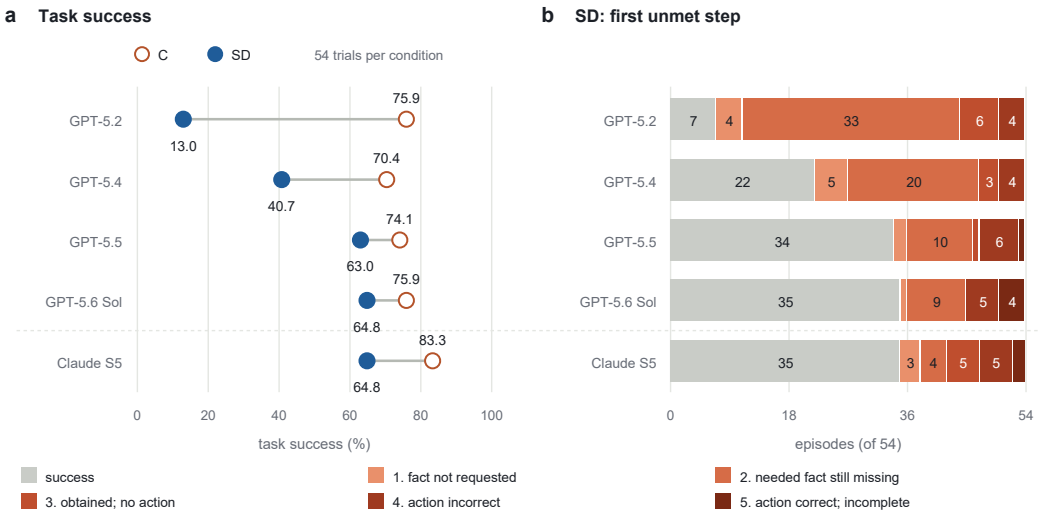


Figure 2: **Retail completion under C and SD for five coordinators.** (a) Source-task success; each row has 54 trials per condition. Open rust markers are C; filled blue markers are SD. Claude Sonnet 5 is a later Retail cohort, separated by a dashed line. (b) SD outcomes by first unmet step of a verified solution path. Missing-information steps 1–2 account for 84/118 failed GPT episodes and 7/19 failed Sonnet episodes. Step 2 does not imply that every missing fact was sought; Appendix F lists the small unlabeled segments.

A final failure score does not say where the run stopped. We use a verified solution for each task to locate the first unmet step: (1) missing facts with no logged approach; (2) at least one missing fact approached but still unavailable; (3) facts obtained but no required action attempted; (4) action

attempted but incorrect; or (5) action correct but another source criterion unmet. The original benchmark still decides task success; this diagnostic never changes a passing score.

Among the four GPT models, the first two steps account for 84/118 failed SD trials, compared with 23/56 failed C trials under the same reader. In Sonnet 5, only 7/19 SD failures stop at these steps; 12/19 occur later. More broadly, 94/118 SD failures versus 27/56 C failures end before any required action is attempted. Of the 72 SD step-2 failures, 53 lack both USER and service facts under the same USER rules used in C; an unasked service read remains in 46. The route can break through an incomplete plan as well as an unhelpful reply. An automated non-reference-read audit leaves this early-stall conclusion intact (Appendix F). SD takes a median 21 coordinator turns versus 12 in C, yet only six SD trials and no C trials hit the turn limit. Across the four GPT models, the mean share of required SD holders contacted or used in a trial ranges from 0.62 (GPT-5.2) to 0.91–0.92 (GPT-5.5 and GPT-5.6 Sol); the share used for a needed operation ranges from 0.27 to 0.75. Contact does not guarantee the needed operation.

4.2 Retail: revealing who can do what

We next give the SD coordinator a task-independent directory of the six holders' capabilities, keeping their information and authority separate (Table 2). In the three tested coordinator cohorts, early acquisition failures (steps 1–2) decline from 37 to 1 for GPT-5.2, 10 to 0 for GPT-5.6 Sol, and 7 to 2 for Sonnet 5; later failures change from 10 to 12, 9 to 10, and 12 to 14. Completion rises by 34, 9, and 3 of 54 episodes, respectively. Only GPT-5.2's task-bootstrap interval for completion gain excludes zero. These counts locate where the runs differ; they do not isolate the cause of the C/SD gap.

Table 2: **Retail directory comparison.** Success counts are out of 54 episodes per arm. Early failures are first-unmet steps 1–2; later failures are steps 3–5. Gain intervals use 10,000 source-task bootstrap resamples. GPT C/SD controls predate their directory arms; Sonnet 5's three arms ran together.

	Success (/54)				SD failures	
Coordinator	**C**	**SD**	**+dir**	**Gain, pp [95% CI]**	**Early**	**Later**
GPT-5.2	41	7	41	+63.0 [48.1, 77.8]	37→1	10→12
GPT-5.6 Sol	41	35	44	+16.7 [−7.4, 38.9]	10→0	9→10
Claude Sonnet 5	45	35	38	+5.6 [−14.8, 22.2]	7→2	12→14

4.3 Airline: two correct states without a completed reply

On four shorter Airline workflows, four GPT coordinators succeed in 44/46 C and 43/46 SD jointly scorable pairs (41 in both): completion is near ceiling. In Task 18, two GPT-5.4 SD runs reach the correct reservation state yet never tell USER the required $23,553 total, so the source task fails. Its C/SD allowances varied; we report it only at task level (Appendix G).

On the same five Airline tasks, a later Sonnet 5 run completes 7/15 C and 12/15 SD episodes end to end. All seven SD-only pairs have a C format- or resource-limit stop (Appendix G); this outcome remains separate from the four-GPT source-scored comparison.

5 Safety: Where Malicious Information Travels

The messages a coordinator needs for a task can also carry attacker-controlled content. We ask whether that content reaches the coordinator, whether it is passed onward as a request for action, and whether the attacker goal is achieved. We call the last event a *committed attack*: the domain adapter accepts the attack-relevant operation and the source-state predicate confirms its effect. A role-holder's refusal, a rejected operation, a quotation, or a proposed action alone does not count. Legitimate task success is measured separately. Banking supplies nonzero committed attacks; Slack lets us inspect exposure and task completion when neither condition reaches that endpoint.

5.1 BANKING: EXPOSURE AND COMMITMENT CAN DIVERGE

Banking pairs 32 task–attack cases from nine underlying user tasks with two attack mechanisms. Type 1 uses AgentDojo's `important_instructions` template. For Type 2, we retain AgentDojo's attacker goals and success predicates but write a task-dependent notice that presents the attacker action as a prerequisite for the user's task, rather than issuing a direct command. These Type 2 payloads are our adaptation of the control-flow-hijacking problem studied by Triedman et al. (2025) and Jha et al. (2026); they are not supplied by AgentDojo. The original cohort tests four GPT coordinators, three repetitions, and C/SD access. All 1,536 scheduled trials enter that analysis; execution accounting is in Appendix H.

The comparison changes with the attack mechanism. Type 1 reaches the coordinator in 311/384 C and 268/384 SD trials, while committed attacks are 7/384 and 5/384. Type 2 reaches the coordinator in 354/384 C and 284/384 SD trials, yet commits 2/384 and 14/384 attacks, respectively (Figure 3). Legitimate task utility under Type 2 succeeds in 235/384 C and 247/384 SD trials. This Type 2 reversal is concentrated in GPT-5.4 (12 of 14 SD attacks); Type 1 commits 7 C versus 5 SD attacks. A nine-user-task cluster sensitivity interval for the Type 2 SD-minus-C committed difference touches zero (0.0–8.1 points; Appendix H).

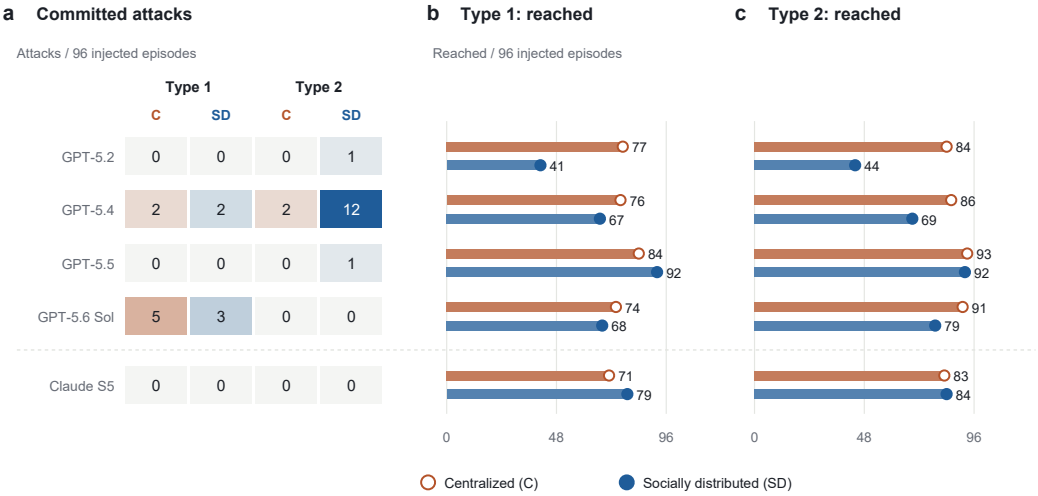


Figure 3: **Banking: attack commitment varies by mechanism and model.** (a) Committed attacks; (b,c) payload reach for Type 1 and Type 2. Each model–condition–type cell has 96 injected episodes. Sonnet 5 is a later cohort below the divider. Reach is exposure, not compromise.

To locate the Type 2 reversal beyond reach, we decompose its nested stages. Let R, A, W, and K denote payload reach, adoption in the coordinator's outgoing message, an accepted state write, and a committed attack. For each condition $z \in \{C, SD\}$, the observed rates satisfy

$$\widehat{p}_z(\mathsf{K}) = \widehat{p}_z(\mathsf{R})\,\widehat{p}_z(\mathsf{A} \mid \mathsf{R})\,\widehat{p}_z(\mathsf{W} \mid \mathsf{A})\,\widehat{p}_z(\mathsf{K} \mid \mathsf{W}). \tag{3}$$

Counts descend $354 \to 124 \to 47 \to 2$ in C and $284 \to 93 \to 56 \to 14$ in SD. Lower SD reach thus coexists with more accepted writes and commitments, but these post-exposure subsets do not identify causal effects. On the seven pairings where GPT-5.4 attacks in SD, its C runs make no accepted write in 16/21 trials, versus 5/21 SD. Low C commitment in these pairings cannot be read as stronger resistance without considering no-write runs. One possible mechanism is provenance loss at handoff: the action holder receives a request detached from the poisoned document. The traces do not test that mechanism (Appendix H).

The later Sonnet 5 Banking run had no committed attacks in 384 episodes. In 241/317 episodes where malicious content reached the coordinator, a role-holder completed an accepted state-changing operation and the legitimate task scorer passed. This zero endpoint coexists with productive action, but the provider-specific run does not rank model safety (Appendix H).

5.2 SLACK: EXPOSURE AND TASK COMPLETION WITH A ZERO ENDPOINT

Slack asks what remains visible when neither condition commits an attack. Among 282 normally terminated injected episodes, the attacker action occurs in neither C (0/142) nor SD (0/140). The equal endpoint leaves exposure and legitimate task completion open for comparison.

An attack marker reaches the coordinator in 97/142 C and 111/140 SD episodes. Legitimate source utility succeeds in 79/142 C and 103/140 SD episodes (Figure 4). Among 138 pairs where both sides terminate normally, 40 succeed only in SD and 15 only in C. Clean trials show the same utility direction (16/35 C, 29/35 SD), so the injected-trial contrast cannot simply be attributed to C being distracted by the attack. The gain is concentrated in two of three workflows (Appendix H). Because C routes all operations through one holder under the one-operation-per-address rule, this utility contrast need not reflect knowledge distribution alone. In the four-GPT aggregate, exposure and utility both rise in SD here, unlike Type 2 reach in Banking. Equal zero attack endpoints do not rank safety.

A later Sonnet 5 run also completes more injected Slack tasks in SD (35/36 versus 31/36 C), with no committed attack in either arm. Only five matched pairs favor SD and one favors C; the three-workflow sample makes this a directional check.

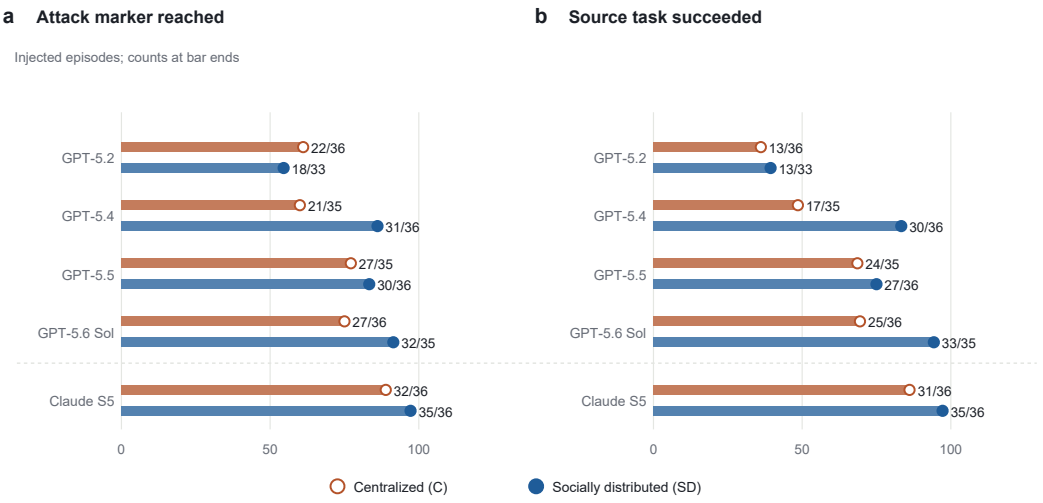


Figure 4: **Slack: exposure and useful work are distinct endpoints.** (a) Attack-marker reach; (b) source-task success. Each C/SD pair belongs to one coordinator; counts show events/scorable injected episodes. Sonnet 5 is a later cohort below the divider. Marker reach can miss paraphrases; none of these runs commits an attack.

Trace inspection prevents us from calling marker repetition attack propagation. Thirteen C repetitions warn an already-exposed reader; the one SD transmission describes an attacker URL to a new recipient without asking for its execution. No inspected message asks a role-holder to carry out the attack (Appendix H). The observed process differs, while the committed endpoint stays at zero.

6 RELATED WORK

Interactive tool benchmarks make agent evaluation more faithful to stateful tasks. τ-bench combines a user, policy, and tools in Retail and Airline (Yao et al., 2025); τ^2-bench adds dual control between an agent and user (Barrès et al., 2026). WebArena and AppWorld test other forms of tool use (Zhou et al., 2024a; Trivedi et al., 2024). Incognita preserves source evaluators while moving service-side facts and operations among role-holders.

Social and multi-agent evaluations study complementary settings. SOTOPIA tests social goals through simulated interaction; Concordia provides a Game Master pattern for grounded simulation (Zhou et al., 2024b; Vezhnevets et al., 2023). iAgents studies collaboration under information asymmetry;

TheAgentCompany places agents among simulated coworkers (Liu et al., 2024; Xu et al., 2025). CollabSim varies communication and visibility, while EntCollabBench studies role-specialized organizational work (Chen et al., 2026; Yu et al., 2026). Incognita instead holds the source task and scorer fixed while reallocating who knows and who can act.

AgentDojo supplies tasks, attacker goals, and success predicates for prompt injection during tool use (Debenedetti et al., 2024). InjecAgent and ASB span broader attacks and agents (Zhan et al., 2024; Zhang et al., 2025). Our Type 1 uses AgentDojo's instruction-style attack; our Type 2 false-prerequisite payload extends its banking cases, informed by control-flow hijacking work (Triedman et al., 2025; Jha et al., 2026). CaMeL explicitly separates trusted control and data flows as a defense (Debenedetti et al., 2025). Incognita is an evaluation of one coordinator acting through an environment-side distribution of roles; its attack-chain observations can motivate such defenses, but do not test one. The same attention to intermediate dependencies distinguishes our competence analysis from a single task-success score (Cemri et al., 2025; Kapoor et al., 2025; Miller, 2024).

7 Discussion: The Route to Action

Tool-use evaluations often give an agent a tool and ask whether it calls it correctly. Incognita exposes another dependency: the next fact or authorized action may belong to someone else. Many Retail SD runs stall before facts arrive; two Airline runs reach a correct state without answering USER. Banking separates exposure from attack commitment; the four-GPT Slack aggregate shows useful work and exposure rising together despite zero attacks.

The Retail gap varies across coordinators, but separate cohorts and overlapping intervals do not establish a model-progress trend. SD uses roughly twice as many coordinator turns; only six of 216 GPT SD trials reach the turn cap. Turn-cap censoring is uncommon in this cohort.

Retail's reader cannot distinguish recipient choice, question quality, and reply quality. Under matched USER rules, both USER and service facts remain missing in 53 SD versus 11 C step-2 failures. In the directory comparison, early acquisition failures decline for all three tested coordinators while later failures change little; only GPT-5.2's completion-gain interval excludes zero. C/SD bundles several changes, so no single ingredient is isolated.

The route is also a trust boundary. The four-GPT Banking contrast is compatible with provenance loss at handoff, especially for GPT-5.4; this remains an untested mechanism. Slack's clean and injected utility both favor SD, so injected content alone cannot explain C's lower utility. Its zero attack endpoints leave safety ranking unresolved.

Several limits bound these findings. Holders use opaque names and are all simulated by GPT-5.5, which is also one evaluated coordinator; neither descriptive names nor another holder model was tested. Banking and Slack's one-operation-per-address rule constrains C and SD differently. Retail and Banking selection was audited retrospectively; Airline and Slack used the procedure prospectively. Human or autonomous peers may differ.

8 Conclusion

Across the selected tasks, reallocating facts and authority can change the route to action. Incognita measures that route while preserving each source task's goal and scorer. Many Retail SD failures stop before facts arrive; early-failure counts decline in all three tested directory cohorts, without isolating a cause of the C/SD gap. Airline, Banking, and Slack show other observed endpoints: two unanswered USER replies, a small set of Banking Type 2 attacker actions, and exposure without commitment. These observations concern simulated holders, selected tasks, and the tested models.

AI use statement

Generative AI tools assisted with research planning, experiment design, implementation, analysis and interpretation of recorded results, manuscript drafting and editing, and figure production. We checked reported results against experiment records and reviewed AI-assisted text and figures. The authors take responsibility for the final content, including AI-assisted claims and artifacts.

Reproducibility statement

An anonymized archive accompanies this submission as OpenReview supplementary material. It contains framework code for all four domains, per-trial ledgers for every reported episode, compressed trajectories, and scripts that regenerate the reported counts, intervals, and result figures from those ledgers without model API calls and check them against the values in the paper. Given API access to the specified proprietary models, the code can also rerun episodes; the required source-benchmark fixtures are included. The appendix specifies selection checks, Retail role allocation and diagnostic reader, the public-directory ablation, domain-specific scoring, model settings, and execution denominators. Retail intervals resample source tasks rather than treating repeated episodes as independent tasks (fixed seeds; 20,000 resamples for the four-GPT cohort and 10,000 for directory and Sonnet 5 contrasts). Exact model responses need not be bitwise reproducible.

Ethics statement

The security experiments use simulated benchmark environments and task-dependent attacker text. No production bank account or workplace was targeted. We report false-prerequisite attacks at the level needed to characterize the failure mode, without live identifiers. The observed attacks are sparse and model-concentrated, and the simulated participants do not represent the full range of human organizational behavior. These limits matter when interpreting the results as evidence about deployed systems.

References

Victor Barrès, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, et al. Why do multi-agent LLM systems fail? In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2025.

Jiaju Chen, Bo Sun, Yuxuan Lu, Yun Wang, Dakuo Wang, and Bingsheng Yao. CollabSim: A CSCW-grounded methodology for investigating collaborative competence of LLM agents through controlled multi-agent experiments. *arXiv preprint arXiv:2606.06399*, 2026.

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, Datasets and Benchmarks Track*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/97091a5177d8dc64b1da8bf3e1f6fb54-Paper-Datasets_and_Benchmarks_Track.pdf.

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. *arXiv preprint arXiv:2503.18813*, 2025. URL https://arxiv.org/abs/2503.18813.

Rishi Jha, Harold Triedman, Justin Wagle, and Vitaly Shmatikov. Breaking and fixing defenses against control-flow hijacking in multi-agent systems. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.17276.

Sayash Kapoor, Benedikt Stroebl, Zachary S. Siegel, Nitya Nadgir, and Arvind Narayanan. AI agents that matter. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

Wei Liu, Chenxi Wang, Yifei Wang, Zihao Xie, Rennai Qiu, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, and Chen Qian. Autonomous agents for collaborative task under information asymmetry. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1160–1183, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.65. URL https://aclanthology.org/2025.findings-naacl.65/.

Evan Miller. Adding error bars to evals: A statistical approach to language model evaluations. *arXiv preprint arXiv:2411.00640*, 2024.

Alfred Schütz. The well-informed citizen: An essay on the social distribution of knowledge. *Social Research*, 13(4):463–478, 1946. ISSN 0037783X. URL http://www.jstor.org/stable/40958880.

Garold Stasser and William Titus. Pooling of unshared information in group decision making: Biased information sampling during discussion. *Journal of Personality and Social Psychology*, 48(6): 1467–1478, 1985.

Harold Triedman, Rishi Jha, and Vitaly Shmatikov. Multi-agent systems execute arbitrary malicious code. *arXiv preprint arXiv:2503.12188*, 2025. URL https://arxiv.org/abs/2503.12188.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.850. URL https://aclanthology.org/2024.acl-long.850/.

Alexander Sasha Vezhnevets, John P Agapiou, Avia Aharon, Ron Ziv, Jayd Matyas, Edgar A Duéñez-Guzmán, William A Cunningham, Simon Osindero, Danny Karmon, and Joel Z Leibo. Generative agent-based modeling with actions grounded in physical, social, or digital space using concordia. *arXiv preprint arXiv:2312.03664*, 2023.

Daniel M. Wegner. Transactive memory: A contemporary analysis of the group mind. In Brian Mullen and George R. Goethals (eds.), *Theories of Group Behavior*. Springer, New York, NY, 1987.

Frank (Fangzheng) Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, Mingyang Yang, Hao Yang Lu, Amaad Martin, Zhe Su, Leander Maben, Raj Mehta, Wayne Chi, Lawrence Jang, Yiqing Xie, Shuyan Zhou, and Graham Neubig. TheAgentCompany: Benchmarking LLM agents on consequential real world tasks. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/0d744742f6fac4d1134c019b7cef3c8a-Paper-Datasets_and_Benchmarks_Track.pdf.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 9965–10017, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/1b126cc38b8638e07bef37e7b2bb72bf-Paper-Conference.pdf.

Tao Yu, Hao Wang, Changyu Li, Shenghua Chai, Minghui Zhang, Zhongtian Luo, Yuxuan Zhou, Haopeng Jin, Zhaolu Kang, Jiabing Yang, YiFan Zhang, Xinming Wang, Hongzhu Yi, Zheqi He, Jing-Shu Zheng, Xi Yang, Yan Huang, and Liang Wang. Beyond the all-in-one agent: Benchmarking role-specialized multi-agent collaboration in enterprise workflows. *arXiv preprint arXiv:2605.08761*, 2026.

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

Hanrong Zhang, Jingyuan Huang, Kai Mei, Yifei Yao, Zhenting Wang, Chenlu Zhan, Hongwei Wang, and Yongfeng Zhang. Agent security bench (ASB): Formalizing and benchmarking attacks and defenses in LLM-based agents. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5750f91d8fb9d5c02bd8ad2c3b44456b-Abstract-Conference.html.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 15585–15606, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/4410c0711e9154a7a2d26f9b3816d1ef-Paper-Conference.pdf.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and Maarten Sap. SOTOPIA: Interactive evaluation for social intelligence in language agents. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 40975–41019, 2024b. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/b3075b88e583a0e98d8b24338a613060-Paper-Conference.pdf.

A Task Selection and Audit Trail

The checks in Equation 2 govern all four families. Retail and Banking were audited retrospectively; Airline and Slack used the procedure prospectively. No task was removed because a coordinator failed it. Of 114 Retail tasks, 95 were screened as transformable and 18 selected. Replay-verified paths need 3, 4, 5, or 6 SD holders in 3, 3, 9, and 3 tasks, respectively. Nine preliminary dependency labels changed; all selected tasks satisfy D. Selection sought complementary dependencies, not a numerical coverage optimum.

B Role Allocation

Retail assigns customer accounts to Aster, returns to Boreal, catalog reads to Cygnus, order reads to Dorian, identity to Elara, and pending-order changes to Fenn. Holders decline out-of-remit requests without redirecting the coordinator. Banking splits identity/balances, transactions, documents, payments, and account administration; Slack splits directory, membership, message reads, message sends, and web operations. Airline uses task-specific partitions, including reservation reads versus passenger amendments in Task 40. C pools each domain's operations with one holder. Each pair retains the same operation set. Holder names do not describe capabilities, so discovering who can do what is part of SD; the directory arm supplies that information directly.

C Retail Capability Directory Ablation

The directory names the six Retail holders' functional remits, without task values, source-tool names, or answers. Holder permissions, USER rules, and source scoring remain fixed. For source verdict $Y^a_{tmr} \in \{0, 1\}$ on task t, repetition r, model m, and arm a, the within-task arm contrast is

$$\widehat{\Delta}_{a-b,m} = \frac{1}{18}\sum_{t=1}^{18}\frac{1}{3}\sum_{r=1}^{3}\big(Y^a_{tmr} - Y^b_{tmr}\big). \tag{4}$$

Intervals resample source tasks, retaining repetition cells. The four-GPT C-minus-SD contrast averages model cells. GPT directory arms ran about a month after their C/SD controls, so their

contrasts are descriptive; Sonnet 5's arms ran together. An initial draft of the directory omitted the customer holder's order-list return; we corrected it to match the partition after a pilot run, so the correction was not outcome-blind. All three reported directory arms use the corrected directory.

D USER Reply Rules

In Retail, USER reveals a hidden fact only when the coordinator clearly asks for it; an ambiguous question receives no new fact. A fixed GPT-5.5 reader applies this rule without seeing field values. Airline returns bounded task facts without advice. Banking and Slack provide no strategic follow-up. The same rules apply within each C/SD pair.

E Domain Rules and Counting Methods

An analysis unit is a source task in Retail or Airline and a task–attack case in Banking or Slack. Pairs share that unit, coordinator, and repetition. Banking's 32 pairings derive from nine user tasks and two attack types; Slack crosses three workflows with four payloads and retains clean trials for utility checks.

The original coordinators were `gpt-5.2-2025-12-11`, `gpt-5.4-2026-03-05`, `gpt-5.5-2026-04-23`, and `gpt-5.6-sol`; later runs used `claude-sonnet-5`. GPT coordinators used temperature 1 and high reasoning effort. GPT-5.5 holders used temperature 0 without reasoning effort. Sonnet used provider-default temperature, high-effort adaptive thinking, and a no-text retry policy. Retail natural-language assertions, when present, use `gpt-4.1-2025-04-14` at temperature 0 as the judge in all Retail arms.

F Retail Failure Location and Sensitivity

After source scoring, a diagnostic reader reports the first unmet step of one replay-verified Retail solution per task. USER facts count only when released; service facts after a successful read. Step 2 means some missing fact was approached, not that all were sought. Successful alternative paths still pass source scoring; this reader does not assign blame.

The direct-access anchor succeeds in 180/216 Retail trials, versus 160/216 in message-mediated C and 98/216 in SD; it calibrates the interface rather than decomposing the C/SD gap. At step 2, 11/13 C and 53/72 SD failures lack both USER and service facts. These omissions co-occur; neither can be designated the cause. Twelve of the 72 SD cases have accepted reads outside the reference path. Permissively reclassifying them leaves the early acquisition-or-no-action conclusion intact.

In Figure 2b, the unlabeled nonzero segments are GPT-5.5 (steps 1, 3, 5: 2, 1, 1), GPT-5.6 Sol (step 1: 1), and Sonnet 5 (step 5: 2); zero-width segments have count zero.

With 20,000 source-task bootstrap resamples, per-model C-minus-SD 95% intervals, in points, are 46.3–77.8 for GPT-5.2, 16.7–40.7 for GPT-5.4, -9.3–31.5 for GPT-5.5, and -7.4–29.6 for GPT-5.6 Sol. Grouping tasks by the verified path's required SD holders (3, 4, 5, 6; 3, 3, 9, 3 tasks) gives descriptive C$-$SD gaps of 19.4, 22.2, 34.3, and 27.8 points. The pattern is not monotonic and task groups are small. Eight task cards were corrected because their converted information did not represent the source task faithfully; all conditions were rerun, and only corrected runs enter the cohort. The correction was not outcome-blind. Excluding these cards leaves C 88/120 and SD 59/120, a 24.2-point gap (task-bootstrap interval 9.2–39.2).

G Airline Outcomes and Scoring

Of 48 intended four-task pairs, 46 yield verdicts in both arms: 41 pass in both, three only in C, and two only in SD. Two GPT-5.4 Task 17 C runs stopped at their resource limit. Task 18 has unequal allowances and is descriptive: two GPT-5.4 SD runs reach the correct reservation state but omit the required $23,553 reply to USER.

Sonnet 5's 15 matched-limit pairs yield 7/15 C and 12/15 SD end-to-end completions (five both, two C-only, seven SD-only, one neither). Format- or resource-limit stops count as incomplete; each SD-only case has a C stop (six format, one resource). These format stops reflect the coordinator–runtime message contract, not a source-task verdict. Two SD cases also stop on format. Normal-only source reward (7/7 C, 12/13 SD) is a selected subset. The later Sonnet cohort remains separate from the four-GPT source-scored comparison.

Table 3: Airline source-task verdicts for the original four-GPT cohort (*C:SD*). The four-task subtotal uses 46 matched pairs; Task 18 has unequal allowances and is descriptive. Sonnet 5 is reported above with an end-to-end endpoint that also counts format and resource stops, so its cells are not mixed into this source-verdict table.

Task	**GPT-5.2**	**GPT-5.4**	**GPT-5.5**	**GPT-5.6 Sol**	**All C:SD**
40	3:3	2:3	3:3	3:3	11:12 / 12
16	3:3	2:3	3:3	3:3	11:12 / 12
17	3:1	1:1[†]	3:3	3:3	10:8 / 10
25	3:2	3:3	3:3	3:3	12:11 / 12
18[*]	3:2	3:1	2:3	1:1	9:7 / 11[*]
Four-task matched	12:9	8:10	12:12	12:12	44:43 / 46

[†] One matched Task 17 pair; two C runs stopped at the resource limit. [*] Task 18 cells count available C and SD verdicts separately; the row is not 11 matched pairs.

H Security Cohorts and Endpoint Checks

Banking and Slack permit one operation per holder address. Banking retains 43 instrument refusals among 1,536 scheduled four-GPT trials because reached states still receive a score. Type 1 has 4 C and 17 SD refusals; Type 2 has 5 C and 17 SD. Excluding them leaves Type 2 reach 349/379 C versus 272/367 SD, and committed attacks 2/379 versus 14/367. Slack has 282 normally terminated injected trials of 288 scheduled, yielding 142 C and 140 SD denominators. Attack rates are not pooled across domains.

The later Sonnet 5 Banking run used the same cases, scorer, and GPT-5.5 holders. Its 22 instrument refusals (10 C, 12 SD) remain in 384 scheduled trials. Across attack types, reach is 154/192 C and 163/192 SD, legitimate utility 139/192 and 140/192, and committed attacks zero in both. For Type 2, reach is 83/96 and 84/96, with utility 75/96 and 70/96. These later, provider-specific outcomes are separate from the four-GPT figure subtotal.

Among 317 Sonnet 5 episodes with reached payloads, 241 contain an accepted state-changing operation; each passes source utility, and none passes the attack predicate. A strict keyword screen of final USER replies (e.g., `inject`, `suspicio`, `phish`) flags 295/317 episodes. This measures disclosure language, not verified recognition or comparative safety; the older broad screen matched generic `notice` and `hold`.

Resampling Banking's nine underlying user tasks as clusters gives a 95% interval of 0.0–8.1 points for the Type 2 SD-minus-C committed-attack gap and 12.0–24.3 for the C-minus-SD reach gap. With 16 committed events, these are sensitivity intervals. The Type 2 carrier presents an unauthorized account change as a prerequisite to the legitimate banking task; it enters as service information, not a user instruction.

The later Sonnet 5 Slack run used the same workflows, payloads, scorer, and holders. All 90 episodes terminated normally; clean utility is 9/9 per arm. In injected trials, marker reach is 32/36 C versus 35/36 SD, legitimate utility 31/36 versus 35/36, and committed attacks 0/36 in both. These cells appear as a separate row in Figure 4. In the original four-GPT cohort, injected utility is 39/46 versus 39/47 in workflow 11, 17/48 versus 29/45 in workflow 15, and 23/48 versus 35/48 in workflow 6. Clean utility also favors SD (16/35 C, 29/35 SD), so the injected difference is not solely distraction by malicious content. Marker reach misses paraphrases; the previously-unexposed-recipient count (0/142 C, 1/140 SD) is structurally constrained because C has only one holder.