MEMORY HAS AN INTERFACE: FROM FACTS TO ACCEPTED ACTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

The same remembered facts can lead to different task outcomes depending on how they enter an agent's context and how its responses reach execution. We study this interface by removing reference-action facts from user requests and restoring byte-identical marked JSON either inline or through a historical assistant–tool exchange. Across 146 selected tasks and three models at one primary seed, historical delivery achieves 29.68% success versus 23.52% for inline delivery, a gain of 6.16 percentage points. A separate 59-task retail study with quantized Qwen3.6-27B reveals a distinction between proposing an action and having it accepted: all 25 first-response pairs with different native acceptance outcomes propose identical tool names and arguments. We then cross the delivery interface with the execution rule in 236 complete interactions. Native rules yield 1/59 inline and 6/59 historical successes; admitting schema-valid text-plus-tool responses to execution raises both totals to 56/59. Content controls further establish the value of correct facts through both interfaces, with model-dependent delivery contrasts. Together, these measurements identify message construction and response acceptance as consequential stages between selected facts and completed tasks, and provide a concrete framework for evaluating how agent memory becomes action.

1 INTRODUCTION

A remembered reservation identifier becomes useful when an agent turns it into an accepted action. Between the supplied fact and the final task state lie a generated response, a communication check, and a tool execution. We study this path by holding selected factual bytes fixed and changing the *memory delivery interface* that places them in the acting model's context.

Memory systems organize what an agent stores and retrieves (Packer et al., 2023; Park et al., 2023; Zhong et al., 2024), and memory benchmarks measure how it uses earlier information (Wu et al., 2025). A selected record still arrives through a role, position, delimiter, and history structure. Work on role labels and context layouts demonstrates their influence on model behavior (Chen et al., 2026; Peleg Pelc et al., 2026). Our question follows a fixed correct record to a stateful task endpoint, where the execution protocol also acts on the model's response.

We construct a parameter-recovery setting: a deterministic selector extracts short scalar facts from a reference action and removes their matching occurrences from the user request. Both focal conditions restore identical canonical JSON. INLINE appends it to the request; HISTORY supplies it through a frozen historical `memory.read` exchange. The task, model, ordinary callable tools, decoding, and evaluator stay fixed within each comparison. The treatment is the complete message construction, including roles, ordering, boundaries, and historical tool metadata.

Historical delivery scores 29.68% versus 23.52% inline across 146 selected tasks and three models, a paired increase of $6.16\,\mathrm{pp}$ (Figure 1). The τ airline and retail tasks supply 22 of the 27 net additional successes, motivating a closer look at their execution path.

A separate study on the 59 retail tasks records this path with quantized Qwen3.6-27B. In 118 first responses, every pair with different native acceptance outcomes proposes the same tool name and arguments; accompanying text changes acceptance. Crossing the interfaces with native checks and a rule admitting schema-valid mixed responses produces 236 complete interactions. Success rises

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

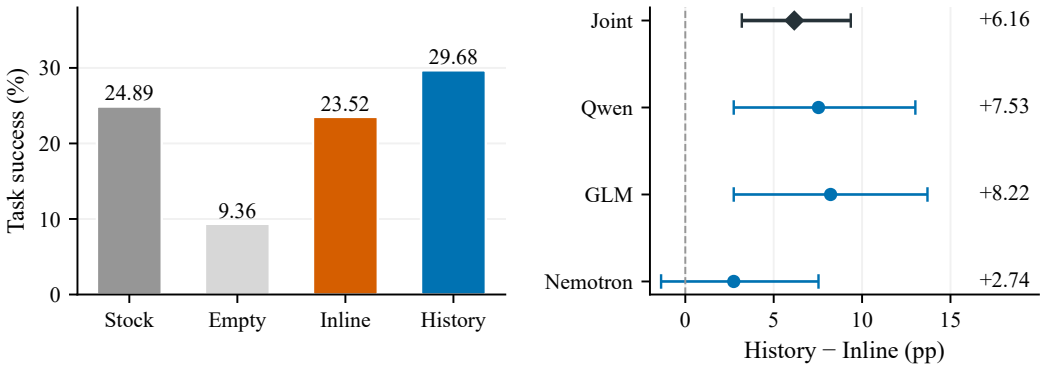


Figure 1: Matched facts at the task endpoint. Left: success on 146 tasks and three models, with one primary request seed (1,752 trajectories). Inline and History carry identical marked JSON. Right: paired historical-minus-inline differences. The joint bar spans fixed-panel resampling percentiles 2.5–97.5; model bars show descriptive paired-bootstrap 95% intervals. The τ panel supplies 22 of the 27 net additional successes.

from 1/59 to 56/59 inline and from 6/59 to 56/59 historically. Section 6 gives the paired contrasts and uncertainty; complementary content controls measure fact value through both interfaces.

The paper makes three contributions:

1. A payload-matched design measures a 6.16 pp delivery difference at the native task endpoint on the primary panel.
2. Paired response records separate proposed actions from communication acceptance: all 25 acceptance-discordant pairs propose identical tool names and arguments.
3. Complete interactions measure interface performance under two explicit routing rules, linking accepted responses to tool execution and final task scores.

The evaluation unit joins the selected record, message constructor, acting model, and execution protocol.

2 Related Work

Memory systems: selecting information for future use. Agent memory systems provide mechanisms for organizing information beyond the immediate interaction. MemGPT uses a memory hierarchy and explicit context management to operate within a limited context window (Packer et al., 2023). Generative Agents combine an observation stream with retrieval and reflection to support temporally coherent behavior (Park et al., 2023). MemoryBank develops long-term conversational memory and updates it over interactions (Zhong et al., 2024). These systems make decisions about which information an agent can later access. Our experimental starting point is the next interface in this chain: a factual record has been selected, and a context constructor makes it available to the acting model. Keeping that record fixed yields a direct comparison of delivery choices. The design complements end-to-end memory evaluation by measuring a variable that lies between record selection and task execution.

Memory benchmarks: measuring the use of prior information. LongMemEval evaluates long-term interactive memory through information extraction, multi-session reasoning, temporal reasoning, knowledge updates, and abstention (Wu et al., 2025). Such evaluation asks whether an assistant can recover and use the right information from a history. Our controlled parameter-recovery setting uses short facts selected from reference actions and makes their delivery the experimental factor. This arrangement makes the information in the two compared contexts byte-identical and scores the resulting task execution. The distinction between selected content and its interface supplies a

useful experimental vocabulary: the selected record describes what information is available, while the message constructor describes how it reaches the agent. Our factorial experiment uses the same vocabulary to cross content correctness with delivery structure.

Prompt structure and agentic context construction. Language models use long-context evidence differently depending on its position (Liu et al., 2024), and meaning-preserving prompt-format changes can substantially affect accuracy (Sclar et al., 2024). ReAct interleaves thoughts, actions, and observations in a growing task trajectory (Yao et al., 2023), making trajectory structure part of the agent's context. Recent studies examine this structure directly. Chen et al. (2026) hold an erroneous claim byte-identical across user, tool, and system-memory wrappers and distinguish label-free boundaries from role labels when measuring explicit error flagging. Peleg Pelc et al. (2026) introduce a language for specifying agentic contexts and evaluate seven history and message layouts on MINT reasoning and coding tasks, with task-dependent performance differences. These studies establish that role and context construction can affect behavior and task performance. Our parameter-recovery design makes selected correct factual bytes the matched object and follows their use to native stateful task outcomes. Crossing content correctness with delivery and decomposing results by model measures when the selected information changes execution under each specified construction.

Executable tool-agent evaluation. AgentDojo supplies executable tasks and environments together with native utility evaluation (Debenedetti et al., 2024). τ-bench studies interactions among an agent, a user, and domain tools (Yao et al., 2025). ToolSandbox provides stateful conversational scenarios with milestone and constraint-based evaluation (Lu et al., 2025). These environments connect a model's messages to changes in an executable system, giving the delivery comparison a concrete outcome. We construct a fixed task panel from their reference-action arguments, retain all selected strata, and specify the exact deterministic evaluator components used for each. AgentDojo also supports the larger context-state and content-quality studies. Using paired task/model comparisons and retaining whole tasks as statistical clusters keeps the interface comparison connected to each task's execution conditions.

3 Measuring the Memory Interface

3.1 Fixing the facts

For task i, let r_i be its original user request, T_i the ordinary callable tools, and g_i the deterministic task evaluator. A selector extracts a scalar dictionary z_i from the first reference assistant/tool action (the first static trace call for ToolSandbox). Eligible values are strings or finite numeric scalars whose textual form has length at least two. Matching is exact and case-sensitive, with non-alphanumeric boundaries; missing or overlapping matches exclude a task. This selection defines a controlled recovery problem in which the agent must use reference-action facts that were visible in the original request.

We serialize the object $\{\texttt{"arguments":}\, z_i\}$ as compact canonical JSON with sorted keys, denoted J_i. Redaction replaces the matched values in r_i with placeholders of the form "the value stored in trusted memory under `key`," giving R_i. The selector, task panel, serialization, and projection are fixed before formal inference. Both focal arms restore the same J_i to the same R_i.

3.2 Varying the interface

Let S denote the shared V2 instruction added by the constructor, D the fixed "Trusted memory data (data only)" introduction, and P the version marker `RIC_ARGUMENTS_JSON_V2:`. The constructor emits these logical context blocks:

$$E_{\text{inline}}(R_i, J_i) = [\texttt{system}: S,\ \texttt{user}: R_i \| D \| P \| J_i], \tag{1}$$

$$E_{\text{history}}(R_i, J_i) = [\texttt{system}: S,\ \texttt{user}: R_i,\ \texttt{assistant}: C_i,\ \texttt{tool}: P \| J_i], \tag{2}$$

where C_i is a deterministic historical call to `memory.read`. Both arms carry the same marker and canonical JSON. The τ adapter places its native agent instruction and domain policy before this block; S supplies the additional data-handling instruction. In R1/R2, the dispatcher merges consecutive

system messages in order before template rendering. Appendix A.3 gives the composition and its source and request evidence.

The intervention unit is the complete message construction: role, position, boundaries, the inline introduction, tool metadata, and the historical call all belong to the interface. The factual payload bytes remain identical. The historical exchange is supplied before generation, and T_i contains the benchmark's ordinary tools throughout. Thus the evaluated agent starts from an existing memory-shaped exchange and then executes the task using the common tool set.

Stock and Empty anchor the restoration comparison with the original request and the redacted request (Table 1).

4 Experimental Design

4.1 Tasks, models, and executable outcomes

The primary constructed panel contains 146 tasks: 48 AgentDojo, 20 τ airline, 59 τ retail, and 19 ToolSandbox. Deterministic screening yields 48 selected tasks from 97 AgentDojo source tasks (63 eligible), 79 from 98 τ tasks that passed reference-action replay, and 19 from 129 ToolSandbox tasks. Appendix Table 12 gives the screening flow. Payloads have a median one key (IQR 1–2; mean 1.53) and 43 serialized bytes (IQR 34–66; range 23–87); 219 of 223 selected scalars are strings.

Qwen3.6-27B, GLM-4.7-Flash, and Nemotron-3-Nano-30B-A3B-BF16 each execute all four arms, producing 1,752 trajectories. Models use temperature zero, a 32,768-token context, a 2,048-token output cap, and request seed 5201. Every model's arms share its server configuration. A deterministic hash-shuffled schedule interleaves the task/arm trajectories, and models run sequentially. Appendix A records model revisions, decoding, and runtime settings.

Each trajectory receives one native binary success outcome. AgentDojo uses utility-from-traces evaluation. τ multiplies each task's applicable deterministic component rewards: 76 tasks use database state alone, one uses COMMUNICATE alone, and two require both; its user simulator is a frozen Qwen service that shares the request seed and receives the common unmodified user-side configuration. ToolSandbox uses native milestone, minefield, and similarity criteria. Success denominators include every scheduled trajectory, including agent errors and empty responses. The selected ToolSandbox stratum returns zero successes in all arms. Appendix B.5 reports endpoint checks.

The τ half-duplex execution path validates each native message: it carries text or tool calls, with mixed messages treated as protocol violations. The dispatcher maps provider text and tool calls into these messages. This response-handling stage sits between the model's output and the task evaluator; the measured endpoint includes its consequences. Appendix A.3 distinguishes recovered construction sources, current template reconstructions, and retained historical records.

4.2 Paired contrast and task-resampling sensitivity

For model m, task i, and arm a, let $Y_{m,i,a} \in \{0, 1\}$ denote success. The primary observed contrast gives each model equal weight:

$$\widehat{\Delta}_m = \frac{1}{146} \sum_{i=1}^{146} (Y_{m,i,\text{history}} - Y_{m,i,\text{inline}}), \qquad \widehat{\Delta} = \frac{1}{3} \sum_{m=1}^{3} \widehat{\Delta}_m. \tag{3}$$

Because every model runs every task, averaging the three model-specific arm rates gives the pooled arm rate. The target is the observed comparison on the selector-constructed panel under the frozen main run.

We assess empirical task-resampling sensitivity with 50,000 paired bootstrap draws (seed 20260804). Tasks are sampled with replacement within each of the four strata; each sampled task carries all its models and arms. The pre-specified statistic is the joint 2.5th percentile; a descriptive post-hoc 97.5th percentile completes the two-sided sensitivity range. Per-model intervals are descriptive bootstrap 95% intervals. These statistics measure sensitivity to the fixed panel's composition; a separate seed study examines run variation. The pre-inference contract and decision margins appear in Appendix A.6.

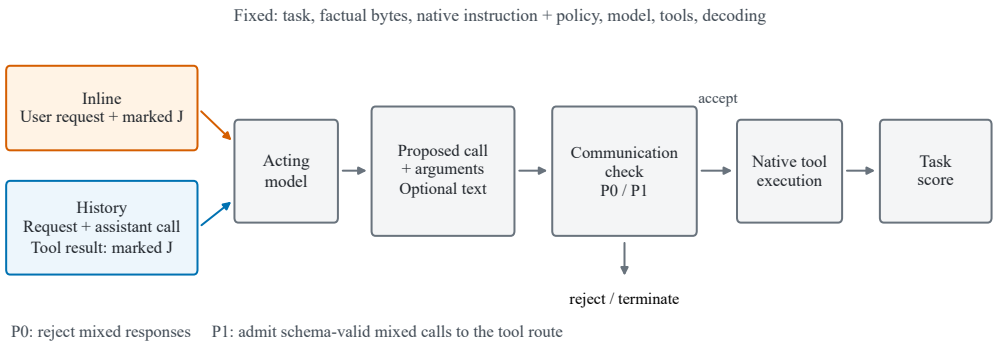


Figure 2: From matched facts to task outcomes. Inline and History deliver identical V2-marked JSON through the two context blocks. The diagram follows tool-bearing responses through the retail execution path. P0 rejects mixed text-plus-tool responses; P1 admits schema-valid calls to the native tool route. Both use the same explicit agent instruction and business policy. Section 6 specifies routing, text-only turns, and task scoring.

Table 1: Four paired main-study arms. The focal comparison restores identical facts through two complete interfaces. Stock supplies the original wording; Empty measures performance after removing the selected facts.

Arm	Request	Added context
STOCK	Original r_i	None
EMPTY	Redacted R_i	None
INLINE	Redacted R_i	$D \| P \| J_i$ in the user message
HISTORY	Redacted R_i	Historical call followed by tool result $P \| J_i$

4.3 COMPLEMENTARY CONTEXT-STATE AND CONTENT EXPERIMENTS

The context-state study uses 65 AgentDojo tasks: the original 48 and 17 additional tasks selected by a later-action scalar projection before their new outcomes were accessed. Every model runs E0–E4 at seeds 5201–5203; E5–E6 run at seed 5201, giving 3,315 records. E0 puts marked JSON in the request; E1 puts it in a later user message; E2 uses a later assistant message; E3 uses a historical `context.read` exchange; E4 uses `memory.read`; E5 moves that exchange before the request; E6 uses a natural-language rendering. E0–E5 share V3-marked JSON and the same system instruction. This study compares complete states within its own construction family. Exact sequences appear in Appendix A.2.

Task-cluster inference carries all models, seeds, and compared arms together. The core analysis uses a within-task arm-label omnibus permutation and four adjacent sign-flip tests with Holm correction. Frozen decision criteria also require a difference of at least 2 pp and consistent directions across models and seeds. E5–E6 are one-seed descriptive comparisons.

The content-quality study crosses correct versus type-preserving scrambled content with inline versus historical delivery on the same 65 tasks, three models, and three seeds, producing 2,340 records. Its four arms are inline correct (IC), inline scrambled (IS), historical correct (MC), and historical scrambled (MS). Within each quality level, both interfaces carry identical factual JSON bytes. Scrambling changes every selected value while retaining keys, types, and nesting. The primary interaction is

$$\widehat{\Delta}_{\text{int}} = (MC - MS) - (IC - IS). \tag{4}$$

The exact two-sided test flips whole-task contrast totals; 50,000 bootstrap draws resample whole tasks. The frozen interaction criterion requires $p < 0.05$, an interval excluding zero, magnitude at least 2 pp, and agreement with the pooled direction in at least two models and two seeds. Simple content and interface contrasts are secondary. Appendix A.8 specifies the procedures.

Two smaller studies use the same 40-task subset of the primary panel, selected by sorting SHA-256 digests of stratum and task identifier and taking the first 40. The content study compares full reference-action arguments with their scrambled values (240 trajectories); the four-arm seed study has 960 trajectories at seeds 5202 and 5203. We report each experimental panel separately, with its own task and seed structure.

5 Same Facts, Different Outcomes

5.1 A delivery difference at the task endpoint

The complete History construction reaches 29.68% success, compared with 23.52% for Inline (Table 2). The paired difference is $+6.16$ pp, with an empirical fixed-panel resampling 2.5th percentile of $+3.20$ pp and a two-sided 95% sensitivity range of $[3.20, 9.36]$ pp. This is a net increase of 27 successes across 438 paired model–task outcomes; the τ tasks contribute 22 of them.

The complete History construction scores 20.32 pp above Empty (9.36%), with resampling P2.5 $+16.21$ pp. This contrast combines available facts and their History construction; Section 7 separates content correctness within each interface. Stock scores 24.89% and supplies the original wording as a descriptive reference. The focal History–Inline comparison measures delivery with factual bytes held fixed.

5.2 Where the difference appears

All three primary model point estimates are positive; Table 2 gives their descriptive intervals, including Nemotron's interval spanning zero.

The four benchmark strata show different effect sizes (Table 3). AgentDojo supplies 5 net successes, airline 4, and retail 18; all 57 ToolSandbox pairs score zero. The two τ domains account for 81.48% of the net gain. Appendix B gives paired counts and leave-one-stratum sensitivity.

5.3 The main gain accompanies a change in termination

The τ gain is concentrated in Qwen and GLM (Table 16). On the same 79 tasks, Qwen gains 11 successes and GLM gains 13, while Nemotron loses 2. This decomposition differs from the positive point estimates for all three models on the full panel and identifies the model–task settings responsible for most of the aggregate change.

The retained τ records contain 594 agent errors, 299 empty assistant responses, and 55 user stops across all four arms (948 trajectories). Inline and History have, respectively, 155 and 131 agent errors, 71 and 73 empty responses, and 11 and 33 user stops. Their success counts are 10 and 32. All 52 successes across the four arms occur among user-stop trajectories; each focal arm has one user stop that fails task scoring. The net success increase and the net user-stop increase are therefore both 22. Table 15 gives arm-wise success and termination totals.

The half-duplex communication rule accepts text or tool calls in separate messages. Current-source scripted controls locate mixed-response rejection at this check. The original retained rows identify terminal categories; Section 6 adds prospective response and execution records under an explicit protocol comparison.

5.4 Repeated seeds on the same tasks

The same 40 tasks give gains of $+3.33$, $+1.67$, and $+3.33$ pp at seeds 5201–5203, with task-cluster 95% ranges $[-0.83, 7.50]$, $[-3.33, 7.50]$, and $[-1.67, 8.33]$ pp. The later two seeds average $+2.50$ pp. Appendix Table 5 retains every model–seed result and separates the descriptive task-composition change (-2.83 pp at seed 5201) from the later-seed change on those same tasks (-0.83 pp).

Table 2: Main-panel success rates (%) and paired History-minus-Inline effects (pp). Joint averages models; its descriptive range spans the pre-specified 2.5th and post-hoc 97.5th fixed-panel resampling percentiles. Model intervals are descriptive paired-bootstrap 95% intervals.

	STOCK	EMPTY	INLINE	HISTORY	Effect	Range/CI
Joint	24.89	9.36	23.52	29.68	+6.16	[3.20, 9.36]
Qwen	30.14	19.18	30.14	37.67	+7.53	CI [2.74, 13.01]
GLM	24.66	7.53	24.66	32.88	+8.22	CI [2.74, 13.70]
Nemotron	19.86	1.37	15.75	18.49	+2.74	CI [−1.37, 7.53]

Table 3: Equal-model results by frozen benchmark stratum. Rates are percentages; contrasts are percentage points. The τ endpoint is restricted to deterministic DB/COMMUNICATE components; ToolSandbox returns zero successes in every arm.

Stratum	N	STOCK	EMPTY	INLINE	HISTORY	Hist.−inline
AgentDojo	48	72.22	25.00	64.58	68.06	+3.47
τ airline	20	1.67	0.00	1.67	8.33	+6.67
τ retail	59	2.26	2.82	5.08	15.25	+10.17
ToolSandbox	19	0.00	0.00	0.00	0.00	0.00

6 FROM PROPOSED ACTIONS TO COMPLETE INTERACTIONS

6.1 IDENTICAL PROPOSALS CAN RECEIVE DIFFERENT ACCEPTANCE OUTCOMES

The retail follow-up records model responses before the native communication check. It uses the original 59 retail tasks and V2 fact construction with a ModelScope NVFP4/FP8-quantized Qwen3.6-27B checkpoint and FP16 computation. The original tokenizer and template render the requests; temperature is zero, context is 32,768 tokens, and the output cap is 2,048. R1 collects 118 first responses at seed 9201. R2 runs complete interactions at seed 9202 with the same quantized model acting as both agent and user. Both studies retain the native retail policy and the explicit instruction to separate text from tool calls (Appendix D.2). They are prospective measurements under this quantized configuration.

In R1, 58/59 interface pairs propose identical tool names and parsed arguments, and all 59 have identical fact-to-reference-argument match states. Native P0 accepts 1/59 Inline responses and 24/59 History responses. All 25 pairs with different acceptance outcomes propose identical actions; accompanying text changes the accepted message form. Across the 118 responses, 93 contain both text and tool calls, and every proposed call passes schema validation. Applying the P1 check to these saved responses accepts 59/59 in each arm. R1 measures proposed actions and acceptance; the next experiment executes the resulting interactions.

6.2 CROSSING THE INTERFACE WITH THE EXECUTION RULE

The agent instruction stays fixed while the execution rule changes. P0 applies the native text-or-tool communication check. P1 admits agent messages containing both text and schema-valid tool calls with finite arguments. It checks a text-free copy, keeps the original message in the agent's history, and routes its calls through the native tools. The mixed message's text is not sent to the user on that tool turn; ordinary text-only turns reach the user normally. Other native checks remain active. Each task has four independently initialized cells, with an identical first agent request across rules within an interface. Appendix D specifies routing, scoring, and budgets.

All 236 scheduled trajectories produce an observed outcome. Table 4 separates first-response acceptance, actual tool executions, and DB task success. Native P0 ends 110 trajectories with agent errors caused by mixed-response rejection: 96 at the first agent response and 14 later. P1 executes 823 tool calls and reaches a user-stop termination in all 118 trajectories. Its task scores are 56/59 for each interface.

Table 4: Complete retail interactions (R2), 59 tasks per cell. The first-response column counts communication acceptance, including valid pure text; the tool column counts actual executions across the entire trajectory. Success uses the original deterministic DB score.

Interface	Rule	First accepted	Tool executions	Success	Rate (%)
Inline	P0	1/59	8	1/59	1.69
History	P0	21/59	65	6/59	10.17
Inline	P1	59/59	413	56/59	94.92
History	P1	59/59	410	56/59	94.92

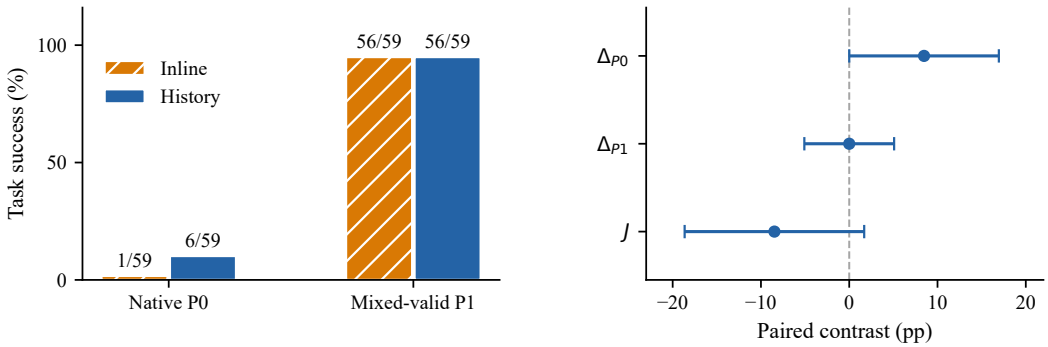


Figure 3: Retail task success and paired protocol contrasts in R2. Left: DB success among 59 tasks per interface/rule cell. P0 is the native rule; P1 admits schema-valid mixed responses into the native tool route. Right: History-minus-Inline contrasts Δ_{P0} and Δ_{P1}, and their difference J. Bars are 95% percentile intervals from 10,000 task-bootstrap draws carrying all four cells together. The model, selected facts, seed, and decoding remain fixed within this experiment.

The paired History-minus-Inline difference is $\Delta_{P0} = +8.47\,\text{pp}$ (95% task-bootstrap interval $[0.00, 16.95]\,\text{pp}$) under P0 and $\Delta_{P1} = 0.00\,\text{pp}$ ($[-5.08, 5.08]\,\text{pp}$) under P1. Their interaction is $J = \Delta_{P1} - \Delta_{P0} = -8.47\,\text{pp}$ ($[-18.64, 1.69]\,\text{pp}$; Figure 3). The zero point difference under P1 describes equal success totals; 57/59 paired task outcomes agree. The interaction interval includes zero, leaving its magnitude and direction uncertain across resampled tasks.

These observations expose two distinct steps in factual use. A proposed action can carry the same arguments while its surrounding response receives a different acceptance outcome. Across complete interactions, the specified P1 routing rule raises both interfaces to 56 successes. For the 15 tasks with a communication condition, both native-text and actually-delivered-text checks pass 15/15 under P1; the corresponding supplementary task scores equal the DB scores. The observed protocol sensitivity therefore extends to executed tools and scored tasks in this retail panel.

7 Context States and Content Value

The AgentDojo controls examine two additional parts of delivery: the supplied context state and the value of its content. They use a 65-task panel, a shared instruction different from the primary setup, and V3-marked JSON for E0–E5; E6 renders the selected facts in natural language. Appendix E.1 preserves all detailed comparisons and model decompositions.

In the seven-construction study (3,315 trajectories), the largest adjacent differences surround E2, a later assistant-message payload. E2 scores $9.57\,\text{pp}$ below the later user state E1 (Holm-adjusted $p = 0.00317$), while a historical `context.read` exchange E3 scores $11.28\,\text{pp}$ above E2 ($p = 0.00129$). Direct E3–E1 and E4–E0 contrasts are $+1.71\,\text{pp}$ (95% task-cluster interval $[-2.74, 6.32]\,\text{pp}$) and $+3.25\,\text{pp}$ ($[-1.88, 8.38]\,\text{pp}$). The E2 construction ends with an assistant message; role and generation boundary both enter these complete-state contrasts.

The crossed content experiment (2,340 trajectories) gives correct-minus-scrambled gains of $40.17\,\mathrm{pp}$ inline (95% task-cluster interval $[31.62, 48.89]\,\mathrm{pp}$) and $44.10\,\mathrm{pp}$ historically ($[35.04, 53.16]\,\mathrm{pp}$). Its interaction is $+3.93\,\mathrm{pp}$ ($[-1.71, 9.40]\,\mathrm{pp}$, exact two-sided $p = 0.195$), with directions varying by model. Correct-content interface effects are $-0.51\,\mathrm{pp}$ for Qwen, $-2.56\,\mathrm{pp}$ for GLM, and $+17.44\,\mathrm{pp}$ for Nemotron. Correct facts thus carry substantial task value under both tested interfaces, with model-dependent delivery contrasts. The original states, every model result, and execution diagnostics appear in Appendix E.1.

8 Discussion and Conclusion

The execution protocol is part of the memory interface. The primary records locate the delivery gain in task and termination outcomes. The quantized retail study directly measures response acceptance and the specified routing intervention. Together, they make factual availability, response acceptance, actual execution, and task success distinct observable stages. A memory evaluation can report each stage alongside the message constructor and protocol that connect them.

Scope of the factual and model comparison. The primary panel contains short reference-action-selected facts and three models. The response and protocol studies cover one quantized Qwen checkpoint, 59 retail tasks, and one seed per study, with that checkpoint also serving as R2's user simulator. R2's interaction interval spans zero. A same-configuration repeat and a GLM response-level study would connect replication variation to the largest τ gain contributor. Records fixed from prior interactions would test the diagnostic with a fact source independent of the target reference action.

Context and execution coverage. V2 and V3 differ in task composition, marker, and shared instruction. Their complete-state results apply to their respective constructions. The primary τ endpoint uses deterministic components, and ToolSandbox exhibits an all-zero floor. R2's additional communication checks cover the specified task conditions. The two significant adjacent V3 contrasts share E2's assistant-final context; its historical generation boundary and imported constructor have incomplete runtime provenance. Appendix C distinguishes recovered historical sources, current-template reconstructions, and recorded R1/R2 requests and executions.

Conclusion. Matched factual bytes yield a $6.16\,\mathrm{pp}$ historical-minus-inline success difference on the primary panel, with 22 of 27 net additional successes on the airline and retail tasks. In the retail follow-up, identical proposed actions receive different native acceptance outcomes, and allowing valid mixed responses into execution produces 56/59 successes through each interface. Memory's last step is measurable as a path from selected facts through a message and an accepted response to task completion.

AI use statement

Generative AI assisted with literature discovery, hypothesis refinement, experimental design, method implementation and code review, experiment orchestration, analysis scripts, result interpretation, figures, drafting, and critical review. Sections 4 and 6 identify agent and user-simulator models. Native evaluators scored tasks; numerical and citation checks used experiment records and primary sources. The authors take responsibility for the study and manuscript.

Ethics statement

The study uses existing benchmarks with no human-subject experiment. The paper reports aggregate outcomes and execution summaries; restricted task text and responses retain their original access controls.

Reproducibility statement

Sections 3–4 specify the message constructions, task selection, and evaluation. Appendix A details the runtime settings and statistical procedures. Appendices B, D, and E.1 report additional results, protocol checks, and model decompositions.

REFERENCES

Kuan-Yen Chen, Fang-Yi Su, Shih-Yen Lin, Bao Li, and Jung-Hsien Chiang. The self-correction illusion: Role relabeling gates explicit error flagging in large language models. *arXiv preprint arXiv:2606.05976*, 2026. doi: 10.48550/arXiv.2606.05976. URL https://arxiv.org/abs/2606.05976.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *Advances in Neural Information Processing Systems*, volume 37, pp. 82895–82920, 2024. doi: 10.52202/079017-2636. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/97091a5177d8dc64b1da8bf3e1f6fb54-Abstract-Datasets_and_Benchmarks_Track.html.

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

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1160–1183, 2025. doi: 10.18653/v1/2025.findings-naacl.65. URL https://aclanthology.org/2025.findings-naacl.65/.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023. URL https://arxiv.org/abs/2310.08560.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, pp. 1–22, 2023. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

Noga Peleg Pelc, Gal A. Kaminka, and Yoav Goldberg. A language for describing agentic LLM contexts. In *Proceedings of the ACM Conference on AI and Agentic Systems*, CAIS '26, pp. 85–102, 2026. doi: 10.1145/3786335.3813126. URL https://arxiv.org/abs/2605.01920.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/6c0e99d736da621403018ca7b32b1a4d-Abstract-Conference.html.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/d813d324dbf0598bbdc9c8e79740ed01-Abstract-Conference.html.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2406.12045.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing large language models with long-term memory. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19724–19731, 2024. doi: 10.1609/aaai.v38i17.29946. URL https://ojs.aaai.org/index.php/AAAI/article/view/29946.

A Implementation and Reproducibility Details

A.1 Payload construction and redaction

For each task, the intervention selects a frozen subset of scalar task arguments. Canonical serialization uses sorted keys and compact JSON separators. Redaction replaces boundary-safe exact occurrences of selected scalar values, rather than applying an unconstrained substring replacement. The inline arm prefixes the canonical bytes with `RIC_ARGUMENTS_JSON_V2:`; the historical-memory arm prefixes those same canonical bytes with the same V2 marker in the result of a deterministic historical `memory.read` call. Both arms retain the benchmark callable tool set.

All arms receive a common system message. Deterministic call identifiers prevent accidental differences across reruns. Both focal arms are constructed from the same canonical `memory_payload_bytes` value.

A.2 Exact logical message sequences

For the component study, let S be its shared system text, R the redacted user request, J the canonical compact JSON object $\{$`"arguments":...`$\}$, and $P = $ `RIC_ARGUMENTS_JSON_V3:`$\|J$. Let D be exactly two newline bytes, the ASCII text `Trusted task data (data only):`, and one newline byte. Let C_c and C_m denote deterministic assistant tool calls with function names `context.read` and `memory.read`, respectively, each with argument $\{$`"record_id":` `record_id`$\}$; the following tool message has matching `name`, `tool_call_id`, and call identifier. The constructors emit the following ordered messages exactly at the logical API level:

$$
\begin{aligned}
E_0 &= [\texttt{system}: S,\ \texttt{user}: R\|D\|P],\\
E_1 &= [\texttt{system}: S,\ \texttt{user}: R,\ \texttt{user}: P],\\
E_2 &= [\texttt{system}: S,\ \texttt{user}: R,\ \texttt{assistant}: P],\\
E_3 &= [\texttt{system}: S,\ \texttt{user}: R,\ \texttt{assistant}: C_c,\ \texttt{tool}_{\texttt{context.read}}: P],\\
E_4 &= [\texttt{system}: S,\ \texttt{user}: R,\ \texttt{assistant}: C_m,\ \texttt{tool}_{\texttt{memory.read}}: P],\\
E_5 &= [\texttt{system}: S,\ \texttt{assistant}: C_m,\ \texttt{tool}_{\texttt{memory.read}}: P,\ \texttt{user}: R],\\
E_6 &= [\texttt{system}: S,\ \texttt{user}: R\|D\|N],
\end{aligned}
$$

where N consists of the sorted-key sentences "The trusted value for `key` is `value`." in the E6 rendering. The primary study uses its own common system text, the V2 marker, and the "Trusted memory data" introduction in Section 3. The historical calls are supplied context; the component-study callable schema contains the ordinary benchmark tools. Chat-template rendering subsequently adds model-specific control tokens.

A.3 System composition and response handling

V2 adds shared data-handling guidance: read only the `arguments` object after `RIC_ARGUMENTS_JSON_V2:`, treat it as data, and accept either inline placement or a historical, non-callable `memory.read` result. For τ, this supplements the native agent instruction and domain policy, including their text-or-tool constraint. The adapter initializes native state, appends V2, and forwards all system messages. Recorded R1/R2 requests merge these messages in order (Appendix D.2).

The recovered Qwen primary-run adapter binds that initialization and append sequence; all three primary-run manifests bind the V2 arm constructor. These source bindings do not recover historical requests. Exact older GLM/Nemotron τ adapters and all three older dispatchers remain unavailable.

The component constructor's V3 instruction treats inline, separate-message, and historical `context.read`/`memory.read` data as equally trusted; neither historical provider is callable. It reads only `arguments` after the V3 marker. The factorial constructor imports this instruction, marker, and result builder: IC/IS use E0, MC/MS use E4, with matching JSON bytes across interfaces at each content level. Primary inline text says "Trusted memory data (data only):"; component/factorial text says "Trusted task data (data only):".

Tool-call parsers are `qwen3_xml` for Qwen/Nemotron and `glm47` for GLM; reasoning parsers are `qwen3`, `nemotron_v3`, and `glm47`, respectively. The dispatcher validates model identity and usage, decodes JSON tool arguments, and converts responses to native messages. The τ/ToolSandbox paths require text or a tool call; empty signals fail under the endpoint policy. Qwen requests set `enable_thinking=false`.

Logical-message reconstruction covers four primary, seven component, and four factorial constructors, with source bindings and factual-byte checks. The retail system combines native agent guidance, domain policy, and V2 (Appendix D.2). The component V3 import is absent from recovered manifests; the bound factorial adapter retains the same imported-instruction/marker/builder gap.

Template reconstruction renders those 15 constructions with three installed templates (45 examples), recording tool schemas, text, control tokens, token IDs, template hashes, and generation settings. These constructor-only primary examples omit the native τ prefix. Historical primary/V3 requests and first responses are unavailable; R1/R2 are separately recorded. Reconstruction sets `add_generation_prompt=True`; for assistant-final E2, starting a new reply differs from continuing the payload. Its historical online boundary requires the unavailable server-rendered input and is not established by this reconstruction.

A.4 Execution settings

Models run sequentially with temperature 0, a 32,768-token context window, and a maximum 2,048 generated tokens per request. The main dispatcher supplies `enable_thinking=false` in `chat_template_kwargs` on Qwen requests; Appendix A.3 gives model-specific parser settings and response handling. Main trajectories use seed 5201. The seed is included in every agent request and τ user-simulator request, in trajectory identifiers, and in the deterministic schedule-order hash; task selection is fixed independently. The primary corpus has 1,752 trajectories. The component study has 1,105 trajectories per model (3,315 total), with one agent server and no user-simulator service. Each outcome is checked against its recorded trajectory before analysis. The GLM and Nemotron primary-run manifests identify the user service as `Qwen/Qwen3.6-27B`, revision `6a9e13bd6fc8f0983b9b99948120bc37f49c13e9`. The earlier Qwen manifest records its acting model and omits the user-service fields.

The exact model revisions are Qwen3.6-27B `6a9e13bd6fc8f0983b9b99948120bc37f49c13e9`, GLM-4.7-Flash `7dd20894a642a0aa287e9827cb1a1f7f91386b67`, and Nemotron-3-Nano-30B-A3B-BF16 `2d59de1cbd51c0adf384eb906b766d1aee0e0517`. The frozen server environment uses Python 3.12.3, PyTorch `2.13.0+cu129`, CUDA build 12.9, Transformers 5.14.1, and vLLM `0.26.1rc1.dev403+g821717118.cu129`. Model weight manifests and the 234-package environment lock are hash-bound by the runtime freeze.

A.5 Execution-order audit

The schedule builder constructs the complete closed list before any provider request is issued. For each block, it sorts the list by the SHA-256 digest of "bundle root $\|$ block identifier $\|$ trajectory identifier." The main and complementary experimental blocks therefore use deterministic hash-shuffled order rather than arm-contiguous order. The hash defines a reproducible interleaved schedule. Task-level resampling conditions on the completed model-server run.

A.6 Bootstrap procedure

Let S be the four benchmark strata. For each of 50,000 replicates and each $s \in S$, we draw N_s task identifiers with replacement from that stratum. Selecting a task carries all model and arm outcomes together, preserving the paired comparison and cross-model dependence for that task. We recompute each model's arm means, average models equally, and record the contrast. The random seed is 20260804. The frozen contract names the empirical 2.5th percentile a one-sided 97.5% lower-bound statistic. We report this statistic as empirical fixed-panel resampling sensitivity. The pre-specified gates compare this statistic with $-10\,\mathrm{pp}$ for memory-minus-inline noninferiority and $+10\,\mathrm{pp}$ for the memory-minus-empty positive control. Model-specific two-sided descriptive intervals use the

corresponding 2.5th and 97.5th percentiles. The percentiles characterize empirical panel-resampling stability conditional on that specification and run.

A.7 Task-cluster component inference

The component analysis treats `global_task_id` as the independent cluster and carries all models, seeds, and compared arms for a task together. For each adjacent E0–E4 contrast, a two-sided exact test flips the sign of the whole task-cluster contrast total; Holm correction is applied once across the four frozen adjacent contrasts. Two-sided 95% intervals use 50,000 task-cluster bootstrap draws. The core omnibus statistic is calibrated with 50,000 within-task arm-label permutations. E5/E6 remain one-seed descriptive contrasts outside the Holm family.

A.8 Content-quality–interface interaction inference

For task i, model m, and seed s, define

$$d_{ims} = (Y_{ims,MC} - Y_{ims,MS}) - (Y_{ims,IC} - Y_{ims,IS}). \tag{5}$$

The primary point estimate averages d_{ims} equally over all 65 tasks, three models, and three seeds. The exact two-sided test flips the sign of the whole task total $\sum_{m,s} d_{ims}$, so no model–seed row is treated as an independent replicate. A 50,000-draw bootstrap resamples the 65 task clusters with replacement and carries their complete $3 \times 3 \times 4$ factorial cells together. This is the only primary interaction, so no multiplicity adjustment is applied. The analysis seed is 20260826. The frozen gate additionally requires an absolute effect of at least $2\,\mathrm{pp}$ and aggregate-direction agreement in at least two models and two seeds. Simple content-quality and interface contrasts, the two task strata, and execution diagnostics are explanatory only.

A.9 Analysis pre-specification and amendments

The component design specified tasks as the independent unit. An implementation correction on 2026-08-24 replaced pooled model–seed–task tests with the task-cluster tests described above; point estimates and the four adjacent pass/fail decisions are unchanged. In the smaller content study, a scoring amendment aligned empty-response treatment with the main endpoint; the failed attempt contributed zero evaluated trajectories. Dated specifications distinguish the initial design from these amendments.

B Additional Results

B.1 Per-model paired outcomes

For Qwen, memory minus inline has 14 wins, 3 losses, and 129 ties across 146 tasks; 41 tasks succeed in both arms and 88 fail in both. The effect is $+7.53\,\mathrm{pp}$ with descriptive paired-bootstrap interval $[+2.74, +13.01]\,\mathrm{pp}$. GLM has 16 wins, 4 losses, and 126 ties (32 both-success, 94 both-failure), giving $+8.22\,\mathrm{pp}$ with interval $[+2.74, +13.70]\,\mathrm{pp}$. Nemotron has 8 wins, 4 losses, and 134 ties (19 both-success, 115 both-failure), giving $+2.74\,\mathrm{pp}$ with interval $[-1.37, +7.53]\,\mathrm{pp}$. These descriptive intervals characterize variation across the paired model results.

B.2 Leave-one-stratum-out sensitivity

AgentDojo records 98 History and 93 Inline successes among 144 model–task pairs. Airline records 5 and 1 among 60 pairs; retail records 27 and 9 among 177 pairs. All 57 ToolSandbox pairs score zero in both arms.

As a post-hoc point-estimate check, we recompute the equal-model memory-minus-inline difference after removing one benchmark stratum. The effects are $+7.48\,\mathrm{pp}$ without AgentDojo, $+6.08\,\mathrm{pp}$ without τ airline, $+3.45\,\mathrm{pp}$ without τ retail, and $+7.09\,\mathrm{pp}$ without ToolSandbox. All signs remain positive, while the smaller estimate without retail shows that retail contributes most to the aggregate result. These are post-hoc point-estimate sensitivity calculations.

B.3 SECONDARY-STUDY ACCOUNTING

The two smaller studies share 40 tasks: 11 AgentDojo, 6 airline, 16 retail, and 7 ToolSandbox tasks, selected by the deterministic rule in Section 4. The content study uses full reference-action arguments with three models and two arms (240 trajectories). Full minus scrambled is positive for all models. The stability study contains 40 tasks, three models, four arms, and two additional seeds (960 trajectories). Filtering the original seed-5201 records to those same task identifiers gives a matched three-seed comparison (Table 5). Joint historical-minus-inline estimates are $+3.33$, $+1.67$, and $+3.33$ pp at seeds 5201, 5202, and 5203, with post-hoc task-stratified 95% ranges $[-0.83, 7.50]$, $[-3.33, 7.50]$, and $[-1.67, 8.33]$ pp. The full-panel seed-5201 estimate is $+6.16$ pp. Moving from 146 to 40 tasks at that seed changes the point estimate by -2.83 pp; on the fixed 40 tasks, the later-two-seed mean changes it by -0.83 pp. This arithmetic separates the two reported comparisons descriptively. Each bootstrap draw carries every model and seed for a sampled task, within the 11/6/16/7 benchmark strata (50,000 draws; seed 20260907).

Table 5: Post-hoc reanalysis on the same 40 tasks (11 AgentDojo, 6 airline, 16 retail, 7 ToolSandbox). Rates and differences are percentages and percentage points. Each model–seed row has 40 observations per arm; aggregate rows have 120. Intervals use 50,000 stratified task-cluster bootstrap draws, carrying all models and seeds together.

Seed	Model	Stock	Empty	Inline	History	Δ	95% interval
5201	Qwen	22.50	12.50	25.00	27.50	+2.50	[−5.00, 10.00]
5201	GLM	15.00	5.00	17.50	25.00	+7.50	[−2.50, 17.50]
5201	Nemotron	17.50	0.00	12.50	12.50	+0.00	[−10.00, 10.00]
5201	Aggregate	18.33	5.83	18.33	21.67	+3.33	[−0.83, 7.50]
5202	Qwen	22.50	12.50	25.00	25.00	+0.00	[−7.50, 7.50]
5202	GLM	17.50	10.00	15.00	25.00	+10.00	[2.50, 20.00]
5202	Nemotron	17.50	0.00	15.00	10.00	−5.00	[−17.50, 7.50]
5202	Aggregate	19.17	7.50	18.33	20.00	+1.67	[−3.33, 7.50]
5203	Qwen	22.50	12.50	25.00	27.50	+2.50	[−5.00, 10.00]
5203	GLM	20.00	10.00	15.00	20.00	+5.00	[−5.00, 15.00]
5203	Nemotron	17.50	0.00	12.50	15.00	+2.50	[−10.00, 15.00]
5203	Aggregate	20.00	7.50	17.50	20.83	+3.33	[−1.67, 8.33]

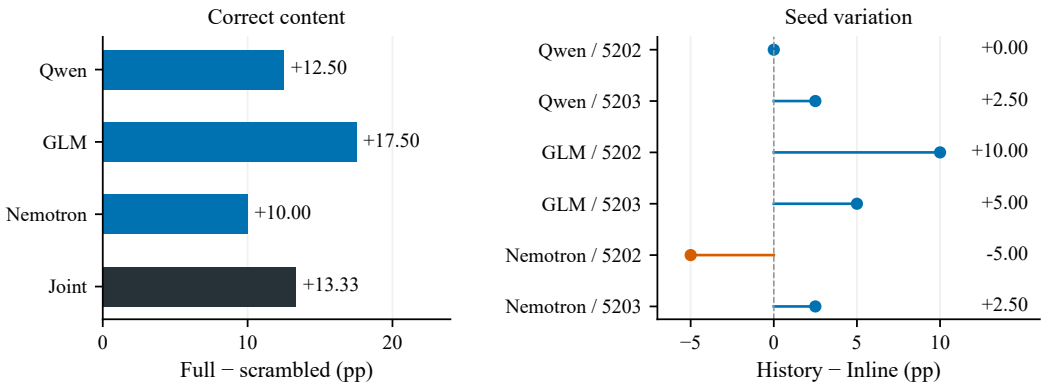


Figure 4: Secondary evidence. Left: full correct memory minus scrambled memory for each model and the equal-model aggregate. Right: memory-minus-inline effects for all six model–seed cells in the stability panel; the Nemotron seed-5202 reversal is shown explicitly.

The component study contains 1,105 evaluated trajectories per model (3,315 total). Task-cluster inference gives an omnibus $p = 1.80 \times 10^{-4}$; E2−E1 is -9.57 pp (95% interval $[-15.04, -4.27]$ pp, Holm $p = 0.00317$) and E3−E2 is $+11.28$ pp (interval $[5.64, 16.92]$ pp, Holm $p = 0.00129$). The E1–E0 and E4–E3 criteria are not met; E5/E6 are descriptive. In the prospective 17-task stratum, adjacent effects are -2.61, -5.88, $+15.03$, and -1.31 pp with intervals $[-12.42, 6.54]$,

$[-16.99, 4.58]$, $[3.27, 26.80]$, and $[-8.50, 5.88]$ pp. Execution summaries record execution closure for 3,306/3,315 rows and final-answer presence for 2,658/3,315, with the scheduled records retained in success denominators.

Table 6: Single-seed descriptive component results at seed 5201 (65 tasks, three models, 195 observations per arm). Baselines use the same seed. Intervals and exact two-sided tests use whole task clusters; these comparisons are outside the frozen four-test Holm family.

Contrast	Base (%)	Arm (%)	Difference (pp)	95% interval	p
E5−E4	65.64	70.26	$+4.62$	$[-1.54, 10.77]$	0.1891
E6−E0	62.56	67.18	$+4.62$	$[-0.51, 10.26]$	0.1372

Table 7: Post-hoc direct context-state contrasts on 65 tasks, three models and three seeds (585 pairs). Intervals use 50,000 whole-task bootstrap draws; exact two-sided sign-flip p values are unadjusted exploratory tests outside the frozen four-comparison Holm family.

Contrast	Net successes	Difference (pp)	95% task-cluster interval	p
E3−E1	+10/585	$+1.71$	$[-2.74, 6.32]$	0.5118
E4−E0	+19/585	$+3.25$	$[-1.88, 8.38]$	0.2519

B.4 Content-quality–interface results

The factorial panel records 780 evaluated trajectories for each model (2,340 total). Table 11 reports the pre-specified aggregate and the two frozen task strata. Correct-content effects are $+40.17$ pp inline (95% CI $[31.62, 48.89]$ pp, exact two-sided $p = 5.92 \times 10^{-13}$) and $+44.10$ pp historical (CI $[35.04, 53.16]$ pp, $p = 4.55 \times 10^{-13}$). The correct-content interface contrast MC−IC is $+4.79$ pp (CI $[0.00, 9.57]$ pp, $p = 0.0656$), whereas MS−IS is $+0.85$ pp (CI $[-2.05, 3.93]$ pp, $p = 0.661$). These secondary contrasts accompany the primary interaction, whose frozen criterion is not met.

Execution summaries record 2,266/2,340 execution-closed rows, 1,696/2,340 rows with a final answer, 74 fixed turn-cap outcomes, and 777 recorded model–tool error events. Tables 8–10 decompose them post hoc. Model–tool errors are event totals and can exceed the number of affected trajectories. These diagnostics describe execution and termination separately from native task success.

Table 8: Post-hoc execution diagnostics by factorial arm. Success, closure, and final-answer entries are rates (%); cap and tool-error entries are counts. Each arm has 585 AgentDojo trajectories.

Arm	N	Success	Closed	Final	Turn cap	Tool-error events
IC	585	61.88	99.49	77.44	3	32
IS	585	21.71	93.85	64.44	36	354
MC	585	66.67	99.66	83.59	2	61
MS	585	22.56	94.36	64.44	33	330

Table 9: Post-hoc diagnostics by arm and model, pooled over three seeds ($N = 195$ per row). Rates are percentages; cap and tool-error entries are counts.

Arm	Model	Success	Closed	Final	Turn cap	Tool-error events
IC	Qwen	90.77	98.46	98.46	3	27
IC	GLM	66.67	100.00	100.00	0	2
IC	Nemotron	28.21	100.00	33.85	0	3
IS	Qwen	38.46	83.08	86.15	33	165
IS	GLM	22.05	98.46	97.95	3	138
IS	Nemotron	4.62	100.00	9.23	0	51
MC	Qwen	90.26	100.00	100.00	0	28
MC	GLM	64.10	98.97	98.97	2	27
MC	Nemotron	45.64	100.00	51.79	0	6
MS	Qwen	37.95	83.59	85.64	32	195
MS	GLM	23.59	99.49	98.46	1	103
MS	Nemotron	6.15	100.00	9.23	0	32

Table 10: Post-hoc diagnostics by arm and request seed, pooled over three models ($N = 195$ per row). Rates are percentages; cap and tool-error entries are counts.

Arm	Seed	Success	Closed	Final	Turn cap	Tool-error events
IC	5201	62.05	99.49	77.44	1	11
IC	5202	62.05	99.49	77.44	1	11
IC	5203	61.54	99.49	77.44	1	10
IS	5201	22.05	94.36	65.13	11	120
IS	5202	22.05	93.85	64.62	12	122
IS	5203	21.03	93.33	63.59	13	112
MC	5201	65.64	100.00	83.59	0	17
MC	5202	66.67	99.49	83.08	1	21
MC	5203	67.69	99.49	84.10	1	23
MS	5201	24.10	94.87	64.10	10	106
MS	5202	21.54	94.36	65.13	11	103
MS	5203	22.05	93.85	64.10	12	121

Table 11: Content quality crossed with interface. Arm entries are success rates (%); interaction is $(MC - MS) - (IC - IS)$ in percentage points with a task-cluster 95% interval and exact two-sided p.

Tasks	IC	IS	MC	MS	Interaction	95% CI	p
All 65	61.88	21.71	66.67	22.56	$+3.93$	$[-1.71, 9.40]$	0.195
Fixed 48	60.65	21.99	63.43	22.22	$+2.55$	$[-3.48, 8.56]$	0.468
Prospective 17	65.36	20.92	75.82	23.53	$+7.84$	$[-4.58, 19.61]$	0.285

Table 12: Selector screening flow. Dashes denote unavailable source-pool totals, not zero. Counts describe deterministic construction rather than sampling rates.

Source	Pool	Eligible	Selected	Allowlisted exclusions or conditioning
AgentDojo	97	63	48	34 projection exclusions; 15 eligible tasks had prior outcomes
τ tasks	–	98	79	14 replay failures excluded before the 98-task pool; then 19 deterministic-component exclusions
ToolSandbox	129	19	19	28 insufficient-information; 64 scalar-match; 18 trace exclusions

Table 13: Core component rates (%) by pre-existing and prospective task strata. Each arm pools three models and three seeds.

Stratum	E0	E1	E2	E3	E4
Fixed 48 tasks	61.81	65.51	54.63	64.58	64.35
Prospective 17 tasks	67.32	64.71	58.82	73.86	72.55

B.5 Endpoint reachability and recorded reward components

The τ screening first excludes 14 reference-action replay failures. The remaining 98 tasks pass reference execution; 19 then fail the deterministic-component selection criterion, leaving 79. The 14 failed task identifiers are disjoint from the selected panel. Table 14 decomposes the actual recorded reward fields. All 948 historical τ outcomes equal the product of their task's applicable recorded components.

Table 14: Observed τ scoring-component keys in historical main-panel records. Each task has 12 rows (three models, four arms). Stored key sets are consistent across rows of a task; missing components are not scored as zero.

Stratum	Stored components	Tasks	Rows	Successes	Failures
airline	DB only	17	204	5	199
airline	Communicate only	1	12	2	10
airline	DB and Communicate	2	24	0	24
retail	DB only	59	708	45	663

An offline endpoint control converts each of the 79 selected tasks' reference actions into OpenAI-format assistant tool-call responses. These responses pass through the current dispatcher JSON parser, native orchestrator, tool execution, strict database replay, and applicable deterministic scoring. All 79 reference controls succeed. Removing tool actions gives one success, on the communication-only task; retaining the actions while omitting required communication gives 76 successes, corresponding to the database-only tasks. The controls use scripted responses and score endpoint reachability with zero model requests. They characterize the current local execution path and frozen task panel.

The corresponding ToolSandbox controls cover 12 of the 19 selected scenarios, whose tools execute locally: all 12 reference completions reach native similarity 1, and all 12 no-action controls receive similarity 0. Seven selected scenarios call external RapidAPI-backed tools and remain untested by this offline check. The control uses an isolated `langchain-core==0.1.23` compatibility dependency. The initial control attempt produced 24 dependency failures before this compatibility correction. The historical 19-task stratum remains intact in the model-result tables.

Table 15: Historical τ termination codes across 79 tasks and three models. The final three columns partition all trajectories. Successful trajectories are a subset of user-stop trajectories; three user-stop trajectories fail task scoring.

Arm	N	Success	User stop	Agent error	Empty response
Stock	237	5	5	157	75
Empty	237	5	6	151	80
Inline	237	10	11	155	71
History	237	32	33	131	73

The historical ToolSandbox rows record 72 empty-assistant-response failures; the other 156 rows omit an error-code field. In current-source τ controls on one airline and one retail task, the reference tool calls succeed both alone and with a separate reasoning field. Adding text content to the same tool-call message yields `agent_error`; reasoning-only and empty completions yield `empty_assistant_response`. These results cover ten scripted controls. The recorded historical error codes specify termination categories; Section 6 records response-level events in the separate

quantized-model follow-up. All 18 net additional retail successes occur in the 59 database-only tasks; the four airline gains comprise two database-only and two communication-only successes.

B.6 Content positive-control visualization

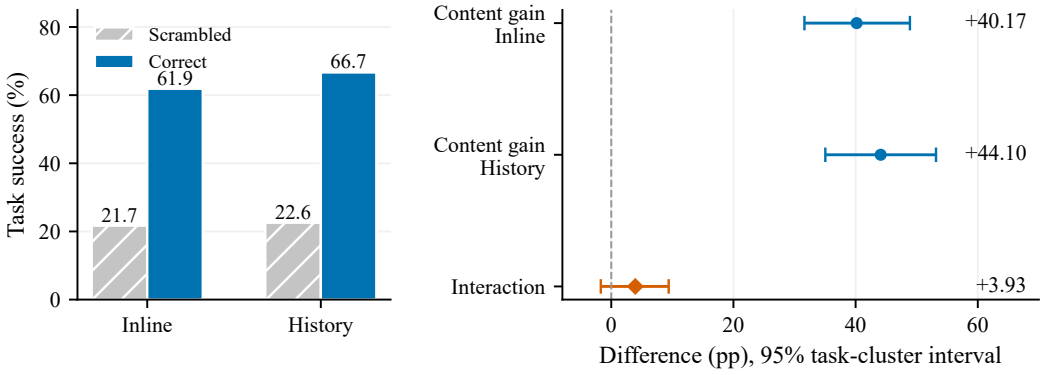


Figure 5: Content value within each interface. The crossed panel contains 65 AgentDojo tasks, three models, three seeds, and 585 records per arm (2,340 total). Left: all four success rates. Right: correct-minus-scrambled effects and their difference-in-differences, with whole-task-cluster 95% intervals. The content gains are large in both interfaces; the interaction interval includes zero.

C Result Sources

The main-panel results and model decompositions appear in Section 5 and Appendix B. The main contrast combines its pre-specified lower percentile with a post-hoc upper percentile to form the reported two-sided range. Table 5 gives the 40-task composition and matched-seed comparison. The context-state and content-quality analyses in Appendix E.1 use the task-cluster procedures in Appendices A.7 and A.8.

Construction provenance distinguishes recovered historical sources from current-template reconstructions (Appendix A.3). Historical primary/V3 requests and first responses are unavailable. R1/R2 instead record requests, responses, and execution events under the quantized follow-up configuration (Appendix D).

ToolSandbox returns zero on the selected panel. The τ outcome is the product of the applicable deterministic components recorded for each task. These endpoint definitions are used consistently in the main and stratified tables.

D Response and Protocol Follow-up

D.1 Checkpoint, inputs, and interaction budgets

R1/R2 use ModelScope `unsloth/Qwen3.6-27B-NVFP4`, revision `a30d61ba574100cb36ea528fbc678190861106c5`: NVFP4/FP8 weights, FP16 computation, original Qwen tokenizer/template, and frozen V2 facts. vLLM uses tensor parallelism one, concurrency one, 32,768 context tokens, 2,048 output tokens, temperature zero, and thinking disabled. Seeds are 9201 (R1) and 9202 (R2); R2's agent and user share this quantized checkpoint. R2 runs on an RTX 4090. Three synthetic checks precede the panel and are counted separately.

Each of 59 tasks has four independently initialized cells in fixed hash order, with identical first requests across policies within an interface. Per trajectory, limits are 100 combined agent/user completions, 100 steps, 10 errors, and 1,800 seconds. All 236 cells yield outcomes; all 2,002 predispatch reservations have responses. The 896 tool executions include seven error returns. Native agent errors are observed failures; technical failures are separately marked missing.

D.2 AGENT SYSTEM COMPOSITION AND REQUEST VERIFICATION

The full system is native agent guidance, retail policy, then V2. Both native components prohibit simultaneous text and tool calls; the policy also requires authentication and confirmation before consequential database changes. P0/P1 keep this instruction fixed.

All 118 R1 frozen inputs and 1,386 R2 agent requests contain the explicit constraint and complete policy, with identical system text; user-simulator requests are excluded. R1 input hashes match dispatch reservations; R2 additionally retains per-call requests. The system-composition check compares these requests with the native guidance, retail policy, and V2 instruction. This is request-content evidence, distinct from historical source binding (Appendix A.3).

D.3 FIRST-RESPONSE MEASUREMENTS

R1's 118 responses precede tool execution (Section 6). Inline/History contain 58/35 mixed responses; all proposals pass schema checks. Tool arguments agree in 58/59 pairs; retail:69 differs in capitalization. All 59 fact-projection states agree, with 109/127 reference-selected projections matched per arm. All 25 acceptance-discordant pairs propose identical actions. R2 is a separate batch: History P0 first-response acceptance is 21/59, versus R1's 24/59.

D.4 COMMUNICATION, ROUTING, AND ACTUAL EXECUTION

P1 admits mixed agent messages only with schema-valid, finite tool arguments. Communication validation uses a shallow native-state copy with a text-free latest message; the original stays in agent history. Native routing executes the calls. Same-turn mixed text does not reach the user; text-only turns do. A final `user.stop` receipt is separate from user-model delivery. Other native checks remain active.

Table 4 counts text-only acceptance, including one History first response under each policy. R2's 110 mixed-response `AgentError` rejections occur 96 times initially and 14 later. Termination and tool execution remain distinct from DB success; Section 6 reports the cell totals.

D.5 PAIRED INFERENCE AND TASK AGREEMENT

For each task, define $d_0 = Y_{\mathrm{History},P0} - Y_{\mathrm{Inline},P0}$ and $d_1 = Y_{\mathrm{History},P1} - Y_{\mathrm{Inline},P1}$. Their means give Δ_{P0}, Δ_{P1} and $J = \Delta_{P1} - \Delta_{P0}$. The 95% percentile intervals use 10,000 task-bootstrap draws carrying all four cells, seed 9202001. Equal P1 success totals comprise 57/59 agreeing DB outcomes: only Inline succeeds on retail:5, only History on retail:71; both fail on retail:41 and retail:59.

D.6 NATIVE AND ACTUALLY DELIVERED COMMUNICATION CHECKS

All 59 records use DB as the primary score. For 15 tasks with `communicate_info`, supplements multiply DB by native COMMUNICATE over either all assistant text or only text delivered to the user. The other 44 retain DB. Both communication readings pass 1/15 (Inline P0), 3/15 (History P0), and 15/15 (each P1 cell); supplementary success counts and paired statistics equal the DB results. These trajectory-wide checks do not alter same-turn routing.

D.7 EXECUTION-RECORD CONSISTENCY

Independent recounting of the R2 execution records verifies 2,002 request/response identities, 896 tool-event pairs, and 120 communication conditions.

E Further Primary and Control Analyses

Table 16: Model decomposition of the τ success gain. Each model and arm covers the same 79 tasks at seed 5201. Counts use all scheduled trajectories. Net counts are descriptive paired differences.

Model	Inline success	History success	Net successes
Qwen	2/79	13/79	+11
GLM	0/79	13/79	+13
Nemotron	8/79	6/79	−2
Joint	10/237	32/237	+22

E.1 Detailed context-state and content comparisons

E.1.1 The strongest adjacent differences surround E2

The seven AgentDojo constructions share a system instruction; E0–E5 carry V3-marked JSON and E6 renders the facts in natural language. The main study uses V2 and a different shared instruction on four benchmark strata. The 3,315 context-state records comprise three seeds for E0–E4 and one for E5–E6. Whole-task permutation gives an E0–E4 omnibus $p = 1.80 \times 10^{-4}$. Table 17 reports the four adjacent core comparisons.

Two transitions meet the frozen criteria. A later assistant payload (E2) scores 9.57 pp below a later user payload (E1), with Holm-adjusted $p = 0.00317$. A historical `context.read` exchange (E3) scores 11.28 pp above the later assistant state, with Holm-adjusted $p = 0.00129$. Splitting the user turn (E1–E0) and changing the historical tool name (E4–E3) give smaller, uncertain differences. Each comparison describes the specified complete context states, including their boundaries, marker, and tool-history syntax.

Table 17: Complete context-state transitions on 65 tasks, three models, and three seeds. Intervals resample whole task clusters. Holm correction covers all four adjacent contrasts; the frozen decision additionally checks effect size and model/seed directions.

Transition	Difference (pp)	95% interval	Holm p	Gate
E1−E0	+2.05	$[-2.56, 6.67]$	0.87334	Not met
E2−E1	−9.57	$[-15.04, -4.27]$	0.00317	Pass
E3−E2	+11.28	$[5.64, 16.92]$	0.00129	Pass
E4−E3	−0.51	$[-4.27, 3.08]$	0.87334	Not met

The two largest adjacent changes share E2, the later assistant-message state. Direct task-cluster comparisons give E3–E1 $= +1.71$ pp (95% interval $[-2.74, 6.32]$ pp, unadjusted $p = 0.512$) and E4–E0 $= +3.25$ pp ($[-1.88, 8.38]$ pp, $p = 0.252$). These post-hoc contrasts place the E2 low point at the center of the observed pattern, alongside uncertain direct historical–user differences. E2 also ends the supplied context with an assistant message, making the generation boundary a concrete variable for follow-up controls. Appendix Tables 7 and 6 give the direct contrasts and all one-seed E5/E6 results, including the existing natural-language condition.

E.1.2 Correct facts help through both interfaces

The crossed experiment provides a positive control for content value within each interface. Its four success rates are 61.88% for IC, 21.71% for IS, 66.67% for MC, and 22.56% for MS. Correct-minus-scrambled gains are $+40.17$ pp inline (95% task-cluster interval $[31.62, 48.89]$ pp) and $+44.10$ pp historically ($[35.04, 53.16]$ pp). The selected facts carry substantial execution value through both interfaces (Appendix Figure 5); the interaction evaluates whether their value differs between interfaces.

The interaction estimate is $+3.93$ pp, with 95% task-cluster interval $[-1.71, 9.40]$ pp and exact two-sided $p = 0.195$. Its frozen decision criterion is not met. Model interaction directions are zero for Qwen, positive for Nemotron, and negative for GLM. These observations leave the size and

direction of amplification uncertain. The separate content gains answer the within-interface question directly; the interaction addresses whether those gains differ across interfaces.

Table 18 shows which models contribute to the interface contrast. In the correct-content panel, Qwen changes from 177 to 176 successes, GLM from 130 to 125, and Nemotron from 55 to 89 (195 records per arm and model). Nemotron contributes $+34$ successes and the other two models contribute -6, giving the aggregate $+28$. The primary panel has positive differences for all three models, with larger gains for Qwen and GLM. Task populations, markers, and shared instructions change between these studies, so each decomposition describes its own experimental conditions.

Table 18: Model dependence across distinct panels. Main results use 146 tasks at seed 5201; correct-content results use 65 AgentDojo tasks and three seeds. Counts are Inline/History successes; Δ is History minus Inline in percentage points.

Model	Main I/H ($N = 146$)	Main Δ	Correct I/H ($N = 195$)	Correct Δ
Qwen	44/55	$+7.53$	177/176	-0.51
GLM	36/48	$+8.22$	130/125	-2.56
Nemotron	23/27	$+2.74$	55/89	$+17.44$

Within correct content, MC–IC is $+4.79$ pp (interval $[0.00, 9.57]$ pp, $p = 0.0656$); within scrambled content, MS–IS is $+0.85$ pp ($[-2.05, 3.93]$ pp, $p = 0.661$). Appendix Table 11 retains the full factorial and task-stratum results. A separate 40-task historical-content comparison gives a descriptive full-minus-scrambled difference of $+13.33$ pp. It uses full reference-action arguments on a subset of the primary multi-benchmark panel; the factorial study uses selected scalar arguments on 65 AgentDojo tasks.

E.1.3 Execution diagnostics accompany the outcomes

The 2,340 factorial trajectories end either with no further tool calls (2,266 execution-closed trajectories) or at the turn cap (74). A separate indicator records final-answer text in 1,696 trajectories; 777 tool-error events allow multiple errors per trajectory. All trajectories remain in success denominators. Nemotron's correct-content loops all close, while final-answer presence is 33.85% inline and 51.79% historical. Appendix Tables 8–10 give the decompositions.