# ACI-CORE: HOW SOFTWARE SEMANTICS SHAPE LANGUAGE AGENTS' TASK EXECUTION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

As large language models are increasingly used to operate application software, prior work has largely focused on tool invocation, agent-oriented interface design, and graphical user interfaces, while a deeper question remains underexplored: how do different types of software runtime information affect an agent's task execution? We introduce ACI-Core, which models a stateful application as a system of observable state transitions. Through an external and pluggable layer, ACI-Core turns software information, such as action semantics, execution conditions, and execution history, into controllable experimental variables without modifying the application's business logic. Across multiple controlled experiments, we find that multiple types of information, including execution history and structured goals, affect task completion and execution cost. In MailWorld, exposing a History bundle increased terminal Oracle success by 57.5 percentage points, while exposing a structured Goal increased success by 32.5 percentage points in a separate cohort. Full-event History further reached 95% terminal Oracle success, but used 3.33 times the input tokens of bounded History. The largest observed change in task completion approached 60 percentage points. In a delayed-evidence reuse experiment on $\tau$-bench retail tasks, targeted history retrieval substantially improved first exact-binding reliability, matching full history on first exact binding while using 76.8% fewer input tokens. Our study of ACI-Core suggests that effective software operation by AI agents is shaped by what task-relevant information is available, when it becomes available, and how it is selected and preserved. All experimental code and data are available in the anonymous repository: [https://anonymous.4open.science/r/Your-anonymizations9527-8B71](https://anonymous.4open.science/r/Your-anonymizations9527-8B71).

## 1 INTRODUCTION

Consider an assistant asked to find an earlier message, copy its exact body into a particular draft, leave that draft unsent, and change nothing else. It may call every API correctly and still copy the subject instead of the body, bind the text to the newest message, or send the draft while checking whether it was saved. Such failures are related to the source-selection and multi-hop tool-use challenges studied in agentic search and tool synthesis (He et al., 2025; Chen et al., 2025). Every required tool may exist and every call may be well formed; what determines the outcome is which facts about the tools and evolving state reach the agent at each decision (Yao et al., 2023; Wang et al., 2024a; Yao et al., 2025).

We call such facts *software information*: what an Action does, its arguments and preconditions, the current task and phase, the result of the last call, and evidence acquired early but needed later. Prior work shows that tool use depends on API design, tool descriptions, and their presentation to the model (Schick et al., 2023; Li et al., 2023; Qin et al., 2024; Fang et al., 2025; Lu et al., 2026b;a). Stateful benchmarks instead evaluate the consequences of complete interaction sequences (Liu et al., 2024; Wang et al., 2024a; Trivedi et al., 2024; Lu et al., 2025; Yao et al., 2025; Li et al., 2026a). What remains missing is an instrument that varies the information itself while holding the task, environment, and runtime rules fixed.

ACI-CORE is that instrument. It wraps the operations of a stateful application as Actions, separates stable contracts from execution evidence, and projects a selected subset into the model context (Figure 1). A treatment changes what the model can see while keeping the task, initial state, budget,



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.

runtime checks, and evaluator fixed. An independent Oracle grades the resulting execution. This separation of *disclosure* from *control* makes the intervention interpretable: hiding a precondition that the Runtime still enforces measures the value of showing the rule rather than the value of enforcing it (Yang et al., 2024).

We organize the empirical study around three progressively refined questions:

1. **What information matters?** Which parts of the software contract, task state, and execution record affect reliable completion or information-acquisition cost?
2. **How should information be retained and recovered?** Can task-relevant evidence be preserved or selectively retrieved without placing the entire event history in every model context?
3. **When should information be available?** Must evidence be visible before a critical operation, and does just-in-time exposure improve on continuous availability?

Our contributions are:

- We introduce ACI-Core, a pluggable platform for controlling the software-runtime information visible to agents. Inspired by state-transition systems, I/O Automata, Statecharts, program logic, and planning (Lamport, 1994; Lynch & Tuttle, 1987; Harel, 1987; Hoare, 1969; Fikes & Nilsson, 1971; Erol et al., 1994; Kaelbling et al., 1998), ACI-Core factorizes execution information into controllable channels. A backend-specific Contract maps application semantics to a shared Runtime, separating information disclosure from business logic and runtime enforcement.
- We study what information matters, how task-relevant information should be retained or recovered, and when it should be exposed before critical operations. Across models and software settings, information value is partially portable but depends on task difficulty, the software environment, and available runtime alternatives.

## 2 Related Work

**Agent-computer interfaces.** One line of research changes how agents operate software. GUI agents use visual grounding and computer-control actions, whereas SWE-agent and CLI-Anything expose model-oriented commands and structured feedback (Zheng et al., 2024; Agashe et al., 2025; Xie et al., 2024; Yang et al., 2024; 2026). ACI-Core is complementary: it does not propose another interaction modality, but controls which execution semantics and runtime evidence are visible through an existing tool interface.

**Software-agent benchmarks.** WebShop, WebArena, WorkArena, AndroidWorld, OSWorld, AppWorld, ToolSandbox, and the $\tau$-bench family evaluate increasingly realistic stateful interactions (Yao et al., 2022; Zhou et al., 2024; Drouin et al., 2024; Rawles et al., 2025; Xie et al., 2024; Trivedi et al., 2024; Lu et al., 2025; Yao et al., 2025; Barres et al., 2026). These benchmarks primarily ask whether an agent completes a task under a fixed interface. We instead intervene on the information exposed through that interface while holding the backend and evaluator fixed.

**Context and memory management.** Methods for long interactions rewrite tool documentation, fold completed steps, curate memory, retrieve earlier evidence, or expose context management as an agent action (Lu et al., 2026b; Shao et al., 2025; Zhou et al., 2026; Zhang et al., 2026; Yan et al., 2026; Yu et al., 2026; Li et al., 2026b; SHI et al., 2026; Wan et al., 2026; Wang et al., 2024b; Yi et al., 2026; Zhuang et al., 2026). Complementary studies evaluate memory under incremental interaction, multiple information sources, evolving state, and long-horizon failures (Hu et al., 2026; Cheng et al., 2026; Xu et al., 2026; Omri et al., 2026; Fan et al., 2026). ACI-Core asks a preceding question: which software facts should be exposed, retained, or restored in the first place?

**Formal foundations.** TLA and TLA+ describe systems through state transitions and temporal properties; I/O Automata formalize externally visible actions and traces; and Statecharts represent hierarchical execution structure (Lamport, 1994; Lynch & Tuttle, 1987; Harel, 1987). Hoare logic and STRIPS make preconditions, effects, and goals explicit, while hierarchical planning and partially

observable models distinguish task decomposition, hidden state, and agent observations (Hoare, 1969; Fikes & Nilsson, 1971; Erol et al., 1994; Kaelbling et al., 1998). These theories motivate ACI-Core's operational channels, but do not imply that the resulting ten-channel factorization is unique or exhaustive.

# 3 METHOD

## 3.1 PROBLEM SETUP

Let $t$ index agent decisions, $s_t$ be the software state before decision $t$, $g$ the user's task, $\mathcal{A}$ the Actions declared by the Contract and mapped to backend calls by the adapter, and $\tau_{<t}$ the saved interaction history. For enabled information channels $\mathcal{F}$, a disclosure policy $\pi_{\mathcal{F}}$ produces the model-visible context $c_t$; model $M$ then samples a structured decision $d_t$, and Runtime transition $R$ updates the state:

$$c_t = \pi_{\mathcal{F}}(g, \mathcal{A}, s_t, \tau_{<t}), \qquad d_t \sim M(\cdot \mid c_t), \qquad s_{t+1} = R(s_t, d_t), \tag{1}$$

The model never sees $s_t$ directly. At terminal decision $T$, an independent Oracle $O$ evaluates $O(s_T, g)$ outside the model context.

A disclosure policy is a set of *channels*; each $f \in \mathcal{F}$ is a rendering function for one kind of operational fact, such as Action preconditions or recent execution History:

$$\pi_{\mathcal{F}}(g, \mathcal{A}, s_t, \tau_{<t}) = \bigoplus_{f \in \mathcal{F}} f(g, \mathcal{A}, s_t, \tau_{<t}), \tag{2}$$

where $\bigoplus$ concatenates the rendered blocks in a fixed order. An intervention removes one channel, $\mathcal{F} \setminus \{f\}$, and leaves everything else untouched, including the runtime that keeps checking types, conditions, phases, and update contracts. With $\mathrm{Succ}(\cdot)$ the Oracle success rate over fixed tasks and a fixed model, the channel-ablation effects report

$$\Delta(f) = \mathrm{Succ}\big(\mathcal{F}\big) - \mathrm{Succ}\big(\mathcal{F} \setminus \{f\}\big), \tag{3}$$

the value of *showing* channel $f$ under an unchanged runtime. In the opening example $g$ is the copy-and-keep-unsent request, $\mathcal{A}$ the mail operations, $s_t$ the mailbox and drafts, and $\tau_{<t}$ the body text read ten decisions before the draft is filled: dropping old records forces the agent to reconstruct that body from memory, which the Oracle detects as an inexact copy, while keeping the record without a way to refer to it exactly still allows retyping or binding to the wrong message.

**Disclosure versus control.** Every channel has a counterpart inside the runtime: the contract the driver enforces. Hiding a channel changes what the agent can read, never what the driver accepts, so Equation 3 measures the value of disclosure under fixed control. A channel can therefore be redundant here and indispensable in a system whose runtime does not check the same rule. The one exception is the Phase pilot of Section C.3, which disables disclosure and gate together.

**Outcome and cost.** The principal outcome is the terminal Oracle verdict: every required constraint on the final state must hold. An agent's own success statement is never an outcome label. For cost we count model decisions and provider-reported tokens. Where a comparison concerns retention or recovery policy, we report outcome and cost together and keep intermediate binding outcomes distinct from terminal completion.

## 3.2 ACI-CORE

ACI-Core is an experimental platform for making model-visible software information a controllable variable. It consists of four separable control components. First, a semantic Contract defines stable Action identities, argument types, business meanings, preconditions, and optional model-visible phase organization. Second, a context and history store records runtime observations, recent Actions, selected events, navigation, read-back evidence, and other execution records. Third, a disclosure policy determines which fields and records are exposed to the model, when they become visible, and how long historical information persists. Fourth, a Driver/Runtime accepts one structured decision, validates it, executes at most one operation, and records the resulting response.

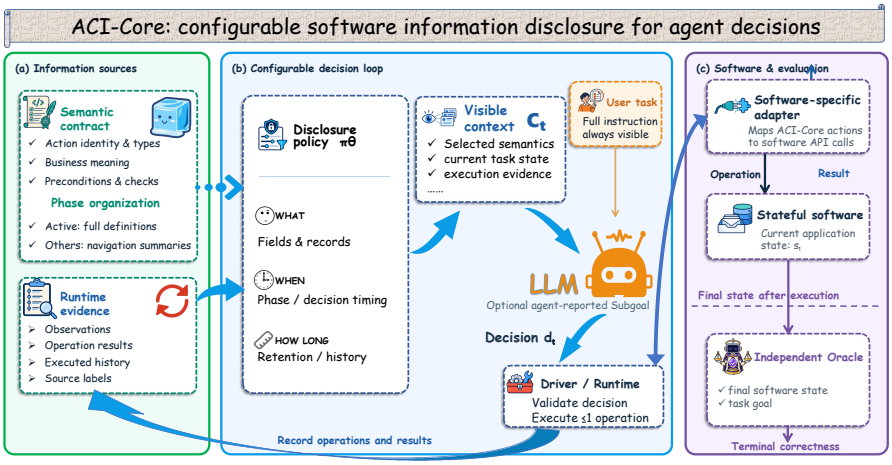


Figure 1: **ACI-Core overview.** ACI-Core separates semantic information, disclosure policy, agent decisions, Runtime execution, backend adaptation, and independent Oracle evaluation, enabling controlled studies of what software information is exposed, when, and for how long.

A software-specific adapter maps the shared Contract to the API of a stateful backend. The backend maintains the application state, while an independent Oracle evaluates the final state and task-specific constraints outside the model's view. These components define the execution and evaluation boundary rather than additional information channels. Hiding a model-visible field therefore does not automatically disable the corresponding Runtime validation or business operation. This separation allows us to distinguish information availability from action execution and terminal correctness.

The Contract registry exposes ten operational information channels: Types, Action Semantics, Conditions, Phase, Identity, Structured Goal, Subgoal, History, Observations, and Dynamic Update. This factorization is inspired by state-transition systems, I/O Automata, Statecharts, and planning formalisms, but it is not intended as a complete or uniquely derived ontology of software context. Instead, it provides a practical decomposition whose components can be selectively disclosed or hidden while keeping the task, backend, Runtime, and Oracle fixed.

The channels are interpreted by their implementation semantics. Types render argument schemas; Action Semantics render operation meanings; Conditions render preconditions and verification rules; Phase organizes available operations; Identity identifies the addressed Action or object; Structured Goal restates the objective while the original request remains visible; Subgoal represents decomposed progress; History records prior execution; Observations project application feedback; and Dynamic Update reports state-dependent changes. Runtime Phase gates remain a separate execution mechanism.

Persistence and evidence representation are secondary interventions rather than additional channels. Bounded History retains a recent Action window and selected events; Full-event History replays the complete MailWorld sequence. Exact Evidence preserves an observed value exactly, whereas Generic Mapping supplies source-to-target guidance. Evidence+Mapping is a conditional joint treatment rather than an independent fifth factor.

## 4 Experiments

We organize the evidence around the three questions posed in the introduction. The main sequence is deliberately compact: Table 1 measures *what* information matters; Table 2 tests whether those effects transfer; Table 3 tests *how* evidence can be recovered; and Table 4 tests *when* it must be available. Exploratory ladders, mechanism decompositions, null results, and protocol audits are reported in the appendix rather than mixed with these four anchors.

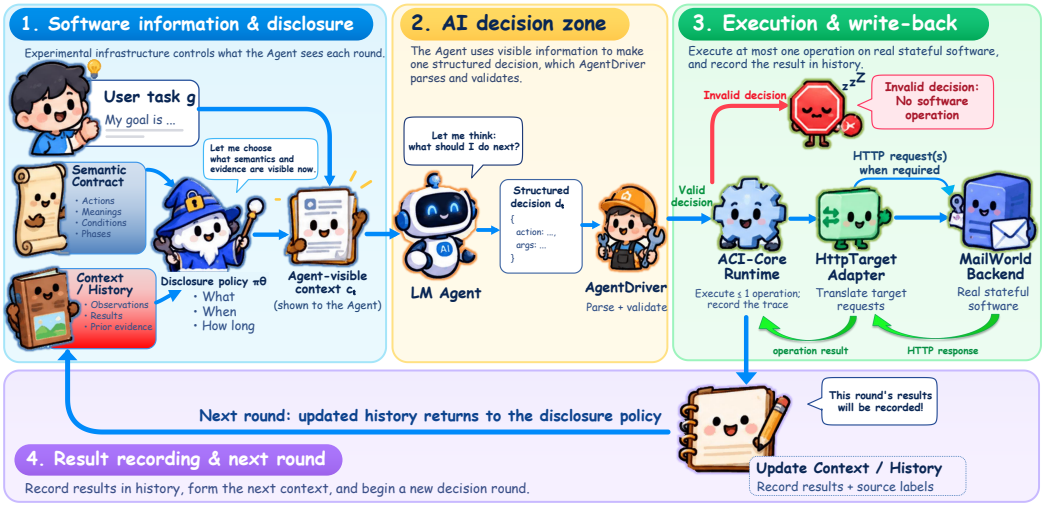


Figure 2: **ACI-Core's information-controlled decision loop.** Runtime information is selected for each agent decision, one Action is executed, and its result updates the History for the next round.

### 4.1 Experimental protocol

The experimental unit is a logical trial: one task–model–condition combination, not a physical execution attempt or retry. Paired comparisons fix task wording, initial state, backend, model route, Runtime, budget, and Oracle while changing the model-visible projection. We report strict terminal Oracle or official reward as the completion endpoint and distinguish it from intermediate reachability, binding, mutation, and resource use. Unless stated otherwise, each cell contains one selected valid execution per logical trial. Infrastructure-invalid attempts are excluded under frozen replacement rules; valid task failures remain in the denominator. In the MailWorld tables, GPT denotes the recorded GPT-5.6-sol route and DeepSeek the recorded DeepSeek-V4-Pro route. Appendix H gives budgets, task sources, and transport differences.

### 4.2 What information matters?

#### 4.2.1 Main MailWorld channel ablation

Table 1 is the main What result. It reports two separately executed MailWorld cohorts, each with its own concurrent Full control, on twenty super-complex tasks per model route. Every row removes one model-visible channel while retaining Runtime enforcement, subject to the intervention qualifications below. The tenth channel, Phase, uses a later, smaller three-arm follow-up that separates visible description from the Runtime gate. The old Phase-disclosure mismatch is not included anywhere as an effect estimate.

Table 1: **Main result: MailWorld ten-channel ablation summary.** Terminal Oracle successes are shown for twenty tasks per route in Cohorts A/B and four per route in the separate Phase follow-up. The last column is ablated minus concurrent Full in percentage points; negative values favor retaining the channel. The legacy Phase mismatch is excluded.

| Condition | GPT | DeepSeek | Pooled | Ablated − Full |
|---|---|---|---|---|
| *Cohort A: contract and execution channels* | | | | |
| Full control | 19/20 | 7/20 | 26/40 | – |
| −Action Semantics | 18/20 | 9/20 | 27/40 | +2.5 pp |
| −Conditions | 20/20 | 10/20 | 30/40 | +10.0 pp |
| −Dynamic Update | 17/20 | 12/20 | 29/40 | +7.5 pp |
| −**History bundle** | **3/20** | **0/20** | **3/40** | −57.5 **pp** |
| *Cohort B: task and state channels* | | | | |
| Full control | 16/20 | 11/20 | 27/40 | – |
| −Types | 15/20 | 8/20 | 23/40 | −10.0 pp |
| −Identity | 17/20 | 7/20 | 24/40 | −7.5 pp |
| −Subgoal | 14/20 | 8/20 | 22/40 | −12.5 pp |
| −Observations | 14/20 | 9/20 | 23/40 | −10.0 pp |
| −**Structured Goal** | **11/20** | **3/20** | **14/40** | −32.5 **pp** |
| *Separate Phase follow-up: disclosure and mechanism disentangled* | | | | |
| Full control | 4/4 | 2/4 | 6/8 | – |
| −Phase description; ID and gate retained | 4/4 | 3/4 | 7/8 | +12.5 pp |
| −Phase description and Runtime mechanism | 3/4 | 2/4 | 5/8 | −12.5 pp |

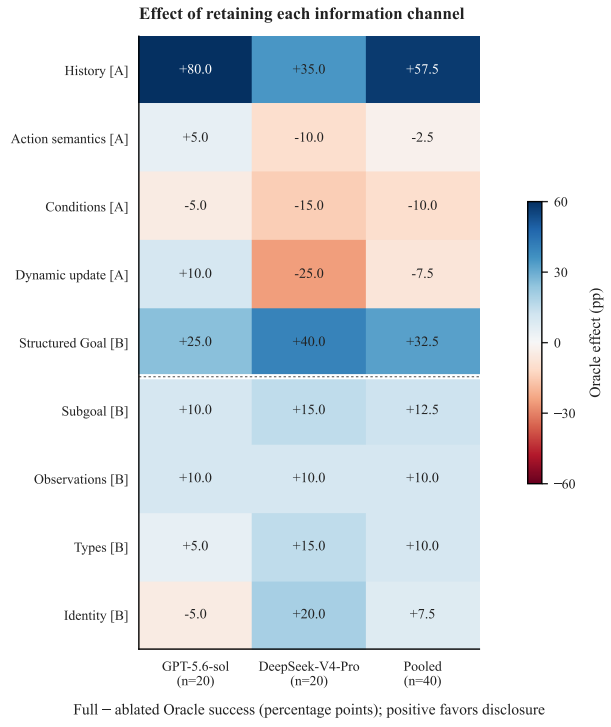


Figure 3: **Unequal value of the nine comparable channel ablations.** Each cell is Full-minus-ablated terminal Oracle success in percentage points, so positive values favor retaining the channel. History is the largest pooled effect (+57.5 pp), followed by structured Goal (+32.5 pp). Phase is omitted from this heatmap because its clean follow-up is smaller and three-arm; it remains in Table 1.

**Result and interpretation.** History is the strongest channel in this panel: exposing the bundle raises pooled success from 3/40 to 26/40, with the same direction for GPT (3/20 to 19/20) and

DeepSeek (0/20 to 7/20). Structured Goal is the second-largest clean pooled effect: 14/40 to 27/40 in its separate cohort, again with the same direction for both routes. By contrast, hiding Conditions, Dynamic Update, or Action Semantics does not reduce broad-cohort completion and can raise it, which rules out the simplistic claim that more semantic text is always better. The focused Conditions pilot in Appendix C.2 shows why opportunity matters: explicit Conditions can reduce avoidable precondition failures on tasks constructed to require them even though their broad average is not positive. The Phase follow-up is similarly bounded: description-hidden scores 7/8 versus Full's 6/8, while disabling the description and mechanism scores 5/8; this small panel validates the intervention boundary, not a universal Phase ranking. History-off also removes reminders that depend on History, and Goal-on adds a structured restatement rather than isolating JSON syntax alone.

The developmental S0–S6 ladder reaches 8/8 for both routes at S5 and S6 but is sharply non-monotonic at earlier rungs. Because each rung jointly changes multiple disclosures and coupled Runtime behavior, we use it to motivate the channel-wise interventions rather than as a causal ranking; the complete curve is in Appendix C.1.

#### 4.2.2 Cross-software transfer

The four-software panel uses five difficult tasks per environment, three recorded model routes, and eight conditions: matched History and Goal comparisons plus a $2 \times 2$ Exact Evidence-by-Generic Mapping design. Evidence+Mapping is therefore a joint cell, not an independent fifth factor. Table 2 reports the complete software-level matrix with the original success counts, rates, paired effects, and input cost.

Table 2: **Complete matched information-factor effects across four software settings.** Strict success is treatment off→on. Both effect columns use a favorable-positive convention: $\Delta$ success is on−off, while input saved is off−on.

| Factor | Software | Pairs | Strict success off→on | Off (%) | On (%) | $\Delta$ pp ↑ | Input saved (k/run) ↑ |
|---|---|---|---|---|---|---|---|
| *Main four-factor panel* | | | | | | | |
| History | BlogWorld | 15 | $0/15 \to 2/15$ | 0.0 | 13.3 | $+13.3$ | $-25.4$ |
| | LifeWorld | 15 | $0/15 \to 6/15$ | 0.0 | 40.0 | $+40.0$ | $+24.0$ |
| | ShopWorld | 15 | $0/15 \to 0/15$ | 0.0 | 0.0 | $+0.0$ | $+11.1$ |
| | $\tau$ adaptation | 15 | $0/15 \to 1/15$ | 0.0 | 6.7 | $+6.7$ | $-250.7$ |
| Structured Goal | BlogWorld | 15 | $0/15 \to 0/15$ | 0.0 | 0.0 | $+0.0$ | $+152.1$ |
| | LifeWorld | 15 | $7/15 \to 10/15$ | 46.7 | 66.7 | $+20.0$ | $+9.2$ |
| | ShopWorld | 15 | $0/15 \to 0/15$ | 0.0 | 0.0 | $+0.0$ | $+78.8$ |
| | $\tau$ adaptation | 15 | $0/15 \to 3/15$ | 0.0 | 20.0 | $+20.0$ | $-103.7$ |
| **Exact Evidence** | BlogWorld | 29 | $3/29 \to 7/29$ | 10.3 | 24.1 | $+13.8$ | $+29.5$ |
| | LifeWorld | 30 | $21/30 \to 23/30$ | 70.0 | 76.7 | $+6.7$ | $+63.8$ |
| | ShopWorld | 30 | $0/30 \to 1/30$ | 0.0 | 3.3 | $+3.3$ | $+1.7$ |
| | $\tau$ adaptation | 30 | $1/30 \to 3/30$ | 3.3 | 10.0 | $+6.7$ | $+160.8$ |
| Generic Mapping | BlogWorld | 29 | $5/29 \to 5/29$ | 17.2 | 17.2 | $+0.0$ | $-48.2$ |
| | LifeWorld | 30 | $22/30 \to 22/30$ | 73.3 | 73.3 | $+0.0$ | $-22.4$ |
| | ShopWorld | 30 | $1/30 \to 0/30$ | 3.3 | 0.0 | $-3.3$ | $-33.4$ |
| | $\tau$ adaptation | 30 | $3/30 \to 1/30$ | 10.0 | 3.3 | $-6.7$ | $+58.3$ |
| *Conditional add-on: Mapping with Exact Evidence already on* | | | | | | | |
| Mapping \| Evidence | BlogWorld | 15 | $3/15 \to 4/15$ | 20.0 | 26.7 | $+6.7$ | $-41.8$ |
| | LifeWorld | 15 | $11/15 \to 12/15$ | 73.3 | 80.0 | $+6.7$ | $-26.1$ |
| | ShopWorld | 15 | $1/15 \to 0/15$ | 6.7 | 0.0 | $-6.7$ | $-4.9$ |
| | $\tau$ adaptation | 15 | $2/15 \to 1/15$ | 13.3 | 6.7 | $-6.7$ | $-16.6$ |

History and Goal use 15 matched task–model pairs per software. Evidence and Mapping are marginal contrasts over the other factor in the shared $2 \times 2$ panel (29 BlogWorld pairs and 30 elsewhere). The final block is the E-only→E+Mapping add-on contrast with 15 pairs; it is not a fifth independent factor. Bold token values mark the only factor with a favorable cost direction in all four settings. $\tau$ is a direct-instruction retail adaptation rather than the native simulated-user protocol; ShopWorld has a pronounced strict-Oracle floor.

**Result and interpretation.** LifeWorld provides the clearest cross-software History effect (0/15 to 6/15, $+40$ pp). Structured Goal is positive in LifeWorld (7/15 to 10/15) and the $\tau$ adaptation (0/15 to 3/15), while BlogWorld and ShopWorld remain unchanged under floor-limited strict outcomes.

Exact Evidence has only modest, heterogeneous success gains, but lowers mean input tokens in every environment, from 1.7k saved per ShopWorld pair to 160.8k in $\tau$. In a separate three-template short-suffix diagnostic, fixing Mapping and adding Evidence reduces source-action rereads from 26 to zero and the aggregate model decisions from 24/25/25 to 8/8/8 across the three Mapping levels. Generic Mapping itself has no stable completion benefit. Indeed, adding Mapping with Evidence fixed on produces 17/60 versus 17/60 strict successes overall and increases input in every environment. Thus the portable result is not that every channel helps: task opportunities and software protocol determine success effects, whereas retaining an already observed exact value more consistently reduces reacquisition cost. Full per-model cells, the short-suffix diagnostic, and floor analyses are in Appendix D.

### 4.3 How should information be retained and recovered?

#### 4.3.1 The quality–cost tension of complete History

The MailWorld panel first establishes the problem that motivates selective recovery. Bounded History reaches 26/40 terminal successes using 12.75M input tokens; replaying the Full event history reaches 38/40 but uses 42.44M input tokens, or 3.33× as much. Full replay therefore buys twelve additional completions at a large context cost. Complementary layer studies show that History is not a single atomic channel: recent actions, important events, navigation/readback, and latest feedback support different recovery behaviors. Their completed small-sample results and the ongoing balanced replication are separated in Appendix C.4.

#### 4.3.2 Targeted recovery: P3B

The P3B study is the targeted How validation. Four official $\tau$-retail training tasks share a controlled acquisition prefix and then run a suffix under Bounded, Targeted, or Full History. Each policy contains eight logical trials across GPT-5.6-sol and DeepSeek V4 Pro.

Table 3: **P3B delayed-evidence reuse.** Each policy aggregates eight logical trials. Targeted and Full reach the same observed first-exact-binding and correct-mutation counts, while final reward and input cost remain distinct.

| History policy | First exact binding | Correct mutation | Official reward = 1 | Input tokens |
|---|---|---|---|---|
| Bounded | 2/8 | 5/8 | 2/8 | 1,303,340 |
| **Targeted** | **8/8** | **8/8** | 5/8 | **536,774** |
| Full | **8/8** | **8/8** | **6/8** | 2,314,279 |

**Result and interpretation.** Targeted improves first exact binding from 2/8 under Bounded to 8/8 and correct mutation from 5/8 to 8/8. It reaches the same observed intermediate counts as Full while using 536,774 rather than 2,314,279 input tokens, a 76.8% reduction; it also uses 58.8% fewer tokens than Bounded. The mechanism is explicit: restore records ranked for the current task and decision instead of replaying all earlier events or trusting a short recency window. The endpoint distinction is essential: Targeted does not match Full on official reward (5/8 versus 6/8), so this experiment supports efficient binding recovery, not universal end-to-end equivalence. The tasks are training tasks and the shared prefix isolates later evidence use rather than unconstrained acquisition.

### 4.4 When should information be available?

The LifeWorld V12 timing study directly varies when the same evidence card is exposed on ten opportunity-enriched tasks using one DeepSeek V4.1 Flash route. The conditions are categorical policies rather than points on a continuous time axis, so Table 4 is the primary presentation.

Table 4: **Information timing in LifeWorld V12.** Counts are out of ten episodes per policy. The study is a narrow, reused-development-bank mechanism probe; it is not an independent multi-software confirmation.

| Policy | First correct binding | Strict completion | Card exposures |
|---|---|---|---|
| Never | 0/10 | 0/10 | 0 |
| Always | **9/10** | **9/10** | 63 |
| On-demand | 8/10 | 8/10 | 21 |
| JIT | 8/10 | 8/10 | 23 |
| Late | 0/10 | 2/10 | 14 |

**Result and interpretation.** The cleanest contrast is before versus after the critical operation. Always and JIT obtain 9/10 and 8/10 correct first bindings, whereas Late obtains 0/10 even though it exposes fourteen cards. Once the first binding opportunity has passed, later disclosure cannot repair that decision. This validates pre-decision availability, not a claim that JIT is optimal: Always is numerically highest, and among seven tasks jointly solved by Always and JIT, JIT input is only about 0.4% lower. That jointly successful comparison is descriptive rather than an all-assigned treatment effect. Complete timing lineage and presentation sensitivities appear in Appendix F.

**Long-horizon boundary.** MailWorld super-complex tasks, the four-software panel, P3B, and V12 all involve delayed information use, but the environments do not share a common task-length scale. We therefore treat long-horizon behavior as a limited boundary analysis, not a headline contribution or a pooled complexity trend; Appendix A inventories the completed studies and Appendix B reports the held-out mechanism audit.

## 5 Conclusion

Information carries distinct decision roles: a fact must be acquired, retained, bound to its object, and used to build arguments. ACI-Core makes these roles experimentally controllable. In the MailWorld cohorts, History and Structured Goal produce the largest clean completion effects; across the four-software panel, Exact Evidence has the most consistent input-cost direction even though completion effects remain heterogeneous. Targeted recovery further restores delayed first binding at substantially lower input cost than Full History, while the timing study establishes the narrower boundary that information arriving after a critical operation cannot support that earlier decision.

## 6 Limitations

The ten-channel decomposition is evaluated most completely in MailWorld. The four-software panel covers only History, Structured Goal, Exact Evidence, and Generic Mapping, and its $\tau$ setting is a direct-instruction adaptation rather than the native simulated-user benchmark. P3B uses four official retail training tasks after a controlled acquisition prefix, while the timing panel uses ten opportunity-enriched tasks and one model route. The two MailWorld cohorts ran separately and their controls are never pooled; opportunity-enriched pilots diagnose mechanisms without estimating population-average effects. Provider behavior can also change behind a stable route label. Finally, every result is measured under an enforcing Runtime, so information that is redundant under these checks may be necessary under a weaker execution layer.

## Reproducibility and AI-use statement

The research record retains frozen configurations, product commits, selected and invalid physical attempts, inputs sent to the model, driver traces, initial and final state snapshots, and independent Oracle outputs. The accompanying internal evidence map identifies the audits behind every number in this draft. Before submission, release artifacts require a stable benchmark package, documented splits by task family, and independent outcome checks. Generative AI assisted research ideation, code and trace analysis, and manuscript drafting; the authors are responsible for verifying every claim and citation.

## REFERENCES


Saaket Agashe, Jiuzhou Han, Shuyu Gan, Jiachen Yang, Ang Li, and Xin Wang. Agent s: An open agentic framework that uses computers like a human. In *International Conference on Learning Representations*, volume 2025, pp. 22924–22946, 2025.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik R Narasimhan. $\tau^2$-bench: Evaluating conversational agents in a dual-control environment. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=OC2z7iSQKa.

Hao Chen, Zhexin Hu, Jiajun Chai, Haocheng Yang, Hang He, Xiaohan Wang, Wei Lin, Luhang Wang, Guojun Yin, et al. Toolforge: A data synthesis pipeline for multi-hop search without real-world apis. *arXiv preprint arXiv:2512.16149*, 2025.

Zihao Cheng, Weixin Wang, Yu Zhao, Ziyang Ren, Jiaxuan Chen, Ruiyang Xu, Shuai Huang, Yang Chen, Guowei Li, Mengshi Wang, Yi Xie, Ren Zhu, Zeren Jiang, Keda Lu, Yihong Li, Xiaoliang Wang, Liwei Liu, and Cam-Tu Nguyen. Lifebench: A benchmark for long-horizon multi-source memory, 2026. URL https://arxiv.org/abs/2603.03781.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, David Vazquez, Nicolas Chapados, and Alexandre Lacoste. Workarena: How capable are web agents at solving common knowledge work tasks? In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=BRfqYrikdo.

Kutluhan Erol, James Hendler, and Dana S Nau. Htn planning: Complexity and expressivity. In *AAAI*, volume 94, pp. 1123–1128, 1994.

Xinyi Fan, Miri Liu, Ruozhen Yang, Siru Ouyang, and Jiawei Han. Can agent memory systems track evolving state?, 2026. URL https://arxiv.org/abs/2608.19652.

Wei Fang, Yang Zhang, Kaizhi Qian, James Glass, and Yada Zhu. Play2prompt: Zero-shot tool instruction optimization for llm agents via tool play. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 26274–26290, 2025.

Richard E Fikes and Nils J Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208, 1971.

David Harel. Statecharts: A visual formalism for complex systems. *Science of computer programming*, 8(3):231–274, 1987.

Hang He, Chuhuai Yue, Chengqi Dong, Mingxue Tian, Zhenfeng Liu, Jiajun Chai, Xiaohan Wang, Yufei Zhang, Qun Liao, Guojun Yin, Wei Lin, Chengcheng Wan, Haiying Sun, and Ting Su. Localsearchbench: Benchmarking agentic search in real-world local life services, 2025. URL https://arxiv.org/abs/2512.07436.

Charles Antony Richard Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576–580, 1969.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in llm agents via incremental multi-turn interactions. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 156259–156291, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/fd1eff9dd295df50a41f2521942fa31d-Paper-Conference.pdf.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Leslie Lamport. The temporal logic of actions. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 16(3):872–923, 1994.

Junlong Li, Wenshuo Zhao, Jian Zhao, Weihao Zeng, Haoze Wu, Xiaochen Wang, Rui Ge, Yuxuan Cao, Yuzhen Huang, Wei Liu, Junteng LIU, Zhaochen Su, Yiyang Guo, FAN ZHOU, Lueyang Zhang, Juan Michelini, Xingyao Wang, Xiang Yue, Shuyan Zhou, Graham Neubig, and Junxian He. The tool decathlon: Benchmarking language agents for diverse, realistic, and long-horizon task execution. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 53359–53394, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/5783212d85c205ef823b8974d44872c5-Paper-Conference.pdf.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. API-bank: A comprehensive benchmark for tool-augmented LLMs. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3102–3116, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.187. URL https://aclanthology.org/2023.emnlp-main.187/.

Mo Li, L.H. Xu, Qitai Tan, Long Ma, Hongyong Song, Ting Cao, and Yunxin Liu. Sculptor: Empowering llms with cognitive agency via active context management. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 153411–153440, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/f9155a06ba1b20e40c38bf8541db456f-Paper-Conference.pdf.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating llms as agents. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 52989–53046, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/e9df36b21ff4ee211a8b71ee8b7e9f57-Paper-Conference.pdf.

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/.

Xuan Lu, Haohang Huang, Rui Meng, Yaohui Jin, Wenjun Zeng, and Xiaoyu Shen. Tools are under-documented: Simple document expansion boosts tool retrieval. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 141631–141653, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/e54a9889f1048526365ed8fb56c8c8cd-Paper-Conference.pdf.

You Lu, Kun Zhang, Bihuan Chen, and Xin Peng. Docschisel: Adaptive tool documentation optimization framework for llm agents, 2026b. URL https://arxiv.org/abs/2608.10037.

Nancy A Lynch and Mark R Tuttle. Hierarchical correctness proofs for distributed algorithms. In *Proceedings of the sixth annual ACM Symposium on Principles of distributed computing*, pp. 137–151, 1987.

Yasmine Omri, Ziyu Gan, Zachary Broveak, Robin Geens, Zexue He, Alex Pentland, Marian Verhelst, Tsachy Weissman, and Thierry Tambe. Agent memory: Characterization and system implications of stateful long-horizon workloads, 2026. URL https://arxiv.org/abs/2606.06448.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, dahai li, Zhiyuan Liu, and Maosong Sun. Toolllm: Facilitating large language models to master 16000+ real-world apis. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun

(eds.), *International Conference on Learning Representations*, volume 2024, pp. 9695–9717, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/28e50ee5b72e90b50e7196fde8ea260e-Paper-Conference.pdf.

Chris Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, Daniel Toyama, Robert Berry, Divya Tyamagundlu, Timothy Lillicrap, and Oriana Riva. Androidworld: A dynamic benchmarking environment for autonomous agents. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 406–441, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/01a83bc2f2732a58e6aa731e659e7101-Paper-Conference.pdf.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551. Curran Associates, Inc., 2023. doi: 10.52202/075280-2997. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/d842425e4bf79ba039352da0f658a906-Paper-Conference.pdf.

Jiaqi Shao, Yufeng Miao, Wei Zhang, and Bing Luo. Foldact: Efficient and stable context folding for long-horizon search agents. *arXiv preprint arXiv:2512.22733*, 2025.

Yaorui SHI, Yuxin Chen, Siyuan Wang, Sihang Li, Hengxing Cai, Qi GU, Xiang Wang, and An Zhang. Look back to reason forward: Revisitable memory for long-context llm agents. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 78055–78082, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/7e0dc9ccba0f1333be13a3f9dc2b3138-Paper-Conference.pdf.

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/.

Guangya Wan, Mingyang Ling, Xiaoqi Ren, Rujun Han, Sheng Li, and Zizhao Zhang. COMPASS: Enhancing agent long-horizon reasoning with evolving context. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3360–3380, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.152. URL https://aclanthology.org/2026.acl-long.152/.

Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. Mint: Evaluating llms in multi-turn interaction with tools and language feedback. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 32593–32627, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/8a0d3ae989a382ce6e50312bc35bf7e1-Paper-Conference.pdf.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory, 2024b. URL https://arxiv.org/abs/2409.07429.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 52040–52094. Curran Associates, Inc., 2024. doi: 10.52202/079017-1650.

URL https://proceedings.neurips.cc/paper_files/paper/2024/file/5d413e48f84dc61244b6be550f1cd8f5-Paper-Datasets_and_Benchmarks_Track.pdf.

Kewei Xu, Xiaoben Lu, Shuofei Qiao, Zihan Ding, Haoming Xu, Lei Liang, and Ningyu Zhang. Longds-bench: On the failure of long-horizon agentic data analysis, 2026. URL https://arxiv.org/abs/2605.30434.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z Pan, et al. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12805–12825, 2026.

John Yang, Carlos Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 50528–50652. Curran Associates, Inc., 2024. doi: 10.52202/079017-1601. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/5a7c947568c1b1328ccc5230172e1e7c-Paper-Conference.pdf.

Yuhao Yang, Tianyu Fan, and Chao Huang. Cli-anything: Towards agent-native computer use. *arXiv preprint arXiv:2606.03854*, 2026.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 20744–20757. Curran Associates, Inc., 2022. doi: 10.52202/068431-1508. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/82ad13ec01f9fe44c01cb91814fd7b8c-Paper-Conference.pdf.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations*, 2023. URL https://mlanthology.org/iclr/2023/yao2023iclr-react/.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik R Narasimhan. {$\tau$}-bench: A benchmark for \underline{T}ool-\underline{A}gent-\underline{U}ser interaction in real-world domains. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=roNSXZpUDN.

Lu Yi, Runlin Lei, Liuyi Yao, Yuexiang Xie, Yuyang Li, Wenhao Zhang, Zhewei Wei, Yaliang Li, and Jian-Yun Nie. Learning agent-compatible context management for long-horizon tasks, 2026. URL https://arxiv.org/abs/2605.30785.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 39458–39486, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/4264ee4376776907c0b87ed70b959585-Paper-Conference.pdf.

Yuxiang Zhang, Jiangming Shu, Ye Ma, Xueyuan Lin, Shangxi Wu, and Jitao Sang. Memory as action: Autonomous context curation for long-horizon agentic tasks. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 19149–19164, 2026.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*, 2024.

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, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/4410c0711e9154a7a2d26f9b3816d1ef-Paper-Conference.pdf.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Bryan Kian Hsiang Low, and Paul Liang. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents. In *International Conference on Learning Representations*, volume 2026, pp. 58413–58438, 2026.

Yu Zhuang, Kefei Chen, Yitong Duan, Shuxin Zheng, Jian Li, and Xu-Yao Zhang. Agentrewind: Recoverable execution for long-horizon llm agents, 2026. URL https://arxiv.org/abs/2608.14380.

## A LONG-HORIZON AND SUPER-COMPLEX TASK COVERAGE

The studies relevant to long-horizon operation are more extensive than a single delayed-reuse pilot, but they do not form one common complexity scale across every backend. Tables 5 and 6 summarize what each completed panel can support. P3B is cross-referenced from Sections 4.3 and 4.4 because it tests delayed recovery before a later binding; its 24 identities are counted once, under the primary retention/recovery comparison.

Table 5: **Completed long-horizon and complexity-oriented evidence (I).** Counts are logical task–model–condition identities unless otherwise noted; overlapping task templates are not independent tasks.

| Environment | Coverage and observed pattern | What it does not establish |
|---|---|---|
| MailWorld | Difficulty panel: 20 Simple, 20 Complex, and 20 Super-complex tasks, two model routes, and Raw/Neutral conditions (240 selected identities). For DeepSeek, Runner-normal Neutral/Raw counts across tiers are 20/20 vs. 20/20, 20/20 vs. 16/20, and 9/20 vs. 9/20; Oracle counts are 20/20 vs. 20/20, 20/20 vs. 16/20, and 9/20 vs. 10/20. The super-complex slice is bidirectional. | Difficulty tiers do not establish a monotonic information effect. Mechanism panels reuse or overlap task structures rather than providing hundreds of independent templates; thirteen GPT identities in the Raw/Neutral audit use an author-accepted configuration-equivalence view. |
| LifeWorld | Three tiers with 10 Simple, 10 Complex, and 10 Ultra tasks; four presentation conditions and three model routes; 359/360 planned identities have valid terminal outcomes. No single presentation policy wins across routes: C is best for DeepSeek Flash, A for GPT-5.5, and A/B/C are nearly tied on complete GPT-5.6-sol blocks. | Each task–condition cell has one rollout. This does not establish a universal Phase/presentation benefit or a randomized performance curve over continuous task length. |
| Four-software panel | BlogWorld, ShopWorld, LifeWorld, and direct-instruction $\tau$-retail adaptation; five difficult multi-step tasks per environment, three model routes, and eight information conditions (479/480 valid identities). Tasks require delayed exact-value reuse, intervening obligations, and final constraints. History and Goal have positive cells in some environments; Exact Evidence reduces mean input tokens in all four. | This is not a matched Simple/Complex/Ultra gradient. Strict outcomes have floor conditions, especially ShopWorld (1/120 strict successes; 39/120 in a post-hoc business-state diagnostic that omits trajectory requirements); the diagnostic does not replace the strict Oracle. |

Table 6: **Completed long-horizon and complexity-oriented evidence (II).** These cohorts are described separately and are not summed as independent tasks.

| Environment | Coverage and observed pattern | What it does not establish |
|---|---|---|
| BlogWorld and ShopWorld | BlogWorld has an earlier 13-task single-model S0–S6 screen and a five-task Advanced V2 panel. All five ShopWorld V2 tasks are Very Advanced and cover cross-object, multi-step mutations. These show that long action chains and final-state constraints were exercised beyond MailWorld. | BlogWorld's early screen is near ceiling; V2 covers one difficulty band. ShopWorld's strict Oracle floor prevents a strong estimate of information-factor effects. |
| $\tau$-bench | P3B uses four official retail training tasks, two model routes, and three History policies (24 identities) after a shared acquisition prefix: first bindings are 2/8, 8/8, and 8/8 for Bounded, Targeted, and Full History. A separate native real-ACI panel uses 30 task blocks, one DeepSeek route, and Raw/A/B/C; reward-1 counts are 22/30, 3/30, 0/30, and 1/30. | P3B uses training tasks and does not match final reward. The native panel has 73 protocol/termination nulls and is a stress/compatibility boundary, not a clean information-factor estimate. These two $\tau$ studies are not pooled. |
| AppWorld | Multi-app integration and development tasks exist, but the planned Simple/Complex/Ultra classification and matched long-horizon panel remain on hold. | This is integration/development work, not a completed systematic complexity study; no formal AppWorld long-horizon generalization claim is supported. |

The long-horizon claim we make is therefore descriptive and bounded: several completed panels exercise complex, multi-step, delayed-reuse software tasks, and P3B directly tests restoration of a fact acquired before a controlled interval. We do not claim a common cross-software complexity gradient, a monotonic effect of complexity, or completed systematic coverage of AppWorld.

## B MECHANISM-STAGE AUDIT ON HELD-OUT TASKS

Table 7: **Mechanism-stage audit on eight valid held-out MailWorld trials.** Four tasks are paired under bounded S6 History and History hidden, all using DeepSeek V4.1 Flash. NA denotes an undefined stage rather than a failure: History-hidden runs never produced a body-bearing first consequential mutation, so Availability, Bind, and Use have no scoring denominator.

| Stage or diagnostic | Bounded History | History hidden |
|---|---|---|
| Evidence available before first consequential mutation | 4/4 | NA |
| Any `read_history` uptake | 3/4 (31 calls) | 0/4 (0 calls) |
| Scorable first consequential mutation reached | 4/4 | 0/4 |
| Correct Bind at first scorable mutation | 4/4 | NA |
| Correct Use with a successful Runtime receipt | 4/4 | NA |
| Terminal Oracle full success | 0/4 | 0/4 |
| Mean model decisions | 26.25 | 48.00 |
| Mean input tokens | 449,234 | 531,828 |

This audit separates intermediate reachability from terminal completion. With bounded History, all four episodes reached a scorable first mutation and correctly bound and executed the Oracle-declared target–body pair. With History hidden, none reached that scoring point and all four exhausted the 48-decision budget, so Bind and Use are NA rather than zero. The hidden arm also uses descriptively more decisions and input tokens. However, neither arm passes the independent terminal Oracle (0/4 in both conditions). The result therefore supports only a process-level claim: bounded History enables progress to the correct first mutation in these four paired tasks, but does not establish improved end-to-end completion. `read_history` records Runtime-tool uptake rather than internal attention and is not a required nested stage: one bounded-History trial binds correctly without calling it.

# C COMPLETE SEMANTIC MATRIX AND INTERVENTION BOUNDARIES

Table 8 reports the complete two-cohort matrix rather than selecting only beneficial removals. Each condition has 20 tasks per model. The user instruction remains visible throughout. Different Full rows are separately executed controls; they are not interchangeable.

Table 8: **MailWorld terminal Oracle counts.** Cohort A uses the author-authorized recovery view (318/320 identities within the original cap, 320/320 with two post-cap recoveries). Cohort B has 240/240. Counts do not imply isolated semantic concepts beyond their implemented treatment.

| Cohort | Condition | GPT /20 | DeepSeek /20 |
|---|---|---|---|
| A | Full | 19 | 7 |
| | −Action Semantics | 18 | 9 |
| | −Conditions | 20 | 10 |
| | −History bundle | 3 | 0 |
| | −Dynamic Update disclosure | 17 | 12 |
| | Full-event History | 19 | 19 |
| | D39 references | 18 | 14 |
| B | Full | 16 | 11 |
| | −Types | 15 | 8 |
| | −Identity | 17 | 7 |
| | −Goal | 11 | 3 |
| | −Subgoal | 14 | 8 |
| | −Observations | 14 | 9 |

History-off also disables reminders that depend on History. Subgoal-off can cause progress-update rejection, so it is not a text-only removal with all update behavior unchanged. Conditions-off retains runtime checks.

## C.1 DEVELOPMENTAL S0–S6 SEMANTIC LADDER

S0 begins with action identifiers and a minimal invocation surface; S1 adds schemas and types; S2 adds action semantics; S3 adds conditions and verification definitions; S4 adds Phase organization; S5 adds task identity, Goal, Subgoal, and compact History; and S6 adds dynamic observations, selection effects, and automatic verification feedback. The non-monotonic middle rungs show why the later Full-minus-one studies are needed: cumulative semantic richness does not identify which component causes a change. The later 312-identity disambiguation plan has only 119 verified valid results (99 semantic and 20 feedback) and is excluded from confirmatory effect claims.

## C.2 CONDITIONS: BROAD AVERAGE VERSUS OPPORTUNITY-RICH TASKS

Table 9: **Conditions disclosure and recovery behavior.** The broad ablation and opportunity-enriched pilot use different tasks and denominators. Hiding Conditions never disables Runtime enforcement.

| Condition | Oracle | Decisions | Input tokens | Avoidable precondition failures |
|---|---|---|---|---|
| *Broad MailWorld ablation (40 trials per condition)* | | | | |
| Full control | 26/40 | 19.95 | 318,642 | – |
| Conditions hidden | 30/40 | 19.43 | 303,469 | – |
| *Opportunity-enriched pilot (9 complete pairs)* | | | | |
| Full | 5/9 | 12.33 | 159,820 | 5 |
| Conditions bundle hidden | 3/9 | 12.33 | 151,262 | 19 |

The broad panel is highest with Conditions hidden (30/40 versus 26/40), whereas the enriched pilot favors Full on Oracle completion (5/9 versus 3/9) and records far fewer avoidable precondition failures (5 versus 19). The latter counts are event totals over valid trials, not rates, and the hidden

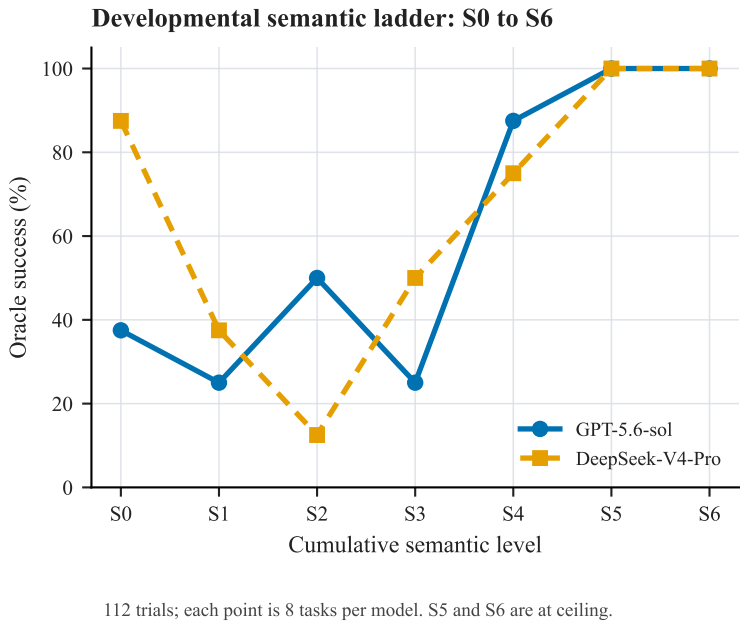


Figure 4: **Developmental semantic ladder.** GPT Oracle counts over eight tasks are 3, 2, 4, 2, 7, 8, and 8 from S0 to S6; DeepSeek counts are 7, 3, 1, 4, 6, 8, and 8. The 112-trial pilot reaches a ceiling at S5/S6 but is non-monotonic before it. Each rung adds a bundle of disclosures and coupled Runtime behavior, so the curve is developmental evidence rather than a single-factor causal estimate.

treatment removes a bundle of condition, verification, and failure-specific disclosure. This contrast supports opportunity-dependent value, not a population-wide positive Conditions effect.

### C.3 Phase treatment boundary

The Phase/History refinement has 49 unique identities with uneven component coverage; Phase has 24 identities (four tasks per cell, two models) and History has 25 (two to five tasks per cell). In the audited Phase subset, Full scores 6/8, description-hidden 7/8, and mechanism-disabled 5/8. These are not a clean ordering: GPT is 4/4, 4/4, 3/4, whereas DeepSeek is 2/4, 3/4, 2/4. Description-hidden retains Phase IDs, Action membership, navigation, and the Runtime gate; mechanism-disabled removes both model-visible Phase information and the Phase gate. This follow-up does not establish that Phase descriptions are unnecessary or that disabling the full mechanism improves completion. The early disclosure-only row that retained the gate is excluded from the main heatmap because it was an interface-mismatch treatment. Figure 5 reports the later three-arm result separately.

### C.4 Exploratory History-layer decomposition

Table 10: **Completed exploratory History-layer results.** Panels use different opportunity-enriched tasks and denominators and therefore must not be ranked as one balanced four-layer ablation. The ongoing balanced replication is not inserted as a measured result.

| Panel | Condition | Oracle success | Input tokens |
|---|---|---|---|
| Complete-layer follow-up | Important events on → off | $4/4 \rightarrow 1/4$ | – |
| | Navigation/readback on → off | $5/6 \rightarrow 3/6$ | – |
| Recent × feedback (DeepSeek) | Both on | 5/5 | 688,173 |
| | Feedback only | 4/5 | 779,442 |
| | Recent actions only | 4/5 | 1,654,623 |
| | Neither | 2/5 | 2,428,999 |

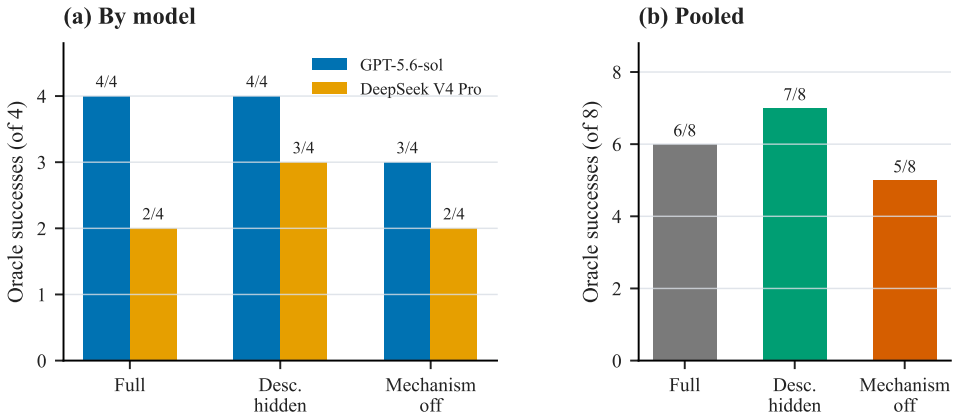


Figure 5: **Phase follow-up with disclosure and mechanism separated.** The later three-arm study reports Full, Phase-description hidden with the Phase ID and Runtime gate retained, and Phase mechanism disabled. Values are the audited Oracle counts (GPT and DeepSeek: four task identities each; pooled: eight). The small sample supports separation of the treatments, not a universal Phase effect.

The complete-layer follow-ups show local losses when important events or navigation/readback are removed. In the five-task DeepSeek factorial, both recent actions and latest feedback yield the highest completion (5/5) and lowest input total, while removing both yields the lowest completion (2/5) and highest cost. The intermediate cells are symmetric in success (4/5) but not cost, suggesting that recent actions can reduce compensatory context use even when the small success count is unchanged. These are mechanism diagnostics: the corresponding GPT half is missing, the layer studies use different cohorts, and removing a named channel does not prove that no duplicate fact survives elsewhere.

### C.5 MailWorld Evidence-by-Mapping factorial

The completed MailWorld factorial crosses exact Evidence and Mapping on 20 opportunity-selected tasks per model. It is reported here in full because its favorable combined cell did not generalize to the later cross-software incremental comparison. Counts below are terminal Oracle outcomes; the combined GPT cell has 20 Oracle successes but 19 clean Runner completions. The observed cells alone do not establish an interaction.

Table 11: **MailWorld Evidence×Mapping factorial.** Oracle successes by model; each condition has 20 tasks.

| Condition | GPT /20 | DeepSeek /20 | Total /40 |
|---|---|---|---|
| Neither | 15 | 9 | 24 |
| Mapping only | 16 | 10 | 26 |
| Evidence only | 16 | 12 | 28 |
| Evidence + Mapping | **20** | **14** | **34** |

## D Cross-software design and statistical scope

Four-factor V2 uses five tasks per software, three model routes, and eight conditions: two History conditions, two Goal conditions, and an Evidence×Mapping factorial. History comparisons keep planning reminders disabled in both arms. Goal-off retains the user task. Evidence is restricted to observed sources; Mapping is generic instruction, not an answer edge. The sole missing valid identity is BlogWorld-001/DeepSeek Flash/F0R0. The execution ledger has 480 main attempts and 65 additional replacement attempts; the selected view contains 479 valid logical identities. Attempts

are not independent experimental units. Table 2 reports the complete software-level matrix; this appendix records the protocol and statistical limits without duplicating it.

The full matrix prevents the positive findings in the main text from being read as a claim that all factors help. History has nonzero strict-success gains in BlogWorld, LifeWorld, and the $\tau$ adaptation, with the clearest difference in LifeWorld; ShopWorld has a strict floor. Goal's positive cells are LifeWorld and the $\tau$ adaptation. Evidence has a favorable input-cost direction in all four settings, but small success differences. Mapping has no consistent strict-success gain and usually increases input. Evidence's marginal effect is not the same as Evidence-only: at Mapping-off, Evidence lowers input in BlogWorld, LifeWorld, and $\tau$, but increases it by 26.8k tokens per ShopWorld pair.

The full per-software, per-model effect registry reports paired improvements and deteriorations, intervals, and exploratory exact tests. Those intervals are based on few reused task structures and the tests are uncorrected for multiple comparisons; they are descriptive, not confirmatory population-level claims. A strict floor can conceal state changes: ShopWorld has 1/120 strict successes but 39/120 under a post-hoc business-state diagnostic that omits major trajectory requirements. We retain the strict task definition; the diagnostic does not replace the primary outcome. LifeWorld's Goal gain is concentrated in Flash, while Pro and GPT are unchanged.

### D.1 Targeted History on adapted tau-retail tasks

The targeted-history study controls a shared execution prefix on four official retail training tasks and compares bounded, targeted, and full History for GPT-5.6-sol (Modcon) and DeepSeek V4 Pro (official endpoint). It contains eight task/model blocks per policy, is opportunity-selected, and is not the native simulated-user leaderboard. Targeted History matches full History on first critical binding, while its final reward remains one success below full History.

Table 12: **Targeted-history results on adapted $\tau$-retail tasks.** Each policy has eight task/model blocks.

| Policy | First binding | Final reward | Input tokens |
|---|---|---|---|
| Bounded History | 2/8 | 2/8 | 1,303,340 |
| Targeted History | 8/8 | 5/8 | 536,774 |
| Full History | 8/8 | 6/8 | 2,314,279 |

Targeted History uses 58.8% fewer input tokens than bounded History and 76.8% fewer than full History. Its first-binding gain versus bounded History is 6/8, while target-versus-full has equal observed first-binding counts; neither should be read as formal equivalence. The four tasks are training tasks, and the controlled-prefix results measure evidence use after common acquisition rather than unconstrained end-to-end search. The 24 selected trials all use attempt-02 after an attempt-01 import-path failure that occurred before any Provider or backend construction.

Table 13: **Mapping's incremental effect with Evidence fixed on.** Each software has 15 valid task/model pairs. Costs are mean paired changes over all outcomes, not only successes. Strict and business-state-only judgments are kept separate elsewhere.

| Environment | E-only $\rightarrow$ E+R strict | Input change | Decision change |
|---|---|---|---|
| BlogWorld | $3 \rightarrow 4$ /15 | $+41,804.3$ | $+2.27$ |
| LifeWorld | $11 \rightarrow 12$ /15 | $+26,096.4$ | $+1.53$ |
| ShopWorld | $1 \rightarrow 0$ /15 | $+4,855.3$ | $+0.20$ |
| $\tau$ adaptation | $2 \rightarrow 1$ /15 | $+16,599.4$ | $+0.27$ |

Across these 60 pairs there are five improvements and five deteriorations. The observed balance does not prove a zero population effect. A strict floor is particularly severe in ShopWorld, where order and exact-once constraints are part of the task. Post-hoc state-only totals are Blog 18/119, Life 81/120, Shop 39/120, and $\tau$ 8/120; corresponding strict totals are 12/119, 67/120, 1/120, and 8/120. Changing the evaluator after seeing these results cannot turn the diagnostic into a preregistered primary outcome.

Original software-level bootstrap intervals and McNemar tests are exploratory and uncorrected for multiple comparisons. For example, Life History has six improvements, no deteriorations, and an uncorrected exact McNemar $p = 0.03125$. The five underlying task structures are reused across models; an interval treating all blocks as independent does not capture task-family uncertainty. We therefore avoid significance claims in the main cross-software table. A zero-width empirical interval at a floor is not certainty about the true treatment effect.

### D.2 Exact Evidence short-suffix mechanism diagnostic

Table 14: **Evidence and Mapping on three short-suffix templates.** Six trials per cell use one DeepSeek route across BlogWorld, LifeWorld, and ShopWorld low/high-ambiguity variants. Episodes share a fixed acquisition prefix and stop at the first critical mutation.

| Condition | First binding | Mean input tokens | Mean decisions |
|---|---|---|---|
| No Evidence, no Mapping | 6/6 | 47,815.0 | 4.00 |
| No Evidence, old Mapping | 6/6 | 49,272.5 | 4.17 |
| No Evidence, revised Mapping | 6/6 | 50,058.8 | 4.17 |
| Evidence only | 5/6 | 16,664.0 | 1.33 |
| Evidence + old Mapping | 6/6 | 16,801.8 | 1.33 |
| Evidence + revised Mapping | 6/6 | 18,194.3 | 1.33 |

At each fixed Mapping level, Evidence reduces mean input by 63.65–65.90%. Aggregate model decisions fall from 24/25/25 to 8/8/8 and source-action rereads from 26 to zero. This directly supports the reacquisition-cost mechanism, but the endpoint is near ceiling (35/36 correct first bindings), the study uses only three templates and one model route, and the revised Mapping fails its predeclared improvement gate. Evidence-only also selects the wrong source once in the high-ambiguity ShopWorld variant; both Mapping versions correct that single trajectory. These observations do not establish a general Evidence-by-ambiguity interaction or a universal Mapping benefit.

## E Reference construction and source visibility

Both arms of the new D39 pilot receive Evidence plus complete, currently referenceable source records. This matters because an Evidence card alone is not necessarily a valid D39 lookup source. The implementation resolves strings only from visible runtime records; it does not retrieve hidden files or query the Oracle. The tested construction operations are literal copy, source/field discrimination, and ordered concatenation on BlogWorld and LifeWorld. Each arm makes 12 model decisions across six tasks. All six reference calls resolve successfully; one literal concatenation introduces an extra newline.

The pilot is deliberately small and has a near-ceiling literal baseline. Its output-token result is stronger than its accuracy evidence, but even the cost conclusion must be component-specific: 3,696 to 2,488 output tokens, 179,556 to 190,610 input tokens, and 183,252 to 193,098 total tokens. No uniform dollar-cost estimate is available. GPT replication and a separate Evidence$\times$D39 factorial on unused tasks remain future work; the current two-arm pilot cannot estimate that interaction.

## F Presentation, Phase, and recent development sensitivity

Successive timing versions reuse task structures and are not independent confirmations. In the 50-episode Flash development comparison, Never/Always/On-demand/JIT/Late first-binding counts are 0,9,8,8,0 out of ten; strict counts are 0,9,8,8,2. Availability before use is supported more directly than the superiority of a particular early policy.

The recent 20-identity position study puts the same card in its original or leading location. Both have 7/10 strict successes, with one improvement and one deterioration. Aggregate input is 20.2% lower in the leading condition (2,396,446 versus 1,911,903 tokens). Within six jointly strict-successful pairs, input falls 28.0% and decisions 24.1%; this is explicitly outcome-conditioned.

A subsequent 40-identity study crosses flat/Phase organization with original/relation-card presentation. Strict and first-binding counts are 6/10, 6/10, 8/10, and 6/10, respectively. One flat-original identity had an ordinary model parse error followed by a metrics-reporting KeyError; its later replacement also fails. Retaining the original valid failure leaves success counts unchanged but changes cost. In an original-failure sensitivity view, flat-original uses 2,094,626 input tokens and 89 decisions, versus Phase-original's 1,149,094 tokens and 132 decisions: input falls 45.1% while decisions increase 48.3%. The existing selected view instead has 2,228,517 flat-original tokens and 95 decisions (48.4% fewer Phase tokens, but 38.9% more decisions). The selected index requires explicit adjudication of this ordinary failure; completion of all identities does not establish compliance with the original replacement rule. This is not a pure Phase-description effect.

The separate preservation-complete D2 follow-up has 20 identities. Original versus complete-card first binding is 5/10 versus 8/10, final state 6/10 versus 9/10, and strict success 5/10 versus 7/10. Only seven pairs expose both arms to their respective cards; three diverge earlier. The all-assigned counts are primary descriptive results, and the exposed-pair diagnostic cannot identify an unbiased treatment effect by itself. Full-panel input increases from 1,135,930 to 1,376,992 tokens (+21.2%), and decisions rise from 125 to 145 (+16.0%). The four jointly strict-successful pairs use 23.0% fewer input tokens, but this conditional saving is dominated by one Blog task; excluding it, input rises 1.96% and decisions are unchanged. Both arms additionally make 22 common private acquisition calls not counted as model decisions or ordinary business Actions. The treatment bundles candidate evidence, relations, and preservation values, rather than isolating preservation alone. Independent tasks and a second model are needed before treating the result as a general preservation principle.

## G FULL-INTERFACE TRANSFER AND PROTOCOL FAILURES

A prior 60-task/model MailWorld comparison obtains Raw/neutral terminal counts of 57/60 versus 59/60 for GPT and 46/60 versus 49/60 for DeepSeek. The small paired differences are not evidence of universal dominance. Thirteen GPT identities use an author-accepted configuration-equivalence view, and provider periods differ. This comparison is kept separate from within-interface disclosure estimates.

Table 15: **Native interactive $\tau$: real-ACI negative transfer.** DeepSeek V4.1 Flash, 30 task blocks. Null is a recorded lack of official terminal reward, not an observed reward-zero. All identities remain in the planned denominator.

| | Raw | A | B | C |
|---|---|---|---|---|
| Reward 1 | 22 | 3 | 0 | 1 |
| Reward 0 | 7 | 5 | 4 | 4 |
| Reward null | 1 | 22 | 26 | 25 |

Across ACI configurations, the 73 nulls consist of 26 illegal top-level response operations, 21 decision-budget terminations, and 26 driver finishes before the official conversation ended. The legal response path is an execute operation targeting the response Action. These are ordinary system failures, not grounds for network replacement. The cohort contains 29 frozen test tasks and one public training task, so it is not an unqualified held-out benchmark estimate. An earlier native-router study is a different product and cannot substitute for these real-ACI results.

**Protocol-repair compatibility check.** A newly completed, separately frozen check uses three official training tasks (1, 3, 4) with Raw/A/B/C on the official DeepSeek Flash route. All 12 selected runs obtain reward one, official termination, and correct mutations; three network-invalid attempts are retained separately. Input totals for Raw/A/B/C are 158,054/331,600/339,481/551,617, with 32/28/29/53 decisions. This demonstrates successful compatibility on those three tasks, not general superiority or a matched before/after estimate against the previous 30 tasks. Raw retains occurred actions and schema hashes but lacks exact per-call prompt/response envelopes; this artifact limitation is disclosed, not repaired by rerunning successful trials. Independent stress evaluation and a systematic AppWorld complexity panel remain incomplete and are not presented as completed evidence.

## H MODEL ROUTES, TASK SPLITS, AND BUDGETS

The cross-software panel uses GPT-5.6-sol on Modcon and DeepSeek V4 Pro/V4.1 Flash on the official DeepSeek endpoint. Temperature is zero and no sampling seed is fixed. GPT reasoning is medium for the three authored backends, but not explicitly set in the direct-instruction $\tau$ runner. DeepSeek thinking is disabled. Authored backends use one-of tool schemas with a per-response output cap of 1,800 tokens; the $\tau$ adaptation uses flat schemas and 4,096. Named tool choice is amended to required in 56 authored-backend GPT replacement configurations. The $\tau$ runner already uses required for GPT. These are disclosed transport differences, not identical upstream model snapshots.

Table 16: **Principal budget and split distinctions.** Decision limits apply per model-driven suffix in controlled-prefix studies. Other runtime and cumulative caps remain software-specific. A dash denotes no independently specified cap in the runner, not a zero budget.

| Study | Decisions | Output/call | Task source |
|---|---|---|---|
| Four-factor Blog | 55 | 1,800 | Five study-authored tasks |
| Four-factor Life | 65 | 1,800 | Five study-authored tasks |
| Four-factor Shop | 55 | 1,800 | Five study-authored tasks |
| Four-factor $\tau$ | 60 | 4,096 | Retail train 197,245,248,305,331 |
| Targeted-history $\tau$ | 20 | 2,048 | Retail train 079,392,466,155 |
| Mapping screen | 8 | 4,096 | Three new templates, paired ambiguity states |
| D39 pilot | 16 | 4,096 | Blog/Life, copy/field/concat |

For four-factor Blog/Life/Shop, cumulative input caps are 1.4M/1.6M/1.4M, cumulative output caps 55k/65k/55k, and wall budgets 3,000/3,600/3,000 seconds. The $\tau$ adaptation has a 3,600-second provider deadline and no separate cumulative token cap. The targeted-history study compares GPT and DeepSeek Pro on four official training tasks after a controlled prefix; it is an opportunity-selected evaluation, not the official held-out leaderboard. Its GPT reasoning level is also unspecified. All selected attempt-02 runs follow an attempt-01 import-path failure before model or backend construction.

The latest 48-identity screen uses only DeepSeek Pro, named one-of tool calls, a 90-second request timeout, and two transport retries. It adds frozen runner/fixture overlays to the existing products: the baseline Git revision alone is insufficient for reproduction. Neither its task templates nor the older development iterations should be relabeled as unseen confirmation data. Exact source and assignment hashes, cumulative per-software caps, and selected-manifest locations are supplied in the accompanying protocol registry.